

Abstract of thesis entitled

**“Collision Detection for Ellipsoids
and Other Quadrics”**

submitted by

Choi Yi King

for the degree of Doctor of Philosophy

at the University of Hong Kong

in March 2008

Collision detection and collision avoidance in a 3D dynamic environment are critical problems in various applications such as computer animation, virtual reality, computer games, robotics, CAD/CAM, computational physics and computational biology. Collision detection is to detect if different moving objects come to occupy the same spatial position at the same time in a computer simulated environment; it is needed for triggering proper action to either avoid collision or determine correct physical response due to collision. Continuous collision detection, on the other hand, further concerns whether objects are guaranteed to be colliding or collision-free within a continuous time span.

This thesis studies collision detection, as well as continuous collision detection of an important class of surfaces, namely the quadrics, which encompasses all the surfaces of degree two, and its 2D counterpart—the conics. We aim to produce efficient and exact collision detection results, which are much desired in many applications where accurate shape is encoded more efficiently using quadric surfaces than using piecewise linear representation. Our approach is algebraic, therefore it requires *neither* geometric approximation of curved boundary surfaces *nor* discrete sampling of motion time interval, which are the main sources of inefficiency and inaccuracy of the conventional collision detection paradigm.

We deal with the collision detection of ellipses and ellipsoids, whose results could be extended readily to solve the collision problems of other conic curves and quadric

surfaces. We are interested in the composite quadrics models (CQMs) that are comprised of piecewise quadric or linear surface patches, and in particular, the CQMs whose boundary curves are conics sections because such objects are widely used in practice. Base on the algebraic formulation, robust and efficient numerical algorithms are devised to solve the CCD problems.

We illustrate the robustness and effectiveness of our algorithms with experiments and numerical examples, whose results demonstrate that they are practical for collision detection in relation to quadrics or conics primitives.

Collision Detection for Ellipsoids and Other Quadrics

Choi Yi King

蔡綺琮

A thesis submitted in partial fulfilment of the requirements for
the Degree of Doctor of Philosophy
at the University of Hong Kong.

March 2008

Dedicated to my beloved family.

Declaration

I declare that this thesis represents my own work, except when due acknowledgment is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed

Choi Yi King

Acknowledgements

I would like to first of all thank Professor Wenping Wang for his inspiring and insightful guidance throughout the course of my graduate study. I am deeply indebted to his continuous support, encouragement, sharing, belief and trust, which are all important to uphold me in many of my difficult times. This thesis would not have been possible without his invaluable suggestions and comments.

My special thanks go to Prof. Myung-soo Kim for his patient guidance and ingenious thinkings in our joint work. I will always remember his good sense of humor, as well as the nice little gifts that he brought whenever he visited us.

Special thanks are also due to Prof. Jiaye Wang, our “shi gong” (grand-advisor), for his genuine care and teachings. I respect him as a role model, and admire very much his scholarly manner and enthusiasm in research.

I am grateful to Prof. Gershon Elber, Prof. Changhe Tu, Jungwoo Chang for their helpful advice and support. Some results in this thesis are from our joint work. I would also like to thank Prof. Tak-Wah Lam, my respectful mentor, for his hearty concerns, particularly about my career development.

And how can I miss out the “401 gang”—Chan Bin, Liu Yang, Su Qi, Lu Lin, Luo Xi, Liang Chen, Zheng Dayue, Sun Feng, Yan Dongming; and those who have left the group already—Dominic Cheng, Yang Huaiping, Wang Bin, Wang Xiaoting? We have indeed shared a very joyful time in this important period of our lives.

I would like to express my gratitude to my lifetime best friend, Janet Fong WF, who has always been available to listen and talk. We grew up, experienced, shared and learnt together, and I appreciated very much the ten years she were back here in HK when our friendship blossomed.

Last but not least, I would like to thank my family, my dearest hubby Yip Wang, and my two little angels Wing Hei and Wing Long, for their love and understandings, and for giving me a loving and cozy home which I cherish the most.

CONTENTS

Declaration	vii
Acknowledgements	ix
Table of Contents	xi
Lists of Figures	xv
Lists of Tables	xix
1 INTRODUCTION	1
1.1 Use of bounding volumes in collision detection	1
1.2 Collision detection of ellipsoids	2
1.3 Continuous collision detection	4
1.4 Collision detection of conics in the plane	5
1.5 CCD of quadrics	6
1.6 Contributions	7
2 ELLIPSES MOVING IN THE PLANE	9
2.1 Condition on separation of two ellipses	10
2.2 Separation condition for two moving elliptic disks	16
2.3 Outline of algorithm	17
2.4 Non-rational motions	19
2.5 Rational motions	21
2.5.1 Planar rational Euclidean motions	22
2.5.2 Properties of $\Delta(t)$	24
2.5.3 Robust computation	26
2.5.4 Experimental results	28
2.6 Possible future work	33
3 ELLIPSES MOVING IN THE SPACE	35
3.1 Reduction from 3D to 1D	35
3.2 Condition for separation of 1D ellipses	36

3.3	The algorithm for detecting collision of two moving ellipses in the 3D space	39
4	STATIC ELLIPSOIDS	43
4.1	Detecting overlap between stationary ellipsoids	43
4.1.1	Characteristic polynomial	45
4.1.2	Computational cost	46
4.1.3	Coefficients of the characteristic polynomial	48
4.1.4	Contact point of two touching ellipsoids	50
4.2	Constructing a separating plane	50
4.3	The complete algorithm	55
4.4	Experimental results	56
4.5	An alternative solution by subresultant sequence	59
4.5.1	Conditions for separation and external touching	60
4.5.2	Differentiating the separation and touching cases	62
4.5.3	A summary	63
4.5.4	Computation costs	64
5	MOVING ELLIPSOIDS WITH CONTINUOUS MOTION	67
5.1	3D rational Euclidean and affine motions	68
5.2	An affine motion interpolant	69
5.3	CCD equation for moving ellipsoids	71
5.3.1	Solving the CCD equation	74
5.3.2	Finding the first contact time only	82
5.3.3	Quadratic convergence of recursive Bézier shoot	82
5.4	Experimental results	82
5.4.1	Test in human character animation	83
5.4.2	Test in robotic arm movements	87
5.4.3	Two further examples	89
5.5	Discussion	89
6	COMPOSITE QUADRIC MODELS	91
6.1	Key ideas	92
6.2	The algorithm	96

6.2.1	Subproblems identification	97
6.2.2	Contact computation	98
6.3	Issues for CCD of general quadrics	102
6.3.1	Contact configuration	102
6.3.2	Tangency occurring on real intersection curves	106
6.4	Two working examples	113
6.5	CCD of CQMs with general boundary curves	120
7	CONCLUSION	121
	BIBLIOGRAPHY	123

LIST OF FIGURES

- Figure 1 A dinopet is tightly bounded by a set of 33 bounding ellipsoids, which are automatically generated using the algorithm in [40]. 3
- Figure 2 A virtual human character in (a) is bounded by the same number of boxes and ellipsoids. Total volume of bounding ellipsoids in (c) is 35% less than that of bounding boxes in (b). 3
- Figure 3 Two elliptic disks \mathcal{A} and \mathcal{B} are (a) separate; (b) overlapping; and (c) touching. 11
- Figure 4 Configuration for Lemma 2.2. 13
- Figure 5 Configuration for Lemma 2.3. 13
- Figure 6 Two elliptic disks and their characteristic polynomial $f(\lambda)$. Left: Overlapping *if and only if* $f(\lambda) = 0$ has no negative root. Middle: Touching externally *if and only if* $f(\lambda) = 0$ has a double negative root. Right: Separate *if and only if* $f(\lambda) = 0$ has two distinct negative roots. 15
- Figure 7 The two moving ellipses of Example 1, progressing from left to right, and the discriminant $\Delta(t)$. 22
- Figure 8 Two elliptic disks with translational motions, and their corresponding discriminant function. 25
- Figure 9 The eight real roots of $\Delta(t) = 0$ for a translational motion, and the corresponding contact points between the two elliptic disks. Note the double roots at t_2 and t_3 . 26

- Figure 10 Average CPU time needed for *CD-DISC* to detect collision for two moving elliptic disks with different degrees of motion, when the two moving elliptic disks collide (a) or are collision-free (b). The solid lines show the total collision detection time for the three variants of *CD-DISC* (see Section 2.5.3) and the dashed lines show the computation time taken for obtaining $\Delta(t)$ in the Bernstein form and in the scaled Bernstein form. 29
- Figure 11 The two moving elliptic disks in Example 2 and their discriminant $\Delta(t)$. 31
- Figure 12 The three configurations of two touching ellipses in 3D. Sub-figures (a), (b), (c) correspond to the three conditions (1), (2), (3), of Theorem 3.1, respectively. 41
- Figure 13 Two (a) disjoint; (b) overlapping ellipsoids and the corresponding $f(\lambda)$. 45
- Figure 14 The tetrahedron $[V_0 V_1 V_2 V_3]$. 52
- Figure 15 (a) Four planes passing through V_0 and V_1 : \mathcal{T}_L^A and \mathcal{T}_R^A are tangent to ellipsoid \mathcal{A} ; \mathcal{T}_L^B and \mathcal{T}_R^B are tangent to ellipsoid \mathcal{B} ; (b) A separating plane is one that passes through V_0 , V_1 and S , where S can be any point on the open line segment $(P_R^A P_L^B)$. 53
- Figure 16 Algorithm for collision detection between two ellipsoids \mathcal{A}_i and \mathcal{B}_i at frame i . 55
- Figure 17 Experimental set-up corresponding to the results in Table 4: a sphere $\mathcal{A}(t)$ moves along a circular orbit around an ellipsoid \mathcal{B} . Bright spheres indicate collision. Framed boxes bounding $\mathcal{A}(t)$ are disjoint from the bounding box of \mathcal{B} . 58
- Figure 18 (a) A CCD function for two collision-free ellipsoids; (b) A CCD function for two colliding ellipsoids. The blue and yellow regions are where $F(u, t) > 0$ and $F(u, t) < 0$, respectively. The zero set $F(u, t) = 0$ is given by the dark blue curve. The red points in (b) represent the moments when the ellipsoids are in external contact. 72

- Figure 19 Examples of intervals classified as (a) a candidate separation interval (CSI); (b) a candidate overlapping interval (COI); and (c) mixed intervals with different collision status at t_1 and t_2 will be divided at a contact instant t^* and each of the two subintervals being classified as either a CSI or a COI. The circled numbers are the collision status at a particular time instant (+1 for separation, 0 for external contact and -1 for overlapping). 76
- Figure 20 A Bézier shoot operation. (a) $F(\hat{u}, t) = 0$ has no real root in $[t_1, t_2]$; (b) \hat{t} is the smallest root of $F(\hat{u}, t) = 0$ in $[t_1, t_2]$. 78
- Figure 21 The handling of a candidate separation interval (CSI) in the algorithm for solving a CCD. 78
- Figure 22 The handling of a candidate overlapping interval (COI) in the algorithm for solving a CCD. 80
- Figure 23 Quadratic convergence of recursive Bézier shoot. 83
- Figure 24 Real-time continuous collision detection in a boxing game. Ellipsoids in collision are highlighted in red. 84
- Figure 25 (a) An F3 robotic arm and an I-shaped obstacle; (b) the bounding ellipsoids; (c), (d) & (e) the robotic arm in motion with $t = 0.104, 0.311$ and 0.778 , respectively, and the colliding ellipsoids are shown in red. 87
- Figure 26 Two moving ellipsoids under degree-4 dependent motion with affine deformations, the CCD equation $F(u, t) = 0$ is of degree 48 in t and four overlapping intervals are detected. 90
- Figure 27 Collision detection of an ellipsoid and a CSG object. 91
- Figure 28 CCD subproblems for two capped cylinders. 94
- Figure 29 Contact verification for CSG objects. Given two objects u and v , and the result w of applying a boolean operation to u and v , the boundary and interior of w , denoted by ∂w and $\overset{\circ}{w}$, respectively, can be determined from $\partial u, \overset{\circ}{u}, \partial v$ and $\overset{\circ}{v}$ accordingly. 96
- Figure 30 Two CQM objects. 97

Figure 31	Tangency of quadrics with different intersection configuration in the neighbourhood. (a) Touching point lying on the real intersection curves of the two quadrics. (b) There is no intersection locally at the touching point between the quadrics. Only (b) will be considered a <i>genuine touching point</i> for CCD of two quadrics. 110
Figure 32	CCD of two capped cylinders. 114
Figure 33	First contact of the two cylinders in example 6.4. 117
Figure 34	CCD of two general CQMs. 118
Figure 35	First contact of the two CQMs in example 6.5. 119

LIST OF TABLES

Table 1	Degrees of various entities for rational motions of different degrees. The last row shows the maximum degrees of the entities for a general motion of degree k . The motion $M(t)$ takes the form of equation (2.3) 25
Table 2	Configuration of two 1D ellipses and the roots of their characteristic equation. Ellipses are represented in pairs of brackets of the same style. Real intervals are represented by solid brackets, while imaginary intervals are represented by dotted brackets. Degenerate intervals of one point is represented by a dot or a cross. 40
Table 3	Experimental results demonstrating the effect of using the separating plane in EECD. 57
Table 4	Computation time for our algorithm, EECD, and for the enhanced GJK method, with a varying number of vertices (n) used in the latter for polyhedral approximation of the ellipsoids. 59
Table 5	Average CPU time taken for CCD of two virtual human characters in a boxing animation. 86
Table 6	Average CPU time taken for CCD of a robotic arm and an obstacle. 88
Table 7	Classification of QSIC that contain non-planar components in $\mathbb{P}R^3$ 107
Table 8	Classification of QSIC with only planar components in $\mathbb{P}R^3$ - Part I 108
Table 9	Classification of QSIC with only planar components in $\mathbb{P}R^3$ - Part II 109

Table 10 Cases 11, 21, 22 and 27 of the QSIC classification that need special handling for determining whether a contact point is genuine or not. [112](#)

INTRODUCTION

Motion design, analysis, and planning are important research topics that furnish a common scientific base to diverse engineering disciplines such as robotics, CAD/CAM, computer animation, simulation of virtual environments and 3D computer games [13, 21, 25, 47, 55]. For the simulation of realistic dynamical motions, rigid objects should not penetrate each other; and when they collide, impulsive response needs to be handled properly. Real-time collision detection is also crucial to physics engines for 3D computer games and simulation of virtual environments [14].

Quadric surfaces are widely used in practice. Ellipsoids and capped cylinders or cones are often used as bounding volumes of complex geometry in graphics and robotics. The rising interests in modeling human characters in computer game/animation and organic forms in bio-science make ellipsoids a serious contender for shape representation. Furthermore, most mechanical parts can accurately be modeled with simple quadrics, such as spheres, cones, and cylinders. Through composite representation or CSG composition, an even wider class of complex objects are modeled by quadric surfaces.

1.1 USE OF BOUNDING VOLUMES IN COLLISION DETECTION

Collision detection for general freeform moving objects is computationally very expensive. A two-phase approach to collision detection is widely adopted in practice. Objects are enclosed by simple geometric entities, called *bounding volumes*, to which simpler collision detection is first applied; more complicated collision detection com-

putation on the detailed objects will only be carried out if their bounding volumes are found to be overlapping. There are, in general, two criteria in choosing the type of bounding volumes to be used in a particular application. The first is *bounding tightness*: bounding volumes should be as tight as possible so that when two enclosed volumes are separate, their bounding volumes should also be separate for most of the time. This criterion saves time by ensuring that many non-colliding pairs are not subject to further processing once their bounding volumes are found to be separate. The second criterion is that collision detection for a pair of bounding volumes should be simple and very fast, since this operation usually needs to be done many times, i.e. for every pair of objects present in an environment.

Simple geometric primitives such as spheres [25], axis-aligned bounding boxes [23, 67], oriented bounding boxes [21], and discrete oriented polytopes [33] have been widely used as bounding volumes for collision detection. Due to its simplicity and superior capability of shape approximation, the ellipsoid is used as the bounding volume for robotic arms and convex polyhedra for collision detection [28, 53, 61, 75]. Bischoff and Kobbelt [5] used a set of overlapping ellipsoids for a compact, robust, and level-of-detail representation of 3D objects defined as polygonal meshes. Hyun et al. [26] showed that sweeps of ellipsoids fit tightly to human arms and legs. To obtain a set of tight bounding ellipsoids for a 3D object (Figure 1), Lu et al. [40] adopted a variational approach to compute an optimal segmentation based on an affine invariant measure. Figure 2(c) shows a human character bounded by the same number of ellipsoids as that of boxes in Figure 2(b). The total volume of the bounding ellipsoids in Figure 2(c) is about 35% less than the total volume of bounding boxes in Figure 2(b), meaning that ellipsoid-based collision detection for human characters or other similar natural object is more accurate than box-based collision detection. Thus ellipsoids have much potential as a bounding volume for 3D freeform objects.

1.2 COLLISION DETECTION OF ELLIPSOIDS

Among all the quadric surfaces, the ellipsoid deserve its own right of studying since it is the only member which is close and compact, and therefore is readily used as

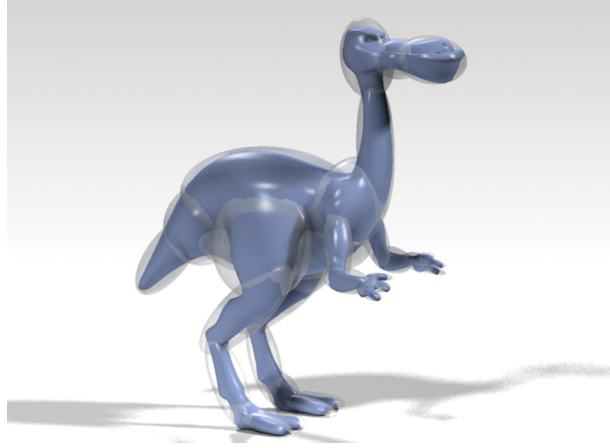


Figure 1: A dinopet is tightly bounded by a set of 33 bounding ellipsoids, which are automatically generated using the algorithm in [40].

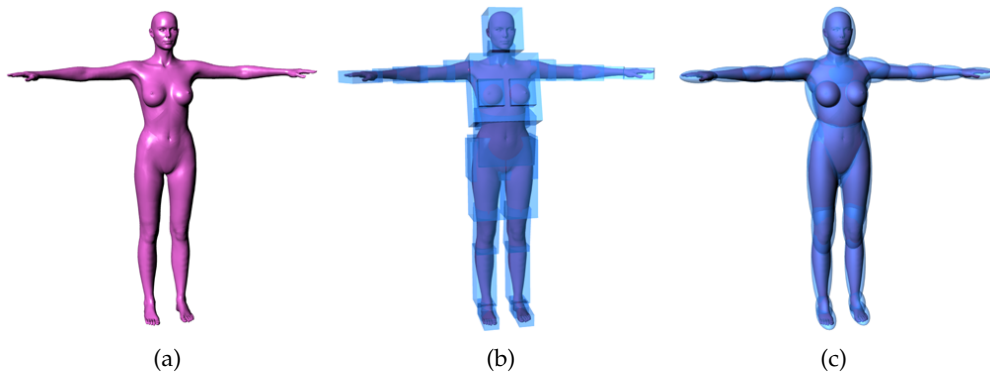


Figure 2: A virtual human character in (a) is bounded by the same number of boxes and ellipsoids. Total volume of bounding ellipsoids in (c) is 35% less than that of bounding boxes in (b).

bounding volume for collision detection. In fact, some objects can easily be modelled by ellipsoids or are inherently ellipsoidal in shape.

In the past, collision detection for ellipsoids was usually performed by faceting, and then applying a collision package appropriate to general convex polyhedra, such as GJK [20], I-COLLIDE [9], or V-Clip [43]. A drawback with this approach is that accuracy and efficiency are compromised by polyhedral approximation.

Rimon and Boyd [53] present a numerical technique for computing the *quasi-distance*, called *margin*, between two separate ellipsoids. Sohn et al. [62] compute the distance between two ellipsoids using line geometry. Using the Lagrange conditions, Lennerz and Schömer [35] present an algebraic algorithm for computing the distance between

two quadrics. Distance computation is a more difficult problem than collision detection since the latter can be solved as a subproblem of the former—a positive distance between two objects implies no collision between the two.

Ellipsoids are also used to represent the shapes of soil particles in geo-mechanics and the iso-potential surface of a molecule in computational physics [12]. The overlap test for ellipsoids is of high interest in these fields [39, 48]; however, these methods are mainly based on numerical iterations, leaving much to be desired for efficiency. Moreover, they are limited to the case of stationary ellipsoids. The algorithms described by Baraff [2] and Gilbert and Foo [19] also belong to this category, but are applicable to a wider class of objects bounded by smooth surfaces. In the field of astronautics, ellipsoids are used to represent threat volumes of space objects to determine possible close approach events [10].

1.3 CONTINUOUS COLLISION DETECTION

A typical framework for collision detection between moving objects is to sample the time interval of the motion at discrete time instants and test whether the objects intersect at each sampled instant. This temporal sampling approach is prone to error, since it may miss collisions that occur between sample instants. To this end, continuous collision detection (CCD) is currently an active research direction, in which collision status (i.e., either collision or collision-free) is guaranteed within a continuous time span.

Schwarzer et al. [56] uses adaptive sampling which guarantees that all configurations along a straight line segment connecting two configurations in the C-space are collision-free, by considering also the distance information in the workspace. Based on this technique, Ferré and Laumond [18] further developed a collision-free planar path through an iterative process. There have also been attempts to use the speed of moving objects as a bound to determine the safe time sampling resolution [8].

Redon et al. [50, 51, 52], Govindaraju et al. [22], and Kim et al. [32] consider CCD in various simulation environments, comprising of hundreds of thousands of polygons as obstacles and complex moving objects composed of articulated links. They develop effi-

cient algorithms of interactive speed for CCD while employing effective computational tools for culling redundant geometry at various stages of computation. Redon et al. [50] use Oriented Bounding Box (OBB) as the basic bounding volume, whereas Redon et al. [51, 52] and Kim et al. [32] employ Line Swept Sphere (LSS). These methods take geometric approaches in culling redundant geometry. In particular, Redon et al. [51, 52] and Kim et al. [32] apply a GPU-based collision detection to the swept volumes of LSS primitives against the environment; moreover, Govindaraju et al. [22] present a GPU-based algorithm that can also deal with deformable models. Zhang et al. [76], on the other hand, deals with the CCD of articulated models with the approach of conservative advancement that repeatedly moves objects by a computed time step while ensuring non-collision. Significant performance gain is achieved by using Taylor models to construct dynamic bounding volume hierarchies of the articulated models.

1.4 COLLISION DETECTION OF CONICS IN THE PLANE

Many collision detection algorithms are catered for 3D applications (see the surveys [27, 38]); however, there are numerous other applications in which objects only move in the plane. Examples are robot or vehicle path planning, where the robots or vehicles are represented by 2D figures and move in the 2D plane, but the robots may follow any paths and make arbitrary motions.

Even in a 2D setting, the outline of an object can be quite complicated. As in the 3D case, it is common to enclose objects by simple geometric entities, called *bounding objects* for fast and preliminary collision tests. Commonly used bounding objects in the plane include circular disks and rectangles.

Considering the criteria for bounding objects as described in Section 1.1 above, it is not hard to understand why circular discs are commonly used as bounding objects for robots in the plane (e.g. [46])—collision detection between a pair of circles is almost trivial, and can therefore be performed very efficiently. Interference testing of multiple circular discs has also been studied intensively in computational geometry [49, 59, 60]. (Spheres are popular bounding objects in the 3D case, for similar reasons.)

Ellipses provide much tighter bounding than circles. When ellipses or circles are used as bounding objects, far fewer ellipses than circles are normally needed to enclose a given object with the same degree of tightness. Therefore the use of ellipses as bounding objects can potentially lead to significant improvement in the accuracy and efficiency of collision detection. However, relatively little work can be found in the literature on the use of ellipses as bounding objects, largely because of the lack of effective means of collision detection for ellipses.

1.5 CCD OF QUADRICS

Regardless of all the advantages and popularity of using quadrics in various applications as we have seen, real-time CCD of quadrics has not been addressed in the literature. Thorough analysis and classification of intersection of general quadrics can be found in classical algebraic geometry e.g. [4, 7, 58]. and CAGD [69]. The Segre characteristics, defined by the elementary divisors of the matrix $\lambda A + B$, are used in algebraic geometry [7, 63] to classify a degenerate intersection curve between two quadric surfaces in complex projective space. Similar work in real projective space is presented by Tu et al. [65]. These results, however, consider quadrics in the complex (projective, affine or Euclidean) space, and are not applicable to the collision detection problem, for which the analysis must be done in the real affine domain: two quadric surfaces always intersect in complex projective space, but this does not mean that they share any common points in real affine space. There is nevertheless an obvious way to detect intersection between stationary quadrics, which is to compute their real intersection curves. Various algorithms have also been studied in CAGD for computing the intersection curve of two quadric surfaces (e.g. [36, 41, 71, 72, 74]). The objectives of these algorithms are to classify the topological or geometric structure of the intersection curve and to derive its parametric representation; efficient collision detection is not their primary purpose. Moreover, these methods are difficult to extend to deal with moving quadrics.

Because swept volumes and distances (which are the common approach to solving CCD) are difficult to compute for quadrics, an algebraic approach seems more suit-

able for the CCD of quadrics. Research in surface-surface-intersection of quadrics, which is closely related to the problem of collision detection, also suggests that the algebraic treatment is a natural approach for quadric surfaces—geometric approaches usually produce efficient intersection algorithms only for a limited class of simple quadrics, called *natural quadrics* (i.e., spheres, circular cylinders and cones) [31, 42], while algebraic techniques are capable of handling general quadrics [65, 71, 72].

1.6 CONTRIBUTIONS

This thesis is a collection of work focuses on the study of continuous collision detection of quadric surfaces. We aim to produce efficient and exact collision detection results, which are much desired in many applications where accurate shape is encoded more efficiently using quadric surfaces than using piecewise linear representation. Our approach is algebraic, therefore it requires *neither* geometric approximation of curved boundary surfaces *nor* discrete sampling of motion time interval, which are the main sources of inefficiency and inaccuracy of the conventional collision detection paradigm. We develop an efficient and robust algebraic or numerical solution that achieves real-time performance. Our algebraic approach leads to accurate solutions to the CCD problem for moving quadric surfaces under rational Euclidean or affine motions. In particular, we highlight the following contributions:

- We establish a simple condition for checking the separation of two ellipses in the real plane based on the number of the real roots of their characteristic equation, and apply it to collision detection of two moving ellipses (Chapter 2).
- We reduce CCD of ellipses moving in 3D to a 1D collision detection problem in a line, which is based on a thorough algebraic analysis (Chapter 3).
- Based on the separation condition for two static ellipsoids, we devise an optimized collision detection algorithm with a minimal number of arithmetic operations (Chapter 4).
- For ellipsoids moving with on-the-fly motions, we establish a simple method to construct a separating plane for two disjoint ellipsoids, and further develop a

collision detection algorithm exploiting inter-frame coherence of the ellipsoids using separating planes (Chapter 4).

- When the motion of the ellipsoids are pre-specified, we formulate the CCD problem algebraically in terms of a bivariate function, which is then solved using an efficient and robust numerical scheme (Chapter 5).
- Due to the projective invariant properties of our formulation, the above CCD methods for ellipses and ellipsoids can be extended readily to other conics and quadrics. A procedure is therefore devised to handle CCD of Composite Quadrics Models whose boundary are made up of piecewise conics edges or quadrics surface elements (Chapter 6).
- Our algorithms work not only for Euclidean motions but also for affine motions, meaning that the moving objects may change their shapes under affine transformations. This facility can be a potential advantage over the traditional methods when adapting our method to collision detection for deformable moving objects, such as human or animal bodies.

ELLIPSES MOVING IN THE PLANE

We shall present in this chapter a fast and accurate algorithm for continuous collision detection between two moving ellipses in the plane. We introduce new conditions on the separation of two ellipses to reduce the collision detection problem to the problem of detecting a real zero of a univariate function which is the discriminant of the characteristic polynomial of the two ellipses. If the ellipses serve as bounding objects, the colliding time intervals computed by our method can then be used as a refined time span to which other algorithms for collision detection on the exact objects may be applied.

Our method is based on theoretical results similar to those of Wang et al. in [70] concerning the separation of two stationary ellipsoids in 3D space, but there are important differences between these results for ellipses and ellipsoids. First, the separation condition for two stationary ellipses cannot be derived as a special case of the result [70] for two stationary ellipsoids, although the former is a low-dimensional counterpart of the latter. Therefore, here we shall prove for the first time an algebraic condition on the separation of two stationary ellipses. Second, compared with ellipsoids, the characteristic polynomial of two ellipses has relatively simple properties, and this simplicity allows us to reduce collision detection in the moving case to a problem of detecting the zero of a univariate function. In general, such a treatment is not possible for two moving ellipsoids, at least not in the same straightforward manner, as will be discussed in detail later in Chapter 5.

The contributions of the work in this chapter is summarized as follows:

1. A simple algebraic condition is established for the separation of two stationary ellipses.
2. An algebraic condition is established for detecting collisions between two moving ellipses.
3. An algorithmic framework for fast and accurate collision detection between two moving ellipses. We discuss in detail two classes of commonly used motions: cycloidal motions and rational motions. We also present robust methods for processing high-degree polynomials arising from the use of rational motions, which contribute to reliable collision detection.

2.1 CONDITION ON SEPARATION OF TWO ELLIPSES

In this section we are going to prove the separation condition for two stationary ellipses. An ellipse is a conic section curve, and can be represented in the Euclidean plane \mathbb{E}^2 by $X^T A X = 0$, where $A = [a_{i,j}]$ is a 3×3 real symmetric matrix, and X is a 3D column vector containing the homogeneous coordinates of a point in \mathbb{E}^2 . Let $A_{i,i}$ denote the leading submatrix of size $i \times i$ of A , $i = 1, 2, 3$. For an ellipse $X^T A X = 0$ we shall assume throughout that the matrix A is normalized such that $\bar{X}^T A \bar{X} < 0$ for any interior point \bar{X} of the ellipse. Then, by elementary geometry, an ellipse $X^T A X = 0$ is characterized by the conditions that $\det(A_{1,1}) = a_{1,1} > 0$, $\det(A_{2,2}) > 0$, and $\det(A_{3,3}) < 0$. Thus, $A_{2,2}$ is positive definite.

An *elliptic disk* \mathcal{A} is defined by $\mathcal{A} \equiv \{X | X^T A X \leq 0\} \subset \mathbb{E}^2$. We use $\partial \mathcal{A}$ to denote the boundary curve of \mathcal{A} , i.e. the set of point satisfying $X^T A X = 0$, and use $\text{Int}(\mathcal{A})$ to denote the interior points of \mathcal{A} . Thus, $\mathcal{A} = \partial \mathcal{A} \cup \text{Int}(\mathcal{A})$. For brevity, we will use the terms *ellipse* and *elliptic disk* interchangeably when there is no danger of confusion.

Two elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$ are said to be *separate* or *disjoint* if $\mathcal{A} \cap \mathcal{B} = \emptyset$. The disks \mathcal{A} and \mathcal{B} are said to be *overlapping* if $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) \neq \emptyset$; and they are said to be *touching* if $\mathcal{A} \cap \mathcal{B} \neq \emptyset$ and $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) = \emptyset$ (see Fig. 3).

Given two elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$, the cubic polynomial $f(\lambda) = \det(\lambda A - B)$ is called the *characteristic polynomial* and $f(\lambda) = 0$ the *characteristic equation* of \mathcal{A} and \mathcal{B} .

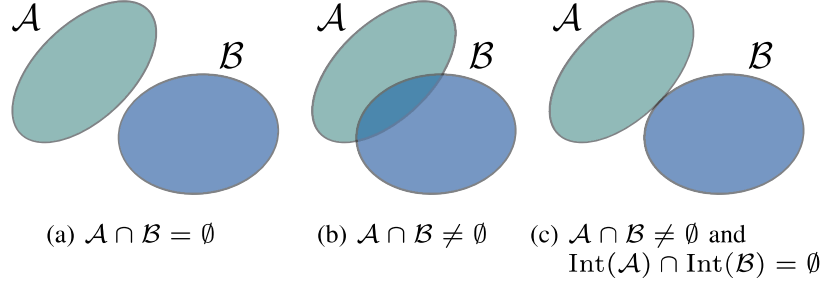


Figure 3: Two elliptic disks \mathcal{A} and \mathcal{B} are (a) separate; (b) overlapping; and (c) touching.

Note that $\lambda A - B$ represents the pencil of the two conics \mathcal{A} and \mathcal{B} , which contains at most three singular conics (if not consists exclusively of singular conics) given by $\lambda_0 A - B$ where λ_0 is a root of $f(\lambda)$ [37].

Lemma 2.1. *For any two elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$, the root pattern of $f(\lambda) = 0$ falls into one of the following three cases:*

1. *three positive roots; or*
2. *one positive and two negative roots; or*
3. *one positive and a pair of complex conjugate roots.*

Proof. Suppose

$$f(\lambda) = a_3 \lambda^3 + a_2 \lambda^2 + a_1 \lambda + a_0.$$

Then $a_3 = \det(A) < 0$ and $a_0 = -\det(B) > 0$. It follows that $f(0) > 0$ and $f(+\infty) < 0$. Hence, $f(\lambda) = 0$ has at least one positive root. Moreover, since $a_3 \neq 0$ and $a_0 \neq 0$, it is clear that 0 or ∞ cannot be a root of $f(\lambda) = 0$. Let $\lambda_0 > 0$, λ_1 and λ_2 denote the three roots. Since $\lambda_0 \lambda_1 \lambda_2 = -a_0/a_3 > 0$, we have $\lambda_1 \lambda_2 > 0$. Hence the other two roots λ_1 and λ_2 must both be positive, both negative, or a pair of complex conjugates. \square

Lemma 2.2. *If $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) = \emptyset$, then $f(\lambda) = 0$ has a negative root.*

Proof. Since $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) = \emptyset$, we may suppose that \mathcal{A} and \mathcal{B} are either separate or touching externally. We make the substitution $\lambda = (\mu - 1)/\mu$, which maps $\mu \in [0, 1]$ to $\lambda \in (-\infty, 0]$, and transforms the characteristic equation $f(\lambda) = \det(\lambda A - B) = 0$ to $g(\mu) \equiv \det((1 - \mu)A + \mu B) = 0$. We will denote $Q(\mu) \equiv (1 - \mu)A + \mu B$, observing

that $Q(0) = A$ and $Q(1) = B$. Clearly, $f(\lambda) = 0$ has a finite negative root if and only if $g(\mu) = 0$ has a real root in $(0, 1)$. We shall now show by contradiction that $g(\mu) = 0$ has a real root in $(0, 1)$.

Assume that $g(\mu) = 0$ has no real root in $(0, 1)$. Since $g(\mu) \equiv \det((1 - \mu)A + \mu B)$ is a continuous function of μ , and $g(0) = \det(A) < 0$, we have $g(\mu) = \det((1 - \mu)A + \mu B) < 0$ for all $\mu \in [0, 1]$. (Recall that $g(1) = \det(B) \neq 0$.) Clearly, $\det(Q(\mu)_{1,1}) = (1 - \mu)a_{1,1} + \mu b_{1,1} > 0$ for all $\mu \in [0, 1]$, since $A_{1,1} > 0$ and $B_{1,1} > 0$; furthermore, $Q(\mu)_{2,2} = (1 - \mu)A_{2,2} + \mu B_{2,2}$ is positive definite for any $\mu \in [0, 1]$, since $A_{2,2}$ and $B_{2,2}$ are positive definite. Thus $\det(Q(\mu)_{2,2}) > 0$ for all $\mu \in [0, 1]$. Hence, $X^T Q(\mu) X = 0$ is an ellipse for all $\mu \in [0, 1]$, with its center at $R(\mu) = Q(\mu)^{-1}[0, 0, 1]^T$.

Let us denote $p(\mu) \equiv R(\mu)^T A R(\mu)$. Then $p(\mu)$ is a continuous function of μ in $[0, 1]$. Clearly, $R(0) \in \text{Int}(\mathcal{A})$, since $R(0)$ is the center of \mathcal{A} . We have $R(1) \notin \mathcal{A}$ since $R(1)$ is the center of \mathcal{B} and $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) = \emptyset$, which is the hypothesis of the lemma. Hence, $p(0) = R(0)^T A R(0) < 0$ and $p(1) = R(1)^T A R(1) > 0$. By a continuity argument, it follows that $p(\mu_1) = 0$ for some $\mu_1 \in [0, 1]$, i.e. the center $R(\mu_1)$ of the ellipse $\mathcal{Q}(\mu_1)$ is on the boundary of the elliptic disk \mathcal{A} (see Fig. 4). We then define a circular disk \mathcal{D} centered at $R(\mu_1)$ which is contained in $\mathcal{Q}(\mu_1)$, i.e. $\mathcal{D} \subset \mathcal{Q}(\mu_1)$. Note that $R(\mu_1)$ may or may not be the tangent point of \mathcal{A} and \mathcal{B} , since \mathcal{A} and \mathcal{B} are either separate or touching externally. If $R(\mu_1)$ is not the tangent point, \mathcal{D} can be made sufficiently small such that $\mathcal{D} \cap \mathcal{B} = \emptyset$. In any case, the tangent of \mathcal{A} at $R(\mu_1)$ (also the common tangent of \mathcal{A} and \mathcal{B} at $R(\mu_1)$, if $R(\mu_1)$ is the tangent point) gives a diameter of the disk \mathcal{D} . We may then choose a point X_1 on this diameter, except for $R(\mu_1)$, such that X_1 is interior to $\mathcal{Q}(\mu_1)$, but is exterior to both \mathcal{A} and \mathcal{B} . Hence we have $X_1^T Q(\mu_1) X_1 < 0$, $X_1^T A X_1 > 0$, and $X_1^T B X_1 > 0$. Since $\mu_1 \in (0, 1)$, it follows that

$$X_1^T Q(\mu_1) X_1 = (1 - \mu_1) X_1^T A X_1 + \mu_1 X_1^T B X_1 > 0.$$

This is a contradiction. Hence, $g(\mu)$ has a real zero in $(0, 1)$. □

Lemma 2.3. *If $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) \neq \emptyset$, then any real root of $f(\lambda) = 0$ is positive.*

Proof. The proof goes by contradiction. Let λ_0 be a real root of $f(\lambda) = 0$. Assume $\lambda_0 \leq 0$ and denote $Q_0 = \lambda_0 A - B$. Then there exists a real point X_0 such that $Q_0 X_0 = 0$,

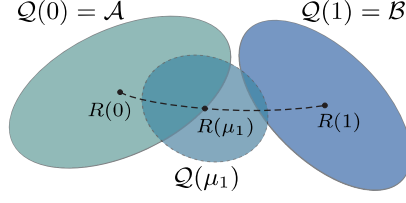


Figure 4: Configuration for Lemma 2.2.

because Q_0 is singular. Since $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) \neq \emptyset$, let $X_1 (\neq X_0)$ denote a common interior point of \mathcal{A} and \mathcal{B} , i.e. $X_1^T A X_1 < 0$ and $X_1^T B X_1 < 0$ (Fig. 5). Then

$$X_1^T Q_0 X_1 = \lambda_0 X_1^T A X_1 - X_1^T B X_1 > 0.$$

Let \mathcal{L} denote the line passing through X_0 and X_1 . Then, since \mathcal{A} and \mathcal{B} are bounded, there exists on the line \mathcal{L} a point \tilde{X} far enough from \mathcal{A} and \mathcal{B} such that \tilde{X} is exterior to both \mathcal{A} and \mathcal{B} . Let us write $\tilde{X} = \alpha X_0 + \beta X_1$, where α and β are real constants that are not both zero. Then $\tilde{X}^T A \tilde{X} > 0$ and $\tilde{X}^T B \tilde{X} > 0$. It follows that

$$\tilde{X}^T Q_0 \tilde{X} = \lambda_0 \tilde{X}^T A \tilde{X} - \tilde{X}^T B \tilde{X} < 0.$$

On the other hand, since $Q_0 X_0 = 0$ and $X_1^T Q_0 X_1 > 0$, we have

$$\begin{aligned} \tilde{X}^T Q_0 \tilde{X} &= \alpha^2 X_0^T Q_0 X_0 + 2\alpha\beta X_1^T Q_0 X_0 + \beta^2 X_1^T Q_0 X_1 \\ &= \beta^2 X_1^T Q_0 X_1 \geq 0. \end{aligned}$$

This is a contradiction. Hence, any real root λ_0 of $f(\lambda) = 0$ is positive. □

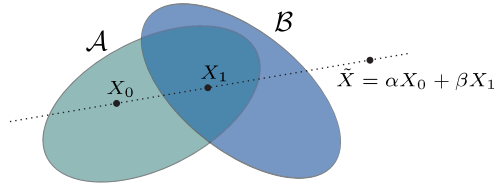


Figure 5: Configuration for Lemma 2.3.

Lemma 2.4. *If two elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$ touch externally, then $f(\lambda) = 0$ has a negative double root.*

Proof. Suppose that \mathcal{A} and \mathcal{B} do touch externally. As a result, the two ellipses $X^T A X = 0$ and $X^T B X = 0$ will have a multiple intersection. Then $f(\lambda) = 0$ has a multiple root λ_0 [37, page 256]. Since $\text{Int}(\mathcal{A}) \cap \text{Int}(\mathcal{B}) = \emptyset$, by Lemma 2.2, $f(\lambda) = 0$ has a negative root λ_1 . Moreover, by Lemma 2.1, $f(\lambda) = 0$ has a positive root λ_2 . Thus, we have either $\lambda_0 = \lambda_1 < 0$ or $\lambda_0 = \lambda_2 > 0$. Again by Lemma 2.1, only the first case is possible. Hence $f(\lambda) = 0$ has a negative double root. \square

Lemma 2.5. *If $f(\lambda) = 0$ has a negative double root, then the elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$ touch each other externally. Moreover, the real touching point X_0 is the unique solution to $(\lambda_0 A - B)X = 0$, where λ_0 is the negative double root of $f(\lambda) = 0$.*

Proof. Let $\lambda_0 < 0$ be a negative double root of $f(\lambda) = 0$. Clearly, λ_0 is not a zero of the first 2×2 minors $\det(\lambda A_{2,2} - B_{2,2})$, because both $A_{2,2}$ and $B_{2,2}$ are positive definite and thus the two zeros of $\det(\lambda A_{2,2} - B_{2,2})$ are positive. It follows that $\text{rank}(\lambda_0 A - B) = 2$ and its null space, $\text{Ker}[\lambda_0 A - B]$, has dimension 1.

Since $\det(\lambda A - B) = 0$ has a double root λ_0 , the pencil $X^T(\lambda A - B)X = 0$ contains the singular conic $\lambda_0 A - B$ with multiplicity 2. In this case, the two ellipses $X^T A X = 0$ and $X^T B X = 0$ are tangential to each other at the singular point X_0 of the conic $X^T(\lambda_0 A - B)X = 0$, that is, $(\lambda_0 A - B)X_0 = 0$ (see [4]).

We are now going to show that X_0 is a real tangent point of \mathcal{A} and \mathcal{B} . Let us suppose that $X_0 = U \pm iV \neq 0$, where U and V are real homogeneous vectors which are not both zero; without loss of generality, we suppose that $U \neq 0$. Then, from $(\lambda_0 A - B)(U \pm iV) = 0$, it follows that $(\lambda_0 A - B)U = 0$ and $(\lambda_0 A - B)V = 0$. This means that U and V are both real solutions of $(\lambda_0 A - B)X = 0$. Hence U and V are linearly dependent, or $V = \alpha U$ for some constant α , since $\text{Ker}[\lambda_0 A - B]$ has dimension 1. It follows that $X_0 = (1 + i\alpha)U$ is a real point, since U stands for a real point and the multiplicative constant $(1 + i\alpha)$ can be ignored in a homogeneous representation. Hence, the elliptic disks \mathcal{A} and \mathcal{B} touch each other externally at the real point X_0 . \square

Lemma 2.5 also suggests a convenient means to compute the contact point of two externally touching ellipses, which will be used to find the first contact point of two moving ellipses in subsequent sections.

The following theorem gives a condition on the separation of two elliptic disks, which is the main result of this section. Fig. 6 illustrates the relationship between two ellipses and the root pattern of their characteristic polynomial.

Theorem 2.6. Given two ellipses $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$,

1. \mathcal{A} and \mathcal{B} touch externally if and only if $f(\lambda) = 0$ has a negative double root;
2. \mathcal{A} and \mathcal{B} are separate if and only if $f(\lambda) = 0$ has two distinct negative roots.

Proof. Part (1) follows from Lemma 2.4 and Lemma 2.5. For part (2), the sufficiency follows from Lemma 2.3 and Lemma 2.4, and the necessity follows from Lemma 2.2 and Lemma 2.5. □

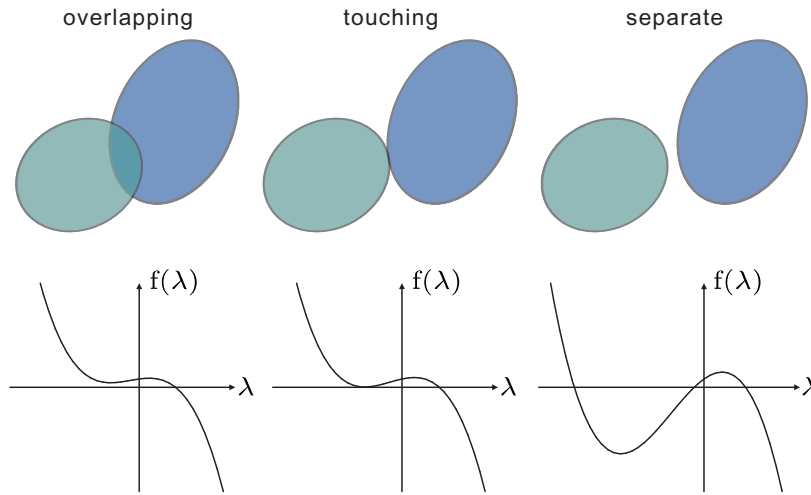


Figure 6: Two elliptic disks and their characteristic polynomial $f(\lambda)$. Left: Overlapping if and only if $f(\lambda) = 0$ has no negative root. Middle: Touching externally if and only if $f(\lambda) = 0$ has a double negative root. Right: Separate if and only if $f(\lambda) = 0$ has two distinct negative roots.

Remark 2.7. The application of the above conditions to detecting the overlapping of two stationary ellipses is rather straightforward. A description of the resulting algorithm will be discussed in Section 2.3 as the first step of our complete algorithm for moving ellipses. Note that a quick and exact test for overlap between two stationary ellipses should be of interest in its own right in some applications.

The next corollary, following from Theorem 2.6(2) and Lemma 2.1, is a key property that enables us to detect collisions between two moving elliptic disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$ by the occurrence of a double root of $f(\lambda; t) = 0$, as will be seen in the next section.

Corollary 2.8. *Suppose that two elliptic disks $\mathcal{A} : X^T A X \leq 0$ and $\mathcal{B} : X^T B X \leq 0$ are separate. Then $f(\lambda) = 0$ does not have any multiple root.*

2.2 SEPARATION CONDITION FOR TWO MOVING ELLIPTIC DISKS

In this section we are going to establish a condition for detecting a collision between two moving elliptic disks. Consider two elliptic disks $\mathcal{A}(t) : X^T A(t) X \leq 0$ and $\mathcal{B}(t) : X^T B(t) X \leq 0$ making continuous motions $M_A(t)$ and $M_B(t), t \in [0, 1]$ respectively. The disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are said to be *collision-free* if $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are separate for all $t \in [0, 1]$. Otherwise, $\mathcal{A}(t)$ and $\mathcal{B}(t)$ *collide*: i.e. $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are either touching or overlapping for some $t \in [0, 1]$.

The characteristic polynomial of $\mathcal{A}(t)$ and $\mathcal{B}(t), t \in [0, 1]$, is

$$f(\lambda; t) = \det(\lambda A(t) - B(t)),$$

and we can write

$$f(\lambda; t) = g_3(t)\lambda^3 + g_2(t)\lambda^2 + g_1(t)\lambda + g_0(t). \quad (2.1)$$

The discriminant of $f(\lambda; t)$ with respect to λ , as a function of t , is

$$\Delta(t) = 18g_3g_2g_1g_0 - 4g_2^3g_0 + g_2^2g_1^2 - 4g_3g_1^3 - 27g_3^2g_0^2 \quad (2.2)$$

(see [11]). By definition, $f(\lambda; t) = 0$ has a multiple root in λ for some t if and only if $\Delta(t) = 0$. Furthermore, it can be shown that $f(\lambda; t) = 0$ has three simple real roots if $\Delta(t) > 0$, and $f(\lambda; t) = 0$ has two complex conjugate roots and a real root if $\Delta(t) < 0$.

The next theorem states the condition that two moving elliptic disks are collision-free.

Theorem 2.9. *Let $\mathcal{A}(t)$ and $\mathcal{B}(t), t \in [0, 1]$, be two moving elliptic disks in \mathbb{E}^2 . Let $f(\lambda; t)$ be their characteristic polynomial. Let $\Delta(t)$ denote the discriminant of $f(\lambda; t)$ with respect to λ . Suppose that $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate. Then $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are collision-free if and only if $\Delta(t)$ has no real zero in $[0, 1]$.*

Proof. First we prove necessity. Suppose that $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are collision-free. Then by Corollary 2.8, $f(\lambda; t) = 0$ does not have a multiple root in λ for any $t \in [0, 1]$. Therefore, $\Delta(t)$ does not have any real zero in $[0, 1]$.

To prove sufficiency, suppose that $\Delta(t) = 0$ has no real root in $[0, 1]$. Now assume that $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide. Then $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ are overlapping or touching for some $t_0 \in [0, 1]$. Since $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate, by a continuity argument, there exists an instant $t_1 \in (0, t_0] \subset [0, 1]$ at which $\mathcal{A}(t_1)$ and $\mathcal{B}(t_1)$ touch each other externally. Then, by Theorem 2.6, $f(\lambda; t_1) = 0$ has a negative double root in λ . Therefore $\Delta(t_1) = 0$. But this contradicts the condition that $\Delta(t)$ has no zero in $[0, 1]$. Hence, $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are collision-free. \square

Corollary 2.10. *Let $\mathcal{A}(t)$ and $\mathcal{B}(t)$, $t \in [0, 1]$, be two moving elliptic disks. Suppose that $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate. If $\Delta(t)$ has a real zero in $[0, 1]$, then $\mathcal{A}(t)$ and $\mathcal{B}(t)$ touch each other externally at $t_{\min} \in [0, 1]$, where t_{\min} is the smallest real zero of $\Delta(t)$ in $[0, 1]$: i.e. $t_{\min} = \min\{t | \Delta(t) = 0, t \in [0, 1]\}$.*

The proof of Corollary 2.10 is similar to the proof of necessity in Theorem 2.9, and is therefore omitted. Here t_{\min} gives the time of first contact between the disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$.

2.3 OUTLINE OF ALGORITHM

Based on the separation conditions proved in the preceding sections, in this section we shall outline the framework of our algorithm for collision detection between two moving ellipses.

Algorithm: CD-DISC

Input: The matrices $A(t)$ and $B(t)$ of two moving elliptic disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$.

Output: Whether the two elliptic disks collide: COLLISION or COLLISION-FREE.

Step 1: Compute the characteristic equation $f(\lambda; 0) = 0$ for $\mathcal{A}(0)$ and $\mathcal{B}(0)$. Then determine whether $f(\lambda; 0) = 0$ has two distinct negative roots.

If yes, by Theorem 2.6, $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate, and go to Step 2; otherwise, report COLLISION and exit.

Step 2: Compute the characteristic polynomial

$$f(\lambda; t) = \det(\lambda A(t) - B(t)).$$

Step 3: Compute the discriminant $\Delta(t)$ of $f(\lambda; t)$ with respect to λ .

Step 4: Determine whether $\Delta(t) = 0$ has any real root in $[0, 1]$.

If yes, by Theorem 2.9, report COLLISION and exit; otherwise, again by Theorem 2.9, report COLLISION-FREE and exit.

In Step 1 of the algorithm *CD-DISC*, we use the Sturm sequence method [11, page 96], a classical real root isolation method, to check whether or not $f(\lambda; 0) = 0$ has two distinct negative roots. By Theorem 2.6, this can determine whether or not the two ellipses are separate at $t = 0$. The Sturm sequence method counts the number of real zeros of a polynomial within a specified interval by taking the difference between the numbers of sign changes exhibited by the Sturm sequence of the polynomial at the two ends of the interval (a multiple real root is counted once only). When applying the Sturm sequence method to $f(\lambda; 0)$ over the interval $(-\infty, 0)$, Lemma 2.1 ensures that the number of zeros can only be 0, 1, or 2, corresponding to situations in which whether $f(\lambda; 0) = 0$ has no negative root, one negative double root, or two distinct negative roots, respectively.

The algorithm *CD-DISC* only reports whether the two moving ellipses collide. By solving for the roots of $\Delta(t)$, this algorithm can be extended to report also the time of first contact, or all instants at which the ellipses are in external contact. By Corollary 2.10, the smallest root t_{\min} of $\Delta(t) = 0$ in $[0, 1]$ is always the instant of first contact between the two disks. However, to report all contact instants, the other roots of $\Delta(t) = 0$ in $[0, 1]$ need to be checked because, while all contact instants must be roots of $\Delta(t) = 0$, a root of $\Delta(t) = 0$ may not correspond to an external contact between the disks. We will discuss more about this in Section 2.5.2. For each external contact time instant t_i , the corresponding touching point of the two ellipses can be obtained by finding the unique solution to $(\lambda_i A(t_i) - B(t_i))X = 0$, where λ_i is the negative double root of $f(\lambda; t_i) = 0$ (Lemma 2.5). Implementation of the algorithm *CD-DISC*, with various enhancements for different types of motions and outputs, will be discussed in the following sections.

2.4 NON-RATIONAL MOTIONS

In the algorithm *CD-DISC*, it is necessary at least to detect the real roots of the univariate equation $\Delta(t) = 0$. When the motions of the ellipses are analytical but otherwise arbitrary, $\Delta(t) = 0$ is a rather general equation, and appropriate root-finding techniques need to be used. If the motions are piecewise analytical, then the algorithm can be applied to each piece. In the rest of the discussion we shall consider some special types of motion that are frequently encountered and which allow relatively easy formulation or efficient handling. In particular, we will consider the cycloidal motion in this section, and the rational motion in the next section.

The *cycloidal motion* is commonly used in cam design. An object with a cycloidal motion has the trajectories of all of its points being cycloids. The simplest cycloidal motion is a circle rolling along a straight line. We shall consider ellipses making cycloidal motions, such that they translate with constant velocities and, at the same time, rotate about their centers with a constant angular velocity. In this case, the elements of the motion matrix $M(t)$ contain not only rational functions of time t , but also trigonometric terms such as $\cos(\gamma_0 t + \varphi_0)$ and $\sin(\gamma_1 t + \varphi_1)$, for some constants $\gamma_0, \gamma_1, \varphi_0$ and φ_1 . Therefore, the coefficients $g_i(t)$ of the characteristic polynomial in (2.1) are not rational functions in t . It might be suggested that the trigonometric functions could be converted into rational functions using the variable substitution $u = \tan(t/2)$. However, this substitution would make the translational part, which is linear in t , non-rational. In fact, this kind of motion is intrinsically transcendental; hence, it can only be approximated, but not exactly represented, by a rational motion.

Suppose that two elliptic disks \mathcal{A} and \mathcal{B} perform cycloidal motions. Since the sizes and shapes of the disks do not change during the motion, the coefficients $g_3(t)$ and $g_0(t)$ of (2.1) are constant and equal to $\det(A)$ and $-\det(B)$ respectively. Let $u = (1-t)\theta_0 + t\theta_1$ and $v = (1-t)\phi_0 + t\phi_1$ be linear interpolations of the initial and

final orientation angles θ_0, θ_1 of \mathcal{A} and ϕ_0, ϕ_1 of \mathcal{B} . Then the other two coefficients $g_2(t)$ and $g_1(t)$ can be expressed as

$$\begin{aligned} g_2(t) &= (\alpha_{22} \cos(2v) + \alpha_{21} \sin(2v) + \alpha_{20})t^2 \\ &\quad + (\alpha_{12} \cos(2v) + \alpha_{11} \sin(2v) + \alpha_{10})t \\ &\quad + \alpha_{03} \cos(2(u-v)) + \alpha_{02} \cos(2v) + \alpha_{01} \sin(2v) + \alpha_{00} \end{aligned}$$

and

$$\begin{aligned} g_1(t) &= (\beta_{22} \cos(2u) + \beta_{21} \sin(2u) + \beta_{20})t^2 \\ &\quad + (\beta_{12} \cos(2u) + \beta_{11} \sin(2u) + \beta_{10})t \\ &\quad + \beta_{03} \cos(2(u-v)) + \beta_{02} \cos(2u) + \beta_{01} \sin(2u) + \beta_{00}, \end{aligned}$$

where the α and β terms are all constants.

Since the coefficients $g_i(t)$ are not rational, neither is the discriminant $\Delta(t)$ for a cycloidal motion. One may use any suitable numerical solver to compute the roots of $\Delta(t) = 0$ or to check for the existence of any real roots. The example below illustrates the steps of the algorithm *CD-DISC* for two ellipses making cycloidal motions.

Example 2.1. Consider two elliptic disks $\mathcal{A} : \frac{x^2}{6^2} + \frac{y^2}{10^2} \leq 1$ and $\mathcal{B} : \frac{x^2}{14^2} + \frac{y^2}{4^2} \leq 1$. Two moving elliptic disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$, $t \in [0, 1]$, are defined by the transformation of \mathcal{A} and \mathcal{B} under the following cycloidal motions:

$$\begin{aligned} M_A &= \begin{pmatrix} \cos(\frac{10\pi t}{9}) & -\sin(\frac{10\pi t}{9}) & 115t - 80 \\ \sin(\frac{10\pi t}{9}) & \cos(\frac{10\pi t}{9}) & 55t - 38 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{and} \\ M_B &= \begin{pmatrix} \cos(\frac{2\pi t}{3}) & -\sin(\frac{2\pi t}{3}) & 76t - 60 \\ \sin(\frac{2\pi t}{3}) & \cos(\frac{2\pi t}{3}) & 97t - 57 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

The characteristic polynomial is

$$\begin{aligned}
f(\lambda; t) &= \det(\lambda A(t) - B(t)) \\
&= -\frac{1}{3600}\lambda^3 + \left(\left(-\frac{243}{125440} \cos\left(\frac{4\pi t}{3}\right) - \frac{117}{4480} \sin\left(\frac{4\pi t}{3}\right) - \frac{3869}{125440} \right) t^2 \right. \\
&\quad + \left(\frac{9}{31360} \cos\left(\frac{4\pi t}{2}\right) + \frac{1581}{62720} \sin\left(\frac{4\pi t}{3}\right) + \frac{13939}{470400} \right) t + \frac{1}{1960} \cos\left(\frac{8\pi t}{9}\right) \\
&\quad + \frac{39}{125440} \cos\left(\frac{4\pi t}{3}\right) - \frac{19}{3136} \sin\left(\frac{4\pi t}{3}\right) - \frac{10519}{1881600} \left. \right) \lambda^2 \\
&\quad + \left(\left(-\frac{27}{39200} \cos\left(\frac{20\pi t}{9}\right) - \frac{13}{1400} \sin\left(\frac{20\pi t}{9}\right) + \frac{1241}{62720} \right) t^2 + \left(\frac{1}{9800} \cos\left(\frac{20\pi t}{9}\right) \right. \right. \\
&\quad + \frac{527}{58800} \sin\left(\frac{20\pi t}{9}\right) - \frac{4471}{235200} \left. \right) t - \frac{1}{1960} \cos\left(\frac{8\pi t}{9}\right) + \frac{13}{117600} \cos\left(\frac{20\pi t}{9}\right) \\
&\quad \left. - \frac{19}{8820} \sin\left(\frac{20\pi t}{9}\right) + \frac{937}{313600} \right) \lambda + \frac{1}{3136}.
\end{aligned}$$

The discriminant $\Delta(t)$ has a long expression, and is therefore omitted.

The disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$, moving from left to right, and the graph of their discriminant are shown in Fig. 7. Using Maple, with floating-point computations to 12 decimal places, the roots are found at $t = 0.226, 0.393, 0.600$ and 0.731 . Therefore, the two ellipses collide during the cycloidal motion and the first contact is at time $t = 0.226$ and the contact point is found to be at $(-47.605, -33.162)^T$. Furthermore, only $t = 0.731$ corresponds to another external touching of the two ellipses and the contact point is $(-2.469, 2.723)^T$. Comparing with the results obtained by Maple integer arithmetics (except for the last step of root solving using floating point computations), the accuracy of the roots found above is up to 9 decimal places.

2.5 RATIONAL MOTIONS

Studies on rational motions [24, 29, 30, 54], and in particular, on planar rational motions [68], have shown that low-degree rational motions are adequate to meet the need for motion design and representation in robotics and CAD/CAM. The use of rational motions also allows effective computation, using various well-developed techniques for processing polynomials. In this section we shall study in detail the application of our method to collision detection between two elliptic disks making rational motions, with particular emphasis on rational Euclidean motions. The resulting

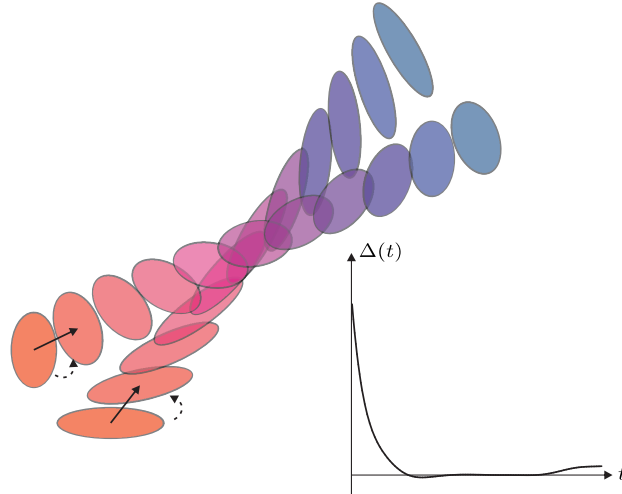


Figure 7: The two moving ellipses of Example 1, progressing from left to right, and the discriminant $\Delta(t)$.

algorithms are also applicable to affine motions that produce continuous deformation of the objects, an effect often required in computer animation.

2.5.1 Planar rational Euclidean motions

We start with a brief review of planar rational Euclidean motions. A Euclidean transformation in \mathbb{E}^2 is given by $X' = MX$, where

$$M = \rho \begin{pmatrix} R & V \\ \mathbf{0}^T & 1 \end{pmatrix}$$

for some nonzero constant ρ , and X, X' are points in \mathbb{E}^2 in homogeneous coordinates. The rotational part of the transformation is described by the 2×2 orthogonal matrix R and the translational part by the vector V . If the elements of R and V are continuous functions of t , then M describes a transformation over time and can therefore be denoted by $M(t)$. In particular, if the elements of $M(t)$ are rational functions and R is orthogonal for all t , then $M(t)$ is called a *rational Euclidean motion* whose degree is the maximal degree of its elements. (Note that $M(t)$ represents an affine motion only if $R(t)$ is non-singular.)

One way to construct a rational Euclidean motion is to use the kinematic mapping that associates the Euclidean transformation M with a point \mathbf{d} in \mathbb{P}^3 , the 3D real projective space, as described in [68]. If we write

$$R = \begin{pmatrix} \cos \phi & -\sin \phi \\ \sin \phi & \cos \phi \end{pmatrix} \text{ and } V = \begin{pmatrix} v_x \\ v_y \end{pmatrix},$$

then the kinematic image $\mathbf{d} \in \mathbb{P}^3$ of M is given by

$$\mathbf{d} = \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} v_x \sin(\phi/2) - v_y \cos(\phi/2) \\ v_x \cos(\phi/2) + v_y \sin(\phi/2) \\ -2 \cos(\phi/2) \\ 2 \sin(\phi/2) \end{pmatrix}.$$

Conversely, any point \mathbf{d} in \mathbb{P}^3 at which $d_2^2 + d_3^2 \neq 0$ corresponds to a Euclidean transformation M in \mathbb{E}^2 , given by

$$M = \begin{pmatrix} d_2^2 - d_3^2 & 2d_2d_3 & 2(d_0d_3 - d_1d_2) \\ -2d_2d_3 & d_2^2 - d_3^2 & 2(d_0d_2 + d_1d_3) \\ 0 & 0 & d_2^2 + d_3^2 \end{pmatrix}. \quad (2.3)$$

It follows that there is a one-to-one correspondence between a Euclidean transformation in \mathbb{E}^2 and a point in the kinematic image space, which is \mathbb{P}^3 with the line $d_2 = d_3 = 0$ removed. Due to this correspondence by means of a kinematic mapping, we may construct a polynomial curve in the kinematic image space and then obtain the corresponding rational Euclidean motion in \mathbb{E}^2 . In general, if the d_i terms are polynomials of degree n , the resulting motion will be of degree $2n$. A C^2 interpolation scheme of a set of given positions in \mathbb{E}^2 with piecewise quartic B-spline rational motions can be found [68]. Another advantage of rational motions is that they permit an algebraic treatment of the collision detection problem.

When applying a rational motion $M(t)$ to an ellipse $\mathcal{A} : X^T A X = 0$, we get a moving ellipse $\mathcal{A}(t) : X^T A(t) X = 0$ where

$$A(t) = (M^{-1}(t))^T A M^{-1}(t).$$

Inverting (2.3), we have

$$M^{-1} = \begin{pmatrix} d_2^2 - d_3^2 & -2d_2d_3 & 2(d_1d_2 + d_0d_3) \\ 2d_2d_3 & d_2^2 - d_3^2 & 2(d_1d_3 - d_0d_2) \\ 0 & 0 & d_2^2 + d_3^2 \end{pmatrix}.$$

Therefore the maximal degree of the entries in $A(t)$ is $2k$, if the degree of the motion $M(t)$ is k .

2.5.2 Properties of $\Delta(t)$

We now analyze the degree of the discriminant $\Delta(t)$. The characteristic equation $f(\lambda; t) = 0$ of the two moving elliptic disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$ is cubic in λ , and its degree in t depends on the degree of the rational motions of the two disks. Suppose that the motions $M_A(t)$ and $M_B(t)$ both have degree k . Then the maximum degree of the elements of $\lambda A(t) - B(t)$ is $2k$, and the maximum degree of the coefficients $g_i(t)$ of the characteristic equation is $6k$. Hence, from (2.2), the maximum degree of $\Delta(t)$ is $24k$. This analysis only gives an upper bound of the degree of $\Delta(t)$, because the actual degree of $\Delta(t)$ depends on the specific motions that are used. For example, consider a linear translational motion

$$M(t) = \begin{pmatrix} R(t) & V(t) \\ \mathbf{0}^T & 1 \end{pmatrix},$$

where the entries in $R(t)$ are all constants and those in $V(t)$ are linear polynomials. Then, the degree of g_0 and of g_3 is 0 while that of g_1 and of g_2 is 2. Therefore the degree of $\Delta(t)$ is only 8, which is much lower than 24, which is the degree that could be deduced from the general analysis. The relationship between the degree of $\Delta(t)$ and the degree of the rational motions is summarized in Table 1.

Now let us consider the geometric meaning of the roots of $\Delta(t)$. If two ellipses $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ touch each other externally, by Theorem 2.6, $f(\lambda, t_0) = 0$ has a negative double root, and we have $\Delta(t_0) = 0$. However, when $\Delta(\tilde{t}) = 0$ for some \tilde{t} , the disks $\mathcal{A}(\tilde{t})$ and $\mathcal{B}(\tilde{t})$ do not necessarily touch each other.

Table 1: Degrees of various entities for rational motions of different degrees. The last row shows the maximum degrees of the entities for a general motion of degree k . The motion $M(t)$ takes the form of equation (2.3)

Motion Type	Degree in t		
	$M(t)$	$g_i(t)$	$\Delta(t)$
Linear Translation	1	0 (g_0, g_3), 2 (g_1, g_2)	8
General Motion	k	$6k$	$24k$

Fig. 8 shows two elliptic disks moving with linear translational motions and the graph of their discriminant. Here, $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate. Note that the first real

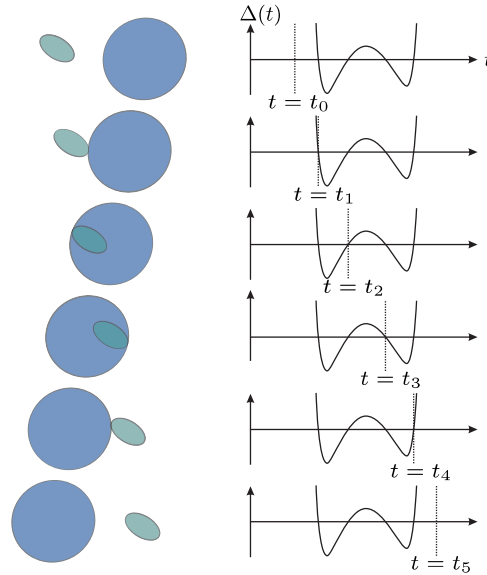


Figure 8: Two elliptic disks with translational motions, and their corresponding discriminant function.

root t_1 of $\Delta(t) = 0$ corresponds to an external contact between $\mathcal{A}(t)$ and $\mathcal{B}(t)$, while the next two roots, t_2 and t_3 , are caused by internal tangency of the two disks.

Since the degree of $\Delta(t)$ is 8 in the case of a linear translational motion, $\Delta(t) = 0$ can have eight real roots at most. Fig. 9 illustrates a case where all these eight real roots are accounted for by real tangencies between two elliptic disks during linear motions. Here, there are two instants (t_2 and t_3) when the two disks are internally tangential to each other simultaneously at two points; t_2 and t_3 are double zeros of $\Delta(t) = 0$.

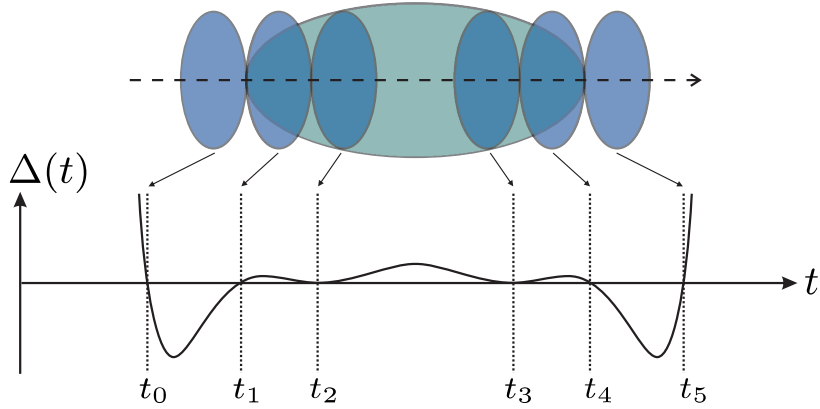


Figure 9: The eight real roots of $\Delta(t) = 0$ for a translational motion, and the corresponding contact points between the two elliptic disks. Note the double roots at t_2 and t_3 .

We have mentioned that a real zero of $\Delta(t) = 0$ may not correspond to any real tangency between the two ellipses. To see this, consider two moving circular disks that become two concentric circles $x^2 + y^2 = 1$ and $x^2 + y^2 = 4$ at time t_0 . It is easy to verify that the characteristic equation $f(\lambda; t_0) = 0$ has a positive double root and therefore $\Delta(t_0) = 0$. But the two circles have no *real* touching point: the two circles are tangential to each other at two complex conjugate points $(1, \pm i, 0)$, known as the circular points in projective geometry. This explains why only the first real root of $\Delta(t)$ in $t \in [0, 1]$ always indicates an external contact between two moving elliptic disks, as assured by Corollary 2.10, on condition that the two elliptic disks are separate at the beginning (i.e. when $t = 0$). For each of the other real roots \bar{t} of $\Delta(t) = 0$, we need to check the root pattern of the characteristic polynomial at time \bar{t} to see whether there is an external contact. Theorem 2.6(1) tells us that a root \bar{t} of $\Delta(t) = 0$ corresponds to an external tangency of the two ellipses if and only if the characteristic polynomial $f(\lambda; \bar{t}) = 0$ has a negative double root in λ .

2.5.3 Robust computation

In this section, we shall discuss robust implementation of *CD-DISC* for testing collisions between two elliptic disks moving with rational motions. One of the steps in *CD-DISC* (see Section 2.3) is to construct the discriminant $\Delta(t)$. The discriminant is a

univariate polynomial in t that is the result of long polynomial computations (mainly polynomial multiplications) from the coefficients of the motion matrices, which are also polynomials. If the computations are carried out using the power series representation of the polynomials (i.e. $p(t) = \sum_{i=0}^n a_i t^i, a_i \in \mathbb{R}$), we found that *CD-DISC* suffered severely from numerical instability when the degree of motion is higher than 2, using double-precision floating-point arithmetic. By comparing intermediate results through the entire process with exact results produced by a Maple implementation of the same algorithm using exact integer computations, significant errors in the coefficients of $\Delta(t)$ are revealed. We perform a test, in which two elliptic disks move with degree 4 motion, using Maple with high-precision floating-point computation, and found that acceptable results could only be obtained when the number of decimal places in the floating point computation is increased above 20. In this case, the degree of $\Delta(t)$ is 96.

To overcome this numerical instability in processing high-degree polynomials, we turned to the Bernstein form of polynomials. The Bernstein form has the expression $\sum_{i=0}^n \binom{n}{i} a_i t^i (1-t)^{n-i}, a_i \in \mathbb{R}$, and is known to be numerically more stable for polynomial computations than the power form [17, 57]. In our current implementation of *CD-DISC*, we still use polynomials in the power form when computing the characteristic equation $f(\lambda; t) = 0$ from the motion matrices, and then convert the coefficients $g_i(t)$ of $f(\lambda; t) = 0$ into the Bernstein form; the numerical condition of this conversion is satisfactory since the terms $g_i(t)$ have relatively low degrees [16]. Finally, we derive $\Delta(t)$ by computing with polynomials in the Bernstein form. Our experiments show that this adoption of the Bernstein form significantly improves the robustness and accuracy of our collision detection procedure.

Having obtained the discriminant $\Delta(t)$ in a robust manner, the next step is to analyze its zeros. The extent of processing of $\Delta(t)$ depends on what kind of collision detection output is required by an application. The following three variants of *CD-DISC* have been implemented that give different collision detection outputs for two elliptic disks making rational motions:

VARIANT 1: reports whether the two elliptic disks collide.

VARIANT 2: reports whether the two elliptic disks collide and, if so, reports the time of first contact.

Variante 3: reports whether the two elliptic disks collide and, if so, reports all instants of external contact.

These different output types require different ways of handling the discriminant $\Delta(t) = 0$. For Variante 1, we only need to check for the existence of real roots of $\Delta(t) = 0$. We make use of a result that uses an idea similar to the Sturm sequence method to count the number of real roots of a polynomial in the scaled Bernstein form [44], which is expressed as $\sum_{i=0}^n b_i t^i (1-t)^{n-i}$, where $b_i \in \mathbb{R}$. This method inherits the robustness provided by the Bernstein form and therefore is suitable for high-degree polynomials. Using this technique, we are able to determine robustly whether $\Delta(t)$ has any real roots in the time interval $t \in [0, 1]$. Here, the coefficients $g_i(t)$ of the characteristic equation $f(\lambda; t) = 0$ are first transformed to the scaled Bernstein form, and then $\Delta(t)$ is obtained by computing with polynomials in the scaled Bernstein form. This treatment avoids the errors that would otherwise be caused by the high degree of $\Delta(t)$, if $\Delta(t)$ were first obtained in the Bernstein basis and then be transformed into the scaled Bernstein form. The scaled Bernstein form is used only in Variante 1.

For Variante 2, we need to solve for the smallest real root of $\Delta(t) = 0$ in $[0, 1]$, if one exists. For Variante 3, we must obtain all the real roots of $\Delta(t) = 0$ in $[0, 1]$. For each of these roots we also need to check for the existence of a negative double root of the characteristic polynomial, in order to verify the external tangency of the two elliptic disks. In both Variante 2 and Variante 3, we use the de Castel'jau algorithm to subdivide $\Delta(t)$ in the Bernstein form to locate all real roots of $\Delta(t)$ in $[0, 1]$. Using the convex hull property, we can discard an interval of t if the Bernstein coefficients of $\Delta(t)$ over that interval are all positive or all negative. For Variante 2, in which only the first contact time instant is needed, we can save computation time by continuing to subdivide only those intervals within which the smallest real root might be contained.

2.5.4 *Experimental results*

We shall first use a large set of synthesized motions to demonstrate the efficiency of our collision detection algorithm, and then use a more detailed example to show its accuracy. We generated 2,000 test cases for each of four kinds of motion: linear

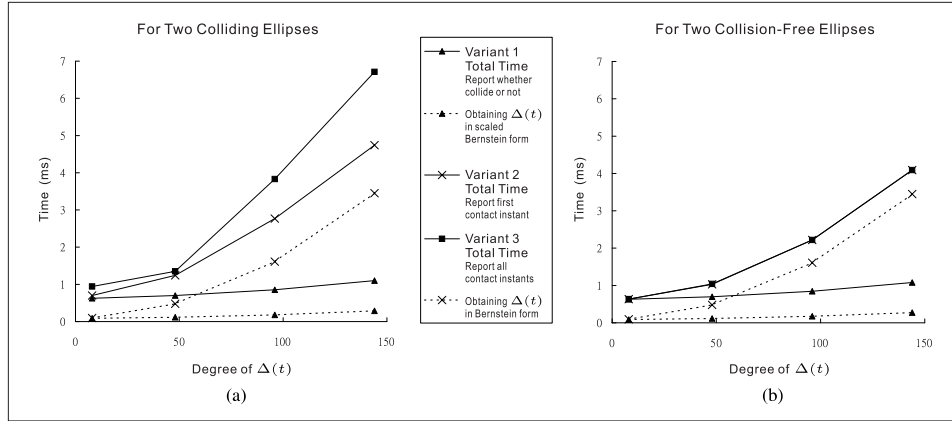


Figure 10: Average CPU time needed for *CD-DISC* to detect collision for two moving elliptic disks with different degrees of motion, when the two moving elliptic disks collide (a) or are collision-free (b). The solid lines show the total collision detection time for the three variants of *CD-DISC* (see Section 2.5.3) and the dashed lines show the computation time taken for obtaining $\Delta(t)$ in the Bernstein form and in the scaled Bernstein form.

translations and general rational motions of degree 2, degree 4 and degree 6. In each set of 2,000 cases, 1,000 cases were randomly generated pairs of colliding elliptic disks, and the other 1,000 were randomly generated pairs of collision-free elliptic disks. The experiments were run on a PC with a 2.2GHz Intel CPU and the timings are shown in Fig. 10. The graphs in Fig. 10(a) and Fig. 10(b) give the average CPU time taken by *CD-DISC* for colliding and collision-free elliptic disks, respectively. The three solid lines correspond to the three different outputs (Variants 1, 2 & 3 as described in Section 2.5.3) that *CD-DISC* can report. Clearly, more time is needed as the degree of motion and hence the degree of $\Delta(t)$ increases. For two elliptic disks making a motion of degree 6, for which the degree of $\Delta(t)$ is 144, it takes less than 1 ms to determine whether there is any collision, less than 5 ms to compute the instant of first contact, and less than 7 ms to compute all instants of contact.

For both colliding and collision-free ellipses, Variant 1 of the algorithm takes the same time, since the computation involved to decide whether there is a collision (i.e. to determine root existence by Sturm sequences) is the same in both cases. In the case of colliding elliptic disks (Fig. 10a), more time is needed for Variants 2 and 3 to detect the instants of contact; in general, the computation time increases as the number of roots of $\Delta(t)$ increases. When there is no collision (Fig. 10b), $\Delta(t)$ has no root and

the average CPU time taken for reporting the instant of first contact (Variant 2) or the instants of all contacts (Variant 3) is the same.

The dashed lines in the graphs show the time needed for obtaining the polynomial $\Delta(t)$ from the characteristic equation; in Variant 1, $\Delta(t)$ is obtained in the scaled Bernstein form, while in Variants 2 and 3, the computation is done in the Bernstein form. It is obvious that in Variants 2 and 3, obtaining $\Delta(t)$ takes up most of the overall time for collision detection. Polynomial multiplications in the scaled Bernstein form are much more efficient than that in the Bernstein form, which explains why the time needed for obtaining $\Delta(t)$ in Variant 1 is much less than that needed in Variants 2 and 3.

Next, we use a worked example to show the robustness of the algorithm *CD-DISC*.

Example 2.2. Consider two ellipses $\mathcal{A} : \frac{x^2}{5^2} + \frac{y^2}{10^2} = 1$ and $\mathcal{B} : \frac{x^2}{5^2} + \frac{y^2}{10^2} = 1$. Two moving elliptic disks $\mathcal{A}(t)$ and $\mathcal{B}(t)$, $t \in [0, 1]$, are defined by applying to \mathcal{A} and \mathcal{B} the following motions M_A and M_B :

$$M_A = \begin{pmatrix} -16t^4 + 32t^3 - 16t + 4 & -32t^3 + 48t^2 - 16t & -160t^3 - 240t^2 \\ & & + 160t - 40 \\ 32t^3 - 48t^2 + 16t & -16t^4 + 32t^3 - 16t + 4 & 480t^4 - 960t^3 \\ & & + 880t^2 - 400t + 80 \\ 0 & 0 & 16t^4 - 32t^3 \\ & & + 32t^2 - 16t + 4 \end{pmatrix},$$

$$M_B = \begin{pmatrix} -16t^4 + 32t^3 - 16t + 4 & 32t^3 - 48t^2 + 16t & 160t^3 - 240t^2 + 160t - 40 \\ -32t^3 + 48t^2 - 16t & -16t^4 + 32t^3 - 16t + 4 & -480t^4 + 960t^3 - 880t^2 + 400t - 80 \\ 0 & 0 & 16t^4 - 32t^3 + 32t^2 - 16t + 4 \end{pmatrix}.$$

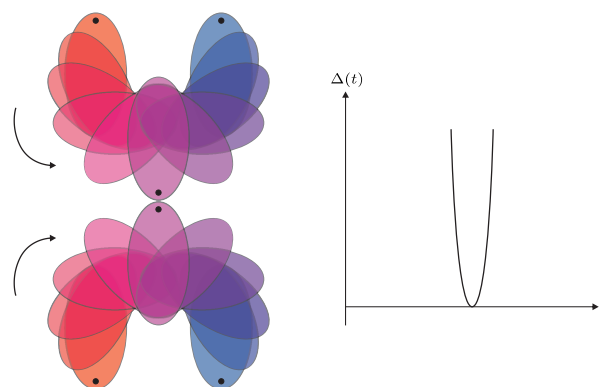


Figure 11: The two moving elliptic disks in Example 2 and their discriminant $\Delta(t)$.

The characteristic equation is

$$\begin{aligned}
f(\lambda; t) &= \det(\lambda A(t) - B(t)) \\
&= (-4096t^{24} + 49152t^{23} - 294912t^{22} + 1171456t^{21} - 3446784t^{20} + 7974912t^{19} \\
&\quad - 15048704t^{18} + 23721984t^{17} - 31756032t^{16} + 36517888t^{15} - 36360192t^{14} \\
&\quad + 31509504t^{13} - 23835904t^{12} + 15754752t^{11} - 9090048t^{10} + 4564736t^9 - 1984752t^8 \\
&\quad + 741312t^7 - 235136t^6 + 62304t^5 - 13464t^4 + 2288t^3 - 288t^2 + 24t - 1)\lambda^3 \\
&\quad + (-135168t^{24} + 1622016t^{23} - 11304960t^{22} + 55959552t^{21} - 206878720t^{20} \\
&\quad + 588967936t^{19} - 1325514752t^{18} + 2409461760t^{17} - 3596409600t^{16} + 4461631488t^{15} \\
&\quad - 4639457280t^{14} + 4065807360t^{13} - 3011391744t^{12} + 1886084608t^{11} - 997282816t^{10} \\
&\quad + 443501312t^9 - 164884848t^8 + 50819520t^7 - 12842880t^6 + 2624352t^5 - 426168t^4 \\
&\quad + 53808t^3 - 5120t^2 + 344t - 13)\lambda^2 \\
&\quad + (135168t^{24} - 1622016t^{23} + 11304960t^{22} - 55959552t^{21} + 206878720t^{20} \\
&\quad - 588967936t^{19} + 1325514752t^{18} - 2409461760t^{17} + 3596409600t^{16} - 4461631488t^{15} \\
&\quad + 4639457280t^{14} - 4065807360t^{13} + 3011391744t^{12} - 1886084608t^{11} + 997282816t^{10} \\
&\quad - 443501312t^9 + 164884848t^8 - 50819520t^7 + 12842880t^6 - 2624352t^5 + 426168t^4 \\
&\quad - 53808t^3 + 5120t^2 - 344t + 13)\lambda \\
&\quad + 4096t^{24} - 49152t^{23} + 294912t^{22} - 1171456t^{21} + 3446784t^{20} - 7974912t^{19} \\
&\quad + 15048704t^{18} - 23721984t^{17} + 31756032t^{16} - 36517888t^{15} + 36360192t^{14} \\
&\quad - 31509504t^{13} + 23835904t^{12} - 15754752t^{11} + 9090048t^{10} - 4564736t^9 + 1984752t^8 \\
&\quad - 741312t^7 + 235136t^6 - 62304t^5 + 13464t^4 - 2288t^3 + 288t^2 - 24t + 1 \\
&= 0
\end{aligned}$$

while the discriminant $\Delta(t)$ is of degree 96 and is omitted. Fig. 11 shows the two moving elliptic disks and the graph of $\Delta(t)$. The two disks are designed only to touch each other externally at $t = 0.5$, and are separate for the rest of the time. All three variants of *CD-DISC* determined correctly that there is a contact and Variants 2 and 3 reported the contact time accurately as $t = 0.5$. To examine the sensitivity of *CD-DISC*, the lower disk in Fig. 11 was translated by a small amount in the negative y direction,

so that the two moving disks attain a minimum separating distance $d > 0$ at $t = 0.5$. *CD-DISC* reports collision when $d < 10^{-6}$, and reports non-collision for larger values of d . Note that the above analysis is only based on the specific setup in this example. The robustness of the method depends not only on the degree of the motion, but also varies with different input, e.g. sizes of the ellipses. However, it is not easy to formulate the theoretical condition for the robustness of this method. If an error bound is crucial to an application, one may consider the use of the interval arithmetic [1] in the computations.

2.6 POSSIBLE FUTURE WORK

We have demonstrated that the use of Bernstein forms for polynomial manipulation significantly increases numerical stability of *CD-DISC* for high-degree rational motions, which conforms the others' observations [17, 16, 57]. Our experiments show that *CD-DISC* is fast and accurate for detecting collisions between moving ellipses under continuous rational motions of degree 6 or less; note that research [68] in planar rational motions suggests that rational motions of degree 4 are adequate for modeling all smooth motions in practice.

There are several problems open for further research. Knowing the minimum distance between two collision-free moving elliptic disks would be useful for motion path planning. Work is needed to study the relationship between the minimum distance and the value of the discriminant $\Delta(t)$. An observation that can already be made is that the difference between the two negative roots of the characteristic polynomial becomes smaller as the two separate disks approach each other gradually, and eventually the two roots merge into a negative double root, at which point the two disks become externally tangential to each other, signaled by the vanishing of the discriminant function.

3

ELLIPSES MOVING IN THE SPACE

We now consider CCD of ellipses moving in \mathbb{R}^3 . Our idea is to transform the problem into a 1D collision detection problem in the line where the containing planes of the ellipses intersect. We will establish the formulation of the problem reduction, and provide an algorithm which is devised based on a detailed analysis of the 1D problem.

3.1 REDUCTION FROM 3D TO 1D

Consider two moving ellipses $\hat{\mathcal{A}}(t) : X^T \hat{\mathcal{A}}(t) X = 0$ and $\hat{\mathcal{B}}(t) : X^T \hat{\mathcal{B}}(t) X = 0$ in \mathbb{R}^3 . Let $\Pi_A(t)$ and $\Pi_B(t)$ be the planes containing $\hat{\mathcal{A}}(t)$ and $\hat{\mathcal{B}}(t)$, respectively. Suppose that $\Pi_A(t) \neq \Pi_B(t)$ and let $\mathcal{L}(s; t)$ be their line of intersection, where s is a parameterization of the line.

Substituting $\mathcal{L}(s; t)$ into the ellipse equations, we have:

$$\begin{aligned} \mathbf{h}(s; t) : X^T(s; t) \hat{\mathcal{A}}(t) X(s; t) &= 0 \\ \mathbf{g}(s; t) : X^T(s; t) \hat{\mathcal{B}}(t) X(s; t) &= 0 \end{aligned} \tag{3.1}$$

We first assume that t is fixed, i.e., we consider two static ellipses $\hat{\mathcal{A}}$ (on plane Π_A) and $\hat{\mathcal{B}}$ (on plane Π_B) in \mathbb{R}^3 , and that Π_A and Π_B intersect at line $\mathcal{L}(s)$. Hence we have:

$$\begin{aligned} \mathbf{h}(s) : X^T(s) \hat{\mathcal{A}} X(s) &= 0 \\ \mathbf{g}(s) : X^T(s) \hat{\mathcal{B}} X(s) &= 0 \end{aligned}$$

The solution of $h(s) = 0$ gives the intersection between $\mathcal{L}(s)$ and $\hat{\mathcal{A}}$. Likewise, the solution of $g(s) = 0$ gives the intersection between $\mathcal{L}(s)$ and $\hat{\mathcal{B}}$.

Since $h(s)$ and $g(s)$ are quadratic in s , we may write

$$h(s) = \begin{pmatrix} s & 1 \end{pmatrix} A \begin{pmatrix} s \\ 1 \end{pmatrix} \quad \text{and} \quad g(s) = \begin{pmatrix} s & 1 \end{pmatrix} B \begin{pmatrix} s \\ 1 \end{pmatrix}$$

where A and B are 2×2 real coefficient matrices. It means that $h(s)$ and $g(s)$ can be considered as two “1D ellipses” \mathcal{A} and \mathcal{B} (i.e., intervals or line segments) which can be either real or imaginary. Now, $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ intersect if and only if there is real intersection between \mathcal{A} and \mathcal{B} . Hence, we have essentially reduced a 3D problem (namely, collision detection of ellipses moving in the space) to a 1D problem (collision detection of intervals in a line.)

3.2 CONDITION FOR SEPARATION OF 1D ELLIPSES

Let us now consider the intersection of $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ in \mathbb{C} , where A and B are 2×2 real matrices and $X = (x \ 1)^T$ is the homogeneous coordinates of a point in \mathbb{C} .

Let $f(\lambda) = \det(\lambda A - B)$ be the characteristic polynomial of \mathcal{A} and \mathcal{B} . We then study the relationship between the roots of $f(\lambda)$ and the intersection of \mathcal{A} and \mathcal{B} .

Let us first assume that both \mathcal{A} and \mathcal{B} are real. By an affine transformation, we may assume that $\mathcal{A} = [-1, 1]$, and that $\mathcal{B} = [a, b]$ for some $a, b \in \mathbb{R}$. Then we have

$$\begin{aligned} h(x) &= X^T A X = x^2 - 1 \\ g(x) &= X^T B X = (x - a)(x - b) \end{aligned}$$

where

$$X = \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & -\frac{a+b}{2} \\ -\frac{a+b}{2} & ab \end{pmatrix},$$

and

$$f(\lambda) = \begin{vmatrix} \lambda - 1 & \frac{a+b}{2} \\ \frac{a+b}{2} & -\lambda - ab \end{vmatrix} = -\lambda^2 + (1 - ab)\lambda - \left(\frac{a-b}{2}\right)^2.$$

Let λ_1, λ_2 be the two roots of $f(\lambda) = 0$. There are three cases:

Case I: λ_1 and λ_2 are real and distinct. Then the discriminant of $f(\lambda)$ is $\Delta = (a^2 - 1)(b^2 - 1) > 0$, and we have (1) $|a| > 1$ and $|b| > 1$; or (2) $|a| < 1$ and $|b| < 1$. Also, since $\lambda_1 \cdot \lambda_2 = \left(\frac{a-b}{2}\right)^2 \geq 0$, λ_1 and λ_2 must be of the same sign, or either one equals 0.

1. If $\lambda_1, \lambda_2 > 0$, then $\lambda_1 + \lambda_2 = 1 - ab > 0$. Hence $ab < 1$ and we have $[-1, 1] \subset [a, b]$ which corresponds to (1) $|a| > 1$ and $|b| > 1$, or $[a, b] \subset [-1, 1]$ which corresponds to (2) $|a| < 1$ and $|b| < 1$. When either λ_1 or λ_2 equals 0, we have $a = b \in [-1, 1]$.
2. If $\lambda_1, \lambda_2 < 0$, then $\lambda_1 + \lambda_2 = 1 - ab < 0$. We have $ab > 1$ which corresponds to (1) $|a| > 1$ and $|b| > 1$, i.e., the interval $[a, b] \cap [-1, 1] = \emptyset$. When either λ_1 or λ_2 equals 0, we have $a = b$.

Case II: $\lambda_1 = \lambda_2$ is a real double root. Then $\Delta = (a^2 - 1)(b^2 - 1) = 0$, i.e. $a = \pm 1$ or $b = \pm 1$. Also, $\lambda_1 = \lambda_2 = (1 - ab)/2$. Since a and b are symmetric in this expression, we will only consider $a = \pm 1$.

- Suppose that $\lambda_1 = \lambda_2 = (1 - ab)/2 > 0$ is a positive double root. If $a = 1$, then $b < 1$; If $a = -1$, then $b > -1$. Hence, the intervals $[-1, 1]$ and $[a, b]$ share a common boundary point and one interval contains the other. Or we say that the two intervals have an internal touch.
- Suppose that $\lambda_1 = \lambda_2 = (1 - ab)/2 < 0$ is a negative double root. If $a = 1$, then $b > 1$; if $a = -1$, then $b < -1$. Hence, the intervals $[-1, 1]$ and $[a, b]$ share a common boundary point; however, they do not overlap. The intervals are said to be in external touch.
- Suppose that $\lambda_1 = \lambda_2 = 0$. Since $\left(\frac{a-b}{2}\right)^2 = 0$, we have $a = b = \pm 1$.

Case III: λ_1 and λ_2 are complex conjugates. We have $\Delta = (a^2 - 1)(b^2 - 1) < 0$. Then, either (1) $|a| < 1$ and $|b| > 1$ or (2) $|a| > 1$ and $|b| < 1$. Hence, we have either $-1 < a < 1 < b$ or $a < -1 < b < 1$. Equivalently, the intervals $[-1, 1]$ and $[a, b]$ overlap but neither one is contained in the other.

We next consider the case when at least one of \mathcal{A} or \mathcal{B} is imaginary. Without loss of generality, let us assume that \mathcal{A} is imaginary. Since the coefficients of the matrix A (and hence the coefficients of $h(x)$) are real, we have $\mathcal{A} = [z, \bar{z}]$ where $z \in \mathbb{C}$ and \bar{z} is the complex conjugate of z . By an affine transformation, \mathcal{A} can be mapped to the two points $[i, -i]$ and \mathcal{B} to $[a, b]$ where a, b may be either real or imaginary. We then write

$$\begin{aligned} h(x) &= X^T A X = x^2 + 1 \\ g(x) &= X^T B X = (x - a)(x - b) \end{aligned}$$

where

$$X = \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & -\frac{a+b}{2} \\ -\frac{a+b}{2} & ab \end{pmatrix},$$

and

$$\begin{aligned} f(\lambda) &= \begin{vmatrix} \lambda - 1 & \frac{a+b}{2} \\ \frac{a+b}{2} & \lambda - ab \end{vmatrix} \\ &= \lambda^2 - (1 + ab)\lambda - \left(\frac{a-b}{2}\right)^2. \end{aligned}$$

Now, the discriminant of $f(\lambda)$ is $\Delta = (a^2 + 1)(b^2 + 1) \geq 0$, and hence we have the following two cases:

Case I: λ_1 and λ_2 are real and distinct. Since $\lambda_1 \cdot \lambda_2 = -\left(\frac{a-b}{2}\right)^2$, we have:

- If λ_1 and λ_2 are of the same sign, i.e., $\lambda_1 \cdot \lambda_2 > 0$, then $\left(\frac{a-b}{2}\right)^2 < 0$ and a, b are imaginary. Moreover, $\lambda_1 + \lambda_2 = 1 + ab > 0$. Hence, $\lambda_1 > 0$ and $\lambda_2 > 0$.
- If λ_1 and λ_2 are of different signs, i.e., $\lambda_1 \cdot \lambda_2 < 0$, then $\left(\frac{a-b}{2}\right)^2 > 0$ and a, b are real.

Case II: $\lambda_1 = \lambda_2$ is a real double root. Then $\Delta = 0$ implies $a^2 + 1 = 0$ or $b^2 + 1 = 0$.

In both cases, we have $\mathcal{B} = \{a, b\} = \{\mathbf{i}, -\mathbf{i}\}$, i.e., $\mathcal{A} = \mathcal{B}$. Moreover, $\lambda_1^2 = -(\frac{a-b}{2})^2 = 1 > 0$.

So far, we have considered the cases where at least one of \mathcal{A} or \mathcal{B} are non-degenerate, i.e., the 1D ellipse is a line segment of non-zero length. Now, let us assume that both \mathcal{A} and \mathcal{B} are degenerate. We may further assume that $\mathcal{A} = \{0\}$ and $\mathcal{B} = \{b\}$. Then,

$$\begin{aligned} h(x) &= X^T A X = x^2 \\ g(x) &= X^T B X = (x - b)^2 \end{aligned}$$

where

$$X = \begin{pmatrix} x \\ 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & \\ & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 1 & -b \\ -b & b^2 \end{pmatrix},$$

and

$$\begin{aligned} f(\lambda) &= \begin{vmatrix} \lambda - 1 & b \\ b & -b^2 \end{vmatrix} \\ &= -b^2 \lambda. \end{aligned}$$

It is easy to see that if $\mathcal{A} = \mathcal{B}$, then $b = 0$ and $f(\lambda) \equiv 0$. Otherwise, $f(\lambda)$ is linear and has a single root equals 0. Table 2 summarizes two 1D ellipses and the roots of their characteristic equation.

3.3 THE ALGORITHM FOR DETECTING COLLISION OF TWO MOVING ELLIPSES IN THE 3D SPACE

Theorem 3.1 follows from the problem reduction in section 3.1 and the separation condition of 1D ellipses in section 3.2:

Theorem 3.1. *Given two ellipses $\hat{\mathcal{A}}$ (on plane Π_A) and $\hat{\mathcal{B}}$ (on plane Π_B) in \mathbb{R}^3 , and that Π_A and Π_B intersect at some line $\mathcal{L} \in \mathbb{R}^3$. Let $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ be the “1D ellipses” characterizing the intersections of \mathcal{L} with $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, respectively (as described in*

Table 2: Configuration of two 1D ellipses and the roots of their characteristic equation. Ellipses are represented in pairs of brackets of the same style. Real intervals are represented by solid brackets, while imaginary intervals are represented by dotted brackets. Degenerate intervals of one point is represented by a dot or a cross.

The two roots of $f(\lambda) = 0$	Configuration
Distinct positive	
Distinct negative	
One zero, one positive	
One zero, one negative	
One negative, one positive	
Double positive	
Double negative	
Double zero	
Complex conjugate	
$f(\lambda)$ is linear with root equals 0	
$f(\lambda) \equiv 0$	

section 3.1.) Furthermore, let $f(\lambda) = \det(\lambda A - B)$ be the characteristic polynomial of \mathcal{A} and \mathcal{B} . Then, the ellipses $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ are in external touch in \mathbb{R}^3 if and only if

1. $f(\lambda)$ has a double negative root (Figure 12a); or
2. $f(\lambda)$ has a double root equals 0 (Figure 12b); or
3. $f(\lambda) \equiv 0$ (Figure 12c).

Here an ellipse is considered to comprise of the set of points that satisfies $X^T E X \leq 0$, where E is the coefficient matrix of the ellipse. Two ellipses $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ are in external touch if and only if $\hat{\mathcal{A}} \cap \hat{\mathcal{B}} \neq \emptyset$ and $\text{Int}(\hat{\mathcal{A}}) \cap \text{Int}(\hat{\mathcal{B}}) = \emptyset$ where $\text{Int}(\cdot)$ denote the interior of a point set.

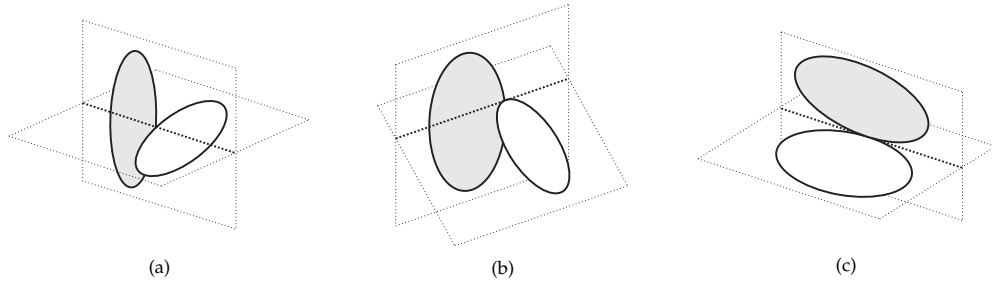


Figure 12: The three configurations of two touching ellipses in 3D. Sub-figures (a), (b), (c) correspond to the three conditions (1), (2), (3), of Theorem 3.1, respectively.

We now present an algorithm for collision detection of two moving ellipses in \mathbb{R}^3 . This algorithm is in spirit similar to *CD-DISC* in section 2.3 for detecting collision of two moving ellipses in a plane. It computes a discriminant $\Delta(t)$ of the characteristic equation $f(\lambda; t) = 0$ of two moving 1D ellipses, whose roots corresponds to time instants t_0 where $f(\lambda; t_0) = 0$ has a double root or $f(\lambda; t_0) \equiv 0$. This in turn indicates when the moving ellipses are in contact in \mathbb{R}^3 .

Algorithm: CD-DISC_{3D}

Input: Two moving ellipses $\hat{\mathcal{A}}(t) : X^T A X = 0$ and $\hat{\mathcal{B}}(t) : X^T B X = 0$ in \mathbb{R}^3 , $t \in [0, 1]$, and that $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate.

Output: Whether $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide in $[0, 1]$: COLLISION or COLLISION-FREE.

Step 1: Let $\Pi_A(t)$ and $\Pi_B(t)$ be the containing plane of $\mathcal{A}(t)$ and $\mathcal{B}(t)$, respectively.

If $\Pi_A(t) \equiv \Pi_B(t)$ for $t \in [0, 1]$, the collision detection of $\mathcal{A}(t)$ and $\mathcal{B}(t)$ can

be reduced to that of two moving ellipses in a plane and therefore be solved using the algorithm *CD-DISC* in Chapter 2.

Step 2: Otherwise, compute the intersection line $\mathcal{L}(s;t) = u(t) + s \cdot v(t)$ between $\Pi_A(t)$ and $\Pi_B(t)$.

Step 3: Compute

$$h(s;t) : X^T(s;t)\hat{A}(t)X(s;t) = 0 \quad \text{and} \quad g(s;t) : X^T(s;t)\hat{B}(t)X(s;t) = 0$$

as in Eq. (3.1).

Step 4: Rewrite $h(s;t)$ and $g(s;t)$, and obtain $A(t)$, $B(t)$ as in

$$h(s;t) = \begin{pmatrix} s & 1 \end{pmatrix} A(t) \begin{pmatrix} s \\ 1 \end{pmatrix} \quad \text{and} \quad g(s;t) = \begin{pmatrix} s & 1 \end{pmatrix} B(t) \begin{pmatrix} s \\ 1 \end{pmatrix}.$$

Step 5: Compute $f(\lambda;t) = \det(\lambda A(t) - B(t))$.

Step 6: Compute the discriminant of $\Delta_f(t)$ of $f(\lambda;t)$.

Step 7: Let the roots of $\Delta_f(t)$ be $0 < t_0 < \dots < t_k < 1$. For each t_i ,

- If $\Pi_A(t_i) \equiv \Pi_B(t_i)$, mark t_i as *valid* if $\mathcal{A}(t_i)$ and $\mathcal{B}(t_i)$ have contact (determined using the separate condition of two ellipses in a plane given by Theorem 2.6); otherwise, mark t_i as *invalid*.
- Else if $\Pi_A(t_i) \neq \Pi_B(t_i)$ and $\Pi_A(t_i) \parallel \Pi_B(t_i)$, mark t_i as *invalid*.
- Else mark t_i as *valid*.

Step 8: If $\Delta_f(t)$ has no valid root in $[0,1]$, report **COLLISION-FREE**; otherwise, report **COLLISION**.

Remark 3.2. Step 7 above serves to verify whether a root t_i corresponds to a valid contact time instant, since the case where $\Pi_A(t_i) \parallel \Pi_B(t_i)$ will also lead to $\Delta(t_i) = 0$.

Similar to *CD-DISC*, we may also compute the contact time for the ellipses. For example, the first contact time instant of $\mathcal{A}(t)$ and $\mathcal{B}(t)$ will be given by the smallest *valid* root of $\Delta_f(t) = 0$ in $[0,1]$.

An example for CCD of two ellipses moving in 3D will be given later in section 6.4.

4

STATIC ELLIPSOIDS

In this chapter, we present an algorithm for detecting overlap between two stationary ellipsoids, focusing on an efficient implementation with a minimized number of arithmetic operations. The result will be further combined with a simple method for constructing a separating plane for two disjoint ellipsoids, and an efficient algorithm is devised for detecting collisions between two moving ellipsoids whose motions are described by sampled positions at consecutive time frames.

4.1 DETECTING OVERLAP BETWEEN STATIONARY ELLIPSOIDS

In this section we present an efficient algorithm for detecting overlap between two stationary ellipsoids which are assumed to be sampled instances of two moving ellipsoids at the same instant. This algorithm is based on the separation condition for two ellipsoids proved in [70]. The contribution here is an optimized algorithm with a minimal number of arithmetic operations; we conclude that 107 additions/subtractions, 141 multiplications and 6 divisions are needed. This efficient implementation, while having practical values in its own right, will be invoked in the subsequent method for the continuous collision detection of moving ellipsoids as described later in Chapter 5.

An ellipsoid \mathcal{A} is represented by a quadratic equation $X^T A X = 0$ in \mathbb{E}^3 , where $X = (x, y, z, w)^T$ are the homogeneous coordinates of a point in 3D space. The symmetric matrix A is normalized so that the interior of \mathcal{A} is given by $X^T A X < 0$; this amounts to assuming that the determinant $|A| < 0$.

Two ellipsoids are said to be *overlapping* if their interiors have non-empty intersection. They are said to be *separate* or *disjoint* if their boundary surfaces and interiors share no common points. Two ellipsoids that are not separate but share no common interior points are said to be *touching*.

For two ellipsoids $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ in \mathbb{E}^3 , the quartic polynomial $f(\lambda) = \det(\lambda A - B)$ is called the *characteristic polynomial* and $f(\lambda) = 0$ is called the *characteristic equation* of \mathcal{A} and \mathcal{B} . The polynomial $f(\lambda)$ has degree 4, its leading term has a negative coefficient, and it always has two positive real roots. The following theorem [70] captures the relationship between the geometric configuration of two ellipsoids and the roots of their characteristic equation:

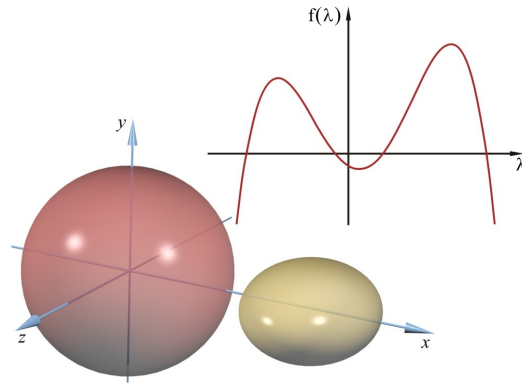
Theorem 4.1. (Separation condition of two stationary ellipsoids) *Let \mathcal{A} and \mathcal{B} be two ellipsoids with characteristic equation $f(\lambda) = 0$. Then,*

1. *\mathcal{A} and \mathcal{B} are separate if and only if $f(\lambda) = 0$ has two distinct negative roots;*
2. *\mathcal{A} and \mathcal{B} touch each other externally if and only if $f(\lambda) = 0$ has a negative double root.*

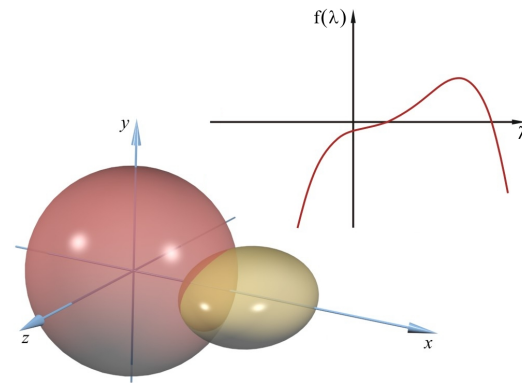
Remark 4.2. Note that the theorem in [70] assumes that the characteristic equation has the form of $f(\lambda) = \det(\lambda A + B) = 0$ and therefore the result there is stated in terms of positive roots. Our changes here make the presentation consistent with the classic literature in linear algebra.

Remark 4.3. Clearly, the leading coefficient and the constant term of $f(\lambda)$ are $|A|$ and $|B|$. So they are negative [70]. This implies that $f(\lambda) = 0$ has two distinct negative roots if and only if $f(\lambda_0) > 0$ for some $\lambda_0 < 0$. The latter condition on a sign test is more convenient, especially when we consider two moving ellipsoids.

Figure 13(a) shows two disjoint ellipsoids. Note that their characteristic equation has two distinct negative roots. In Figure 13(b), two ellipsoids overlap and their characteristic equation does not have any negative root.



(a)



(b)

Figure 13: Two (a) disjoint; (b) overlapping ellipsoids and the corresponding $f(\lambda)$.

4.1.1 Characteristic polynomial

For efficient implementation, it is crucial to set up the characteristic equation using a minimal number of arithmetic operations. We now present an efficient algorithm for this computation.

An ellipsoid is said to be in canonical form if it is represented by a diagonal matrix

$$A = \begin{bmatrix} 1/a^2 & 0 & 0 & 0 \\ 0 & 1/b^2 & 0 & 0 \\ 0 & 0 & 1/c^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \quad (4.1)$$

Under an affine transformation M_A this ellipsoid is transformed to one in a general form with coefficient matrix $(M_A^{-1})^T A M_A^{-1}$. Now assume that we use two transforms M_A and M_B to obtain two ellipsoids $(M_A^{-1})^T A M_A^{-1}$ and $(M_B^{-1})^T B M_B^{-1}$, where A and B are diagonal matrices representing ellipsoids in canonical positions. Then the characteristic polynomial of the two ellipsoids is $f(\lambda) = \det(\lambda(M_A^{-1})^T A M_A^{-1} - (M_B^{-1})^T B M_B^{-1})$.

In the following we first compute the coefficients of the quartic polynomial $f(\lambda)$, and then the signs of the roots of the polynomial are computed to determine the relative configuration of the two ellipsoids. Given two ellipsoids represented as the images of their standard diagonal form (cf. (4.1)) under the transformations M_A and M_B , we may simultaneously transform them to A and $M_A^T (M_B^{-1})^T B M_B^{-1} M_A$, where A is a diagonal matrix as in (4.1) and $M_A^T (M_B^{-1})^T B M_B^{-1} M_A$ is treated as a general 4×4 matrix. The characteristic polynomial then takes the following form: $f(\lambda) = \det(\lambda A - M_A^T (M_B^{-1})^T B M_B^{-1} M_A)$; obviously the roots of the characteristic polynomial remain the same as before. The power form of $f(\lambda)$ in λ can be obtained by expanding the determinant $\det(\lambda A - M_A^T (M_B^{-1})^T B M_B^{-1} M_A)$. Then we can use its Sturm sequence to determine whether the two ellipsoids overlap, by Theorem 4.1.

4.1.2 Computational cost

To count the number of the negative real roots of $f(\lambda)$, we will first compute the Sturm sequence of $f(\lambda)$ and then check the sign flips of this sequence at 0 and $-\infty$. For the moment we assume that M_A and M_B are Euclidean transformations, since this is a case that is used most often in applications. To compute $M_A^T (M_B^{-1})^T B M_B^{-1} M_A$, we note that M_B is the composition of a rotation R_B followed by a translation V_B , so its inverse M_B^{-1} is equivalent to a rotation R_B^T followed by a translation $-R_B^T V_B$. Based on this observation, we can count the arithmetic operations as follows:

1. Computing M_B^{-1} requires 9 additions/subtractions and 9 multiplications.
2. $M_B^{-1} M_A$ requires 27 additions/subtractions and 36 multiplications.
3. $M_A^T (M_B^{-1})^T$ is the transpose of $M_B^{-1} M_A$, and so needs no arithmetic operation.

4. Since B is a diagonal matrix, $BM_B^{-1}M_A$ requires 12 multiplications.

5. Finally, $M_A^T(M_B^{-1})^TBM_B^{-1}M_A$ can be constructed using additional 21 additions/-subtractions and 30 multiplications.

Thus we need 57 additions/subtractions and 87 multiplications to obtain $M_A^T(M_B^{-1})^TBM_B^{-1}M_A$. Then the characteristic polynomial can be computed with another 29 additions/subtractions and 39 multiplications using the algorithm presented in section 4.1.3. The derivative of a quartic polynomial can be computed using 3 multiplications. To divide a degree n polynomial by a degree $(n - 1)$ polynomial, we need $2(n - 1)$ additions/subtractions, $2(n - 1)$ multiplications and 2 divisions. Thus we can compute the Sturm sequence using 12 additions/subtractions, 15 multiplications and 6 divisions. To find the number of negative real roots, we need to examine the signs of the leading term and constant term of the polynomials in the Sturm sequence, for which 8 additions/subtractions are needed to count the number of sign flips. In summary, we need a total of 107 additions/subtractions, 141 multiplications and 6 divisions for collision detection between two stationary ellipsoids.

When the two stationary ellipsoids above are sampled from affine motions, it can be shown that we need a total of 125 additions/subtractions, 156 multiplications and 18 divisions for detecting their collision. The different arithmetic operation count from the case of rigid motions is due solely to the inverse computation of M_B^{-1} which takes a lot more operations for affine motions. We skip the detailed counting here.

We have implemented the collision detection algorithm in C++ and run our tests on a desktop PC with an Intel Core 2 Duo E6600 2.40 GHz CPU (single-threaded) and a 2GB main memory. In the case of motion matrices with elements of rational degree 4, the matrices M_A and M_B are constructed using about 100 additions/subtractions and 100 multiplications. Including this, the whole procedure of detecting overlap between two ellipsoids took less than 0.7 μ sec.

4.1.3 Coefficients of the characteristic polynomial

We present an efficient algorithm for computing the five coefficients of the characteristic polynomial $f(\lambda)$ of degree 4. Let $M_A^T(M_B^{-1})^T B M_B^{-1} M_A = [b_{ij}]_{4 \times 4}$. Then the characteristic polynomial is given in the following simple form:

$$f(\lambda) = \det(\lambda A - M_A^T(M_B^{-1})^T B M_B^{-1} M_A)$$

$$= \begin{vmatrix} \lambda/a^2 - b_{11} & -b_{12} & -b_{13} & -b_{14} \\ -b_{21} & \lambda/b^2 - b_{22} & -b_{23} & -b_{24} \\ -b_{31} & -b_{32} & \lambda/c^2 - b_{33} & -b_{34} \\ -b_{41} & -b_{42} & -b_{43} & -\lambda - b_{44} \end{vmatrix}.$$

By expanding this determinant, the five coefficients can be constructed as follows:

The 4th-degree term (T4):

$$-\frac{1}{a^2 b^2 c^2}$$

The 3rd-degree term (T3):

$$\frac{b_{11}}{b^2 c^2} + \frac{b_{22}}{a^2 c^2} + \frac{b_{33}}{a^2 b^2} - \frac{b_{44}}{a^2 b^2 c^2}$$

The 2nd-degree term (T2):

$$\frac{b_{33}b_{44} - b_{34}b_{43}}{a^2 b^2} + \frac{b_{11}b_{44} - b_{14}b_{41}}{b^2 c^2} + \frac{b_{22}b_{44} - b_{24}b_{42}}{a^2 c^2}$$

$$+ \frac{b_{23}b_{32} - b_{22}b_{33}}{a^2} + \frac{b_{13}b_{31} - b_{11}b_{33}}{b^2} + \frac{b_{12}b_{21} - b_{11}b_{22}}{c^2}$$

The 1st-degree term (T1):

$$\frac{-b_{22}b_{33}b_{44} + b_{22}b_{34}b_{43} + b_{33}b_{42}b_{24}}{a^2} + \frac{b_{44}b_{32}b_{23} - b_{32}b_{24}b_{43} - b_{42}b_{23}b_{34}}{a^2}$$

$$+ \frac{-b_{11}b_{33}b_{44} + b_{11}b_{34}b_{43} + b_{33}b_{14}b_{41}}{b^2} + \frac{b_{44}b_{13}b_{31} - b_{31}b_{14}b_{43} - b_{41}b_{13}b_{34}}{b^2}$$

$$+ \frac{-b_{11}b_{22}b_{44} + b_{11}b_{24}b_{42} + b_{22}b_{14}b_{41}}{c^2} + \frac{b_{44}b_{12}b_{21} - b_{21}b_{14}b_{42} - b_{41}b_{12}b_{24}}{c^2}$$

$$+ b_{11}b_{22}b_{33} - b_{11}b_{23}b_{32} - b_{22}b_{13}b_{31} - b_{33}b_{12}b_{21} + b_{21}b_{13}b_{32} + b_{31}b_{12}b_{23}$$

The constant term (To):

$$\begin{aligned}
& b_{11}b_{22}b_{33}b_{44} - b_{11}b_{22}b_{34}b_{43} - b_{11}b_{33}b_{24}b_{42} - b_{11}b_{44}b_{23}b_{32} \\
& \quad - b_{22}b_{33}b_{14}b_{41} - b_{22}b_{44}b_{13}b_{31} - b_{33}b_{44}b_{12}b_{21} \\
& \quad + b_{11}b_{32}b_{24}b_{43} + b_{11}b_{23}b_{34}b_{42} + b_{22}b_{13}b_{34}b_{41} \\
& + b_{22}b_{31}b_{14}b_{43} + b_{33}b_{12}b_{24}b_{41} + b_{33}b_{21}b_{14}b_{42} + b_{44}b_{12}b_{23}b_{31} \\
& + b_{44}b_{21}b_{13}b_{32} + b_{12}b_{21}b_{34}b_{43} + b_{13}b_{31}b_{24}b_{42} + b_{14}b_{41}b_{23}b_{32} \\
& \quad - b_{21}b_{14}b_{43}b_{32} - b_{21}b_{13}b_{34}b_{42} - b_{31}b_{12}b_{24}b_{43} \\
& \quad - b_{31}b_{14}b_{42}b_{23} - b_{41}b_{12}b_{23}b_{34} - b_{41}b_{13}b_{32}b_{24}
\end{aligned}$$

If M_A and M_B are rigid transformations, the constant term equals $\det(-B)$ and the following function efficiently computes the coefficients $f(\lambda)$ using 29 additions/subtractions and 39 multiplications.

Generate-Characteristic-Polynomial

```

/* Variable definition
ea,eb,ec are the diagonal members of matrix A
ab = ea * eb, ac = ea * ec, bc = eb * ec,
abc = ea * eb * ec
bij is a member of the matrix  $M_A^T(M_B^{-1})^T B M_B^{-1} M_A$ 
*/
begin
b12s = b12 * b12; b13s = b13 * b13;
b14s = b14 * b14; b23s = b23 * b23;
b24s = b24 * b24; b34s = b34 * b34;
b2233 = b22 * b33;

termA = b11 * bc + b22 * ac + b33 * ab;
termB = (b2233 - b23s)*ea + (b11 * b33 - b13s)*eb
      + (b11 * b22 - b12s)*ec;

T4 = -abc;
T3 = termA - b44 * abc;
T2 = termA * b44 - termB - b34s * ab - b14s * bc
      - b24s * ac;
tmp1 = termB * b44;
tmp2 = b11*(b2233 + eb * b34s + ec * b24s - b23s);

```

```

tmp3 = b22*(ea * b34s + ec * b14s - b13s);
tmp4 = b33*(ea * b24s + eb * b14s - b12s);
tmp5 = b34*(ea * b23 * b24 + eb * b13 * b14)
      + b12*(ec * b14 * b24 - b13 * b23);
tmp5 += tmp5; // multiply by 2

T1 = -tmp1 + tmp2 + tmp3 + tmp4 - tmp5;
T0 = constant; // constant value det[-B]
end;

```

4.1.4 Contact point of two touching ellipsoids

Part (2) of Theorem 4.1 states that two ellipsoids have external contact if and only if their characteristic equation $f(\lambda) = 0$ has a negative double root λ_0 . It is also proved in [70, Lemma 5] that the contact point of two touching ellipsoids is given by the solution of $(\lambda_0 A - B)X = 0$, as summarized in the following theorem.

Theorem 4.4. *Suppose that two ellipsoids $X^T A X = 0$ and $X^T B X = 0$ touch externally, i.e., $f(\lambda) = 0$ has a negative double root λ_0 . Then $\text{rank}(\lambda_0 A - B) = 3$ and the homogeneous coordinates of the contact point X_0 are given by the unique nontrivial solution (up to a multiplicative constant) of $(\lambda_0 A - B)X = 0$.*

4.2 CONSTRUCTING A SEPARATING PLANE

It is well known that the efficiency of collision detection can be greatly improved by the use of a separating plane [2]. Once a plane separating two ellipsoids is found, there can be no collision between the ellipsoids until one of them collides with the separating plane. Thus the original problem is reduced to two simpler subproblems of searching for an intersection between a plane and an ellipsoid. Applying an affine transformation, an ellipsoid and a plane can be reduced to a sphere and a plane, and each of these subproblems then becomes equivalent to computing the distance between the center of the sphere and a plane.

Since ellipsoids are preserved under affine motion, our approach can be applied to ellipsoids that are moving and deforming under affine transformation. This is an important advantage over specialized algorithms that work only for simple geometric shapes such as axis-aligned boxes, spheres, cylinders, cones, or tori; such algorithms may not be generalized when affine motions are used.

In this section we show how to construct a separating plane of two disjoint ellipsoids.

Theorem 4.5. *Let $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ be two disjoint ellipsoids. Let V_i denote the four eigenvectors of $A^{-1}B$ associated with the eigenvalues λ_i , $i = 0, 1, 2, 3$. Then their endpoints V_i form the vertices of a tetrahedron, denoted by $[V_0 V_1 V_2 V_3]$, that is self-polar for the ellipsoids \mathcal{A} and \mathcal{B} ¹. Furthermore, V_0 and V_1 are outside \mathcal{A} and \mathcal{B} , V_2 is inside \mathcal{B} , and V_3 is inside \mathcal{A} .*

Proof. Since \mathcal{A} and \mathcal{B} are disjoint, by Theorem 4.1, $\det(\lambda A - B) = 0$ has two distinct negative roots and two positive roots: $\lambda_3 < \lambda_2 < 0 < \lambda_1 \leq \lambda_0$. When $\lambda_1 < \lambda_0$, the four eigenvalues of $A^{-1}B$, which are equal to the roots of the quartic equation $f(\lambda) \equiv \det(\lambda A - B) = 0$, are distinct. Thus their corresponding eigenvectors V_i are linearly independent. From the equalities $(\lambda_i A - B)V_i = 0$ and $(\lambda_j A - B)V_j = 0$, $0 \leq i < j \leq 3$, it follows that $\lambda_i V_i^T A V_j - V_i^T B V_j = 0$ and $\lambda_j V_i^T A V_j - V_i^T B V_j = 0$, respectively. Since $\lambda_i \neq \lambda_j$, we have $V_i^T A V_j = V_i^T B V_j = 0$. When $\lambda_1 = \lambda_0$, it can easily be shown [70] that the eigenspace of $A^{-1}B$ has dimension 2. Therefore, two linearly independent vectors V_0 and V_1 can be selected to be the eigenvectors associated with λ_2 such that $V_i^T A V_j = V_i^T B V_j = 0$ for $0 \leq i < j \leq 3$. Hence, the tetrahedron $[V_0 V_1 V_2 V_3]$ is self-polar with respect to both \mathcal{A} and \mathcal{B} [58, 63]. This means that, for both ellipsoids, each of the points V_i is the pole of a plane which passes through the other three vertices of the tetrahedron $[V_0 V_1 V_2 V_3]$ (see Figure 14).

It follows from $(\lambda_0 A - B)V_0 = 0$ that $\lambda_0 V_0^T A V_0 - V_0^T B V_0 = 0$, since $\lambda_0 > 0$, $V_0^T A V_0$ and $V_0^T B V_0$ have the same sign. If $V_0^T A V_0 < 0$ and $V_0^T B V_0 < 0$, then V_0 would be inside both \mathcal{A} and \mathcal{B} , and hence \mathcal{A} and \mathcal{B} would overlap. We deduce that $V_0^T A V_0 > 0$ and $V_0^T B V_0 > 0$: i.e., V_0 is outside \mathcal{A} and \mathcal{B} ; similarly, V_1 is outside \mathcal{A} and \mathcal{B} .

¹ $[V_0 V_1 V_2 V_3]$ is a self-polar tetrahedron for a quadric $X^T A X = 0$, if $V_i^T A V_j = 0$ for $i \neq j$. See [58], page 272.

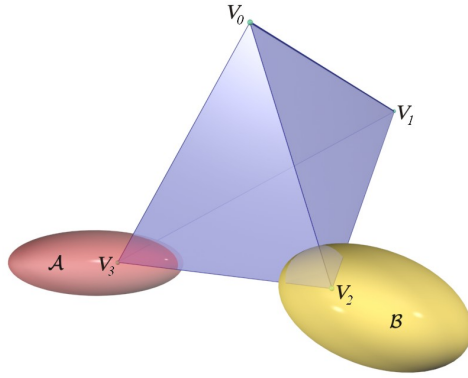


Figure 14: The tetrahedron $[V_0V_1V_2V_3]$.

It follows from $(\lambda_3A - B)V_3 = 0$ that $\lambda_3V_3^TAV_3 - V_3^TBV_3 = 0$. Since $\lambda_3 < 0$, $V_3^TAV_3$ and $V_3^TBV_3$ have different signs. We are going to show that $V_3^TAV_3 < 0$: i.e. that V_3 is inside \mathcal{A} .

Clearly, there is a point P on the line V_2V_3 which is outside both \mathcal{A} and \mathcal{B} . Thus $P = aV_2 + bV_3$ for some constants $a \neq 0$ and $b \neq 0$. Then $P^TAP = a^2V_2^TAV_2 + b^2V_3^TAV_3 > 0$, since P is outside \mathcal{A} . Similarly, $P^TBP = a^2V_2^TBV_2 + b^2V_3^TBV_3 > 0$. Since $(\lambda_2A - B)V_2 = 0$ and $(\lambda_3A - B)V_3 = 0$, we have $BV_2 = \lambda_2AV_2$ and $BV_3 = \lambda_3AV_3$. Substituting for BV_2 and BV_3 yields

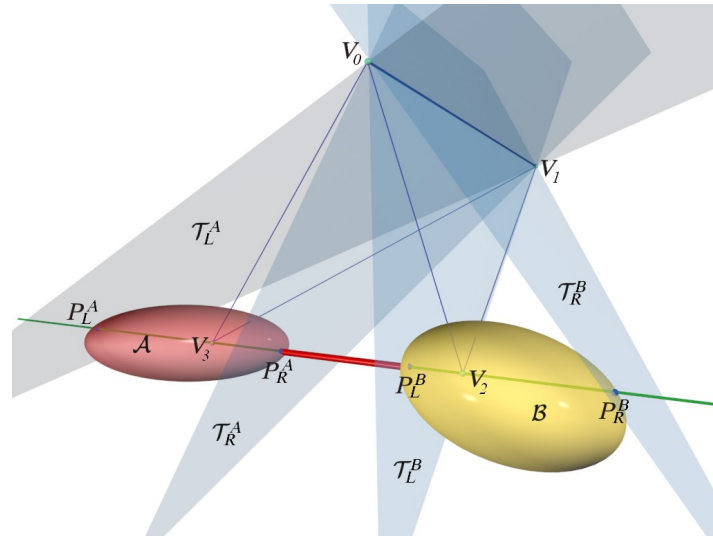
$$\begin{aligned}
 P^TBP &= a^2V_2^TBV_2 + b^2V_3^TBV_3 \\
 &= a^2V_2^T(\lambda_2AV_2) + b^2V_3^T(\lambda_3AV_3) \\
 &= \lambda_2(a^2V_2^TAV_2 + b^2V_3^TAV_3) + b^2(\lambda_3 - \lambda_2)V_3^TAV_3 \\
 &= \lambda_2P^TAP + b^2(\lambda_3 - \lambda_2)V_3^TAV_3.
 \end{aligned}$$

Thus

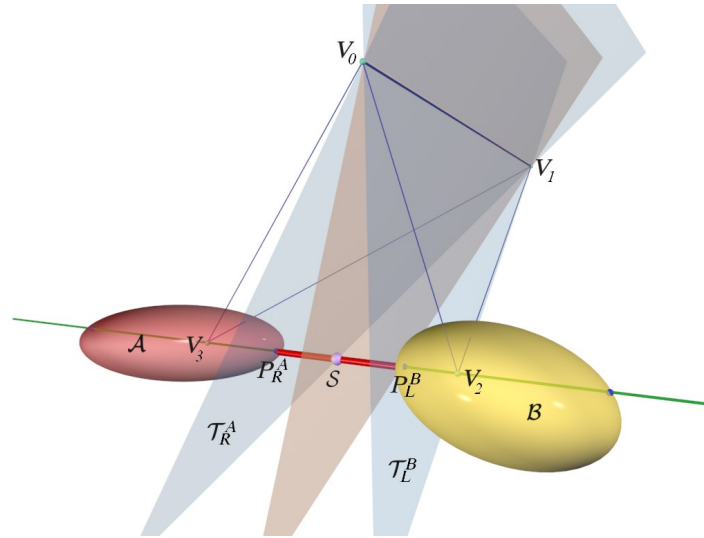
$$V_3^TAV_3 = b^{-2}(\lambda_3 - \lambda_2)^{-1}(P^TBP - \lambda_2P^TAP) < 0.$$

Hence, V_3 is inside \mathcal{A} . Similarly, it can be shown that V_2 is inside \mathcal{B} . This completes the proof. \square

Suppose that $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ are two disjoint ellipsoids. Since V_3 is inside \mathcal{A} , its polar plane $V_0 V_1 V_2$ does not intersect \mathcal{A} [58]; and, since V_2 is inside \mathcal{B} , its polar plane $V_0 V_1 V_3$ does not intersect \mathcal{B} . Thus, the line $V_0 V_1$, which is the intersection of the planes $V_0 V_1 V_2$ and $V_0 V_1 V_3$, does not intersect either \mathcal{A} or \mathcal{B} . So there are two planes, \mathcal{T}_L^A and \mathcal{T}_R^A , tangent to \mathcal{A} that pass through $V_0 V_1$ (see Figure 15(a)).



(a)



(b)

Figure 15: (a) Four planes passing through V_0 and V_1 : \mathcal{T}_L^A and \mathcal{T}_R^A are tangent to ellipsoid \mathcal{A} ; \mathcal{T}_L^B and \mathcal{T}_R^B are tangent to ellipsoid \mathcal{B} ; (b) A separating plane is one that passes through V_0 , V_1 and S , where S can be any point on the open line segment $(P_R^A P_L^B)$.

Let P_L^A and P_R^A denote the points at which \mathcal{T}_L^A and \mathcal{T}_R^A touch \mathcal{A} . Clearly, \mathcal{A} is contained entirely between the planes \mathcal{T}_L^A and \mathcal{T}_R^A . Since the tetrahedron $[V_0V_1V_2V_3]$ is self-polar with respect to \mathcal{A} , the line V_0V_1 is conjugate with the line V_2V_3 . Hence, the points P_L^A and P_R^A are on the line V_2V_3 : i.e. the line segment $[P_L^AP_R^A]$ is contained in V_2V_3 . Let P_L^B and P_R^B denote the points where the two tangent planes \mathcal{T}_L^B and \mathcal{T}_R^B , that pass through V_0V_1 touch the ellipsoid \mathcal{B} . By a similar argument to that just used for ellipsoid \mathcal{A} , we can show that \mathcal{B} is contained between the planes \mathcal{T}_L^B and \mathcal{T}_R^B , and the line segment $[P_L^BP_R^B]$ is contained in the line V_2V_3 .

Since \mathcal{A} and \mathcal{B} are disjoint, the line segments $[P_L^AP_R^A]$ and $[P_L^BP_R^B]$ are disjoint. Let the four tangent points $P_L^A, P_R^A, P_L^B,$ and P_R^B be labeled such that the open line segment $(P_R^AP_L^B)$ is outside \mathcal{A} and \mathcal{B} (see Figure 15). Clearly, any plane passing through V_0V_1 and intersecting $(P_R^AP_L^B)$ does not intersect \mathcal{A} or \mathcal{B} and lies between \mathcal{A} and \mathcal{B} ; hence, this plane separates \mathcal{A} and \mathcal{B} .

Therefore, we have proved

Theorem 4.6. *Any plane passing through the line V_0V_1 and intersecting the open line segment $(P_R^AP_L^B)$ is a separating plane of \mathcal{A} and \mathcal{B} (see Figure 15(b)). In particular, the plane passing through $V_0, V_1,$ and P_R^A touches \mathcal{A} at P_R^A , and the plane passing through $V_0, V_1,$ and P_L^B touches \mathcal{B} at P_L^B .*

Now we will consider the computational procedure for obtaining a separating plane for two disjoint ellipsoids \mathcal{A} and \mathcal{B} . Using Theorem 4.1, we first compute the four real roots of the characteristic equation $f(\lambda) = \det(\lambda A - B) = 0$: two of the roots are positive and the other two are distinct negative roots, and can be labeled according to the inequalities $\lambda_3 < \lambda_2 < 0 < \lambda_1 \leq \lambda_0$. Since $\det(\lambda A - B) = \det(A) \det(\lambda I - A^{-1}B)$, these roots λ_i are also the eigenvalues of the matrix $A^{-1}B$. The eigenvectors V_2 and V_3 are obtained by solving the equations $(\lambda_i I - A^{-1}B)X = 0, i = 2, 3$. Next we compute the points P_R^A and P_L^B where the line V_2V_3 intersects the ellipsoids \mathcal{A} and \mathcal{B} (see Figure 15). Then we obtain the tangent plane \mathcal{T}_R^A of \mathcal{A} at P_R^A and the tangent plane \mathcal{T}_L^B of \mathcal{B} at P_L^B . Since, by Theorem 4.6, \mathcal{T}_R^A and \mathcal{T}_L^B intersect on the line V_0V_1 , a separating plane can be found by taking an appropriate linear combination of the equations of \mathcal{T}_R^A and \mathcal{T}_L^B .

4.3 THE COMPLETE ALGORITHM

Based on the results discussed in the previous sections, we present below an algorithm for detecting collision between two moving ellipsoids whose motions are given by sampled positions in consecutive frames. Figure 16 gives a schematic description of the algorithm.

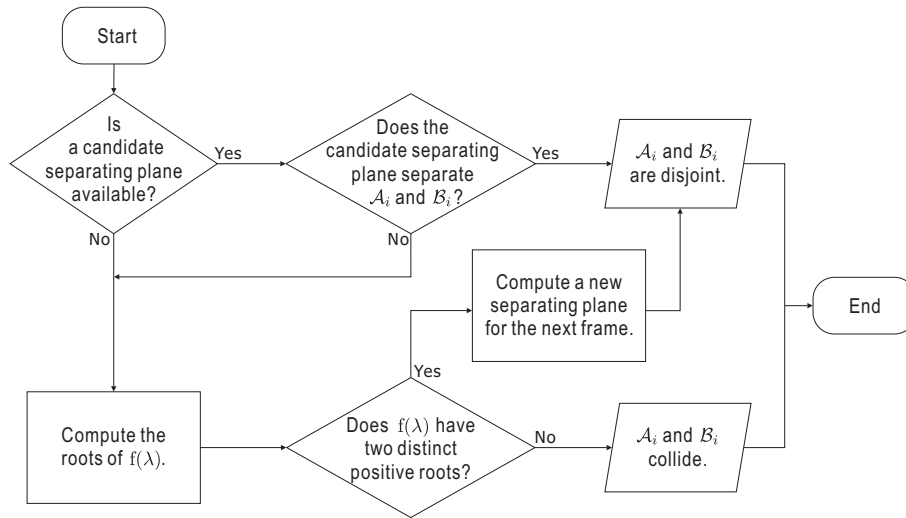


Figure 16: Algorithm for collision detection between two ellipsoids \mathcal{A}_i and \mathcal{B}_i at frame i .

Suppose that two ellipsoids \mathcal{A}_{i-1} and \mathcal{B}_{i-1} at frame $i-1$ are disjoint. The separating plane is computed and becomes the *candidate separating plane* at the next frame, i . Under continuous motion, the positions and orientations of the moving ellipsoids may be expected to change little between frame $i-1$ and frame i . Thus, in most cases, the candidate separating plane may still separate \mathcal{A}_i and \mathcal{B}_i , and this can be verified efficiently. If the candidate separating plane at frame i does indeed separate \mathcal{A}_i and \mathcal{B}_i , then a great deal of time is saved by avoiding solving the characteristic equation of \mathcal{A}_i and \mathcal{B}_i , which is a relatively expensive procedure; otherwise, we need to compute the roots of the characteristic equation, either to find that the ellipsoids collide, or to find that they are disjoint and then to compute a new separating plane. The algorithm is efficient because the non-collision case occurs much more frequently

than the collision case, due to inter-frame coherence. It is also possible to maintain more than one candidate separating plane, so as to increase the likelihood of detecting disjoint ellipsoids.

A plane separates two ellipsoids if and only if (i) the plane does not intersect either of the ellipsoids; and (ii) the centers of the two ellipsoids are on the opposite sides of the plane. Since it is relatively easy to test the second condition, the problem essentially reduces to that of testing whether a plane intersects an ellipsoid. Note that an ellipsoid can be transformed to the unit sphere by an affine transformation, which is normally available as the inverse of the motion matrix. Applying this transformation, the problem is simplified to one of detecting the intersection between a plane and a unit sphere centered at the origin. The problem can be further reduced to one of finding the distance between the origin and a plane.

The availability of a separating plane not only simplifies the collision test in many frames due to inter-frame coherence, but it also provides useful geometric information about the relative positions of the ellipsoids. For example, the tangent points P_R^A and P_L^B serve as good approximations to the pair of mutually closest points on the two ellipsoids; in fact, if the ellipsoids are actually spheres, then P_R^A and P_L^B are exactly the closest points. Furthermore, when two ellipsoids touch each other externally (i.e. $\lambda_2 = \lambda_3$), P_R^A and P_L^B merge into the contact point, which provides very useful information for computing collision impulses.

4.4 EXPERIMENTAL RESULTS

In this section we shall demonstrate the effectiveness of the separating plane in speeding up our method, and also compare our method with the GJK method [20]. For brevity, our new algorithm will be referred to as EECD (for Exact Ellipsoid Collision Detection.) To demonstrate the improved performance of EECD achieved by using the separating plane, two ellipsoids of the same size (with principal axes of half-lengths 3, 3 and 5) are made to rotate continuously about their centers for 10,000 frames. The distance between the two centers is 8. Three sets of experiments are carried out in which the ellipsoids rotate about a random axis with low, medium and high angular speeds

(angular increments of 0.1, 0.5 and 1 radians per frame.) The simulation was run on a PC equipped with a Pentium 4 2.26GHz processor and the results are shown in Table 3.

When the ellipsoids rotate at low angular speed, the orientations of the ellipsoids between successive frames differ only by a little and we see that the use of separating planes allows the algorithm to identify as many as 8,310 separations out of the overall 8,798 separations. As a consequence, the time taken for collision detection is significantly reduced. As the angular speed increases, there are fewer cases where the candidate separating plane reports separation, but a considerable amount of time is still saved. EECD takes on average less than 5 microseconds per frame to detect collisions between two ellipsoids. It is also obvious that, when the distance between the two ellipsoids increases, more candidate separating planes will remain valid and therefore the benefit of using separating planes becomes more remarkable. Optimal performance is achieved when the candidate separating plane remains valid for all frames.

The same sets of experiments are performed using a collision detection scheme that solves the characteristic equation in every frame without using separating planes. It takes about 8 microseconds to detect collision between two ellipsoids, longer than that needed by EECD. We note that the running time is insensitive to the number of collisions and separations in each test. This is because the same amount of computation is required no matter whether the ellipsoids are disjoint or overlapping.

Angular velocity	Low	Medium	High
Number of collisions (out of 10,000 frames)	1,202	1,554	1,622
Number of separations (out of 10,000 frames)	8,798	8,446	8,378
Number of separations reported by separating plane tests	8,310	6,029	4,502
Time per frame (μ s) using the separating plane, averaged over 10,000 frames (EECD)	2.474	3.985	4.982
Time per frame (μ s) without using the separating plane, averaged over 10,000 frames	7.906	7.951	7.899

Table 3: Experimental results demonstrating the effect of using the separating plane in EECD.

Another experiment was conducted to compare EECD with the enhanced GJK method, which has been described [43] as one of the most robust and efficient collision

detection methods for convex polyhedra currently available. We have adapted the routines for the enhanced GJK method ² by removing the code for distance computation, thus improving their efficiency for collision detection.

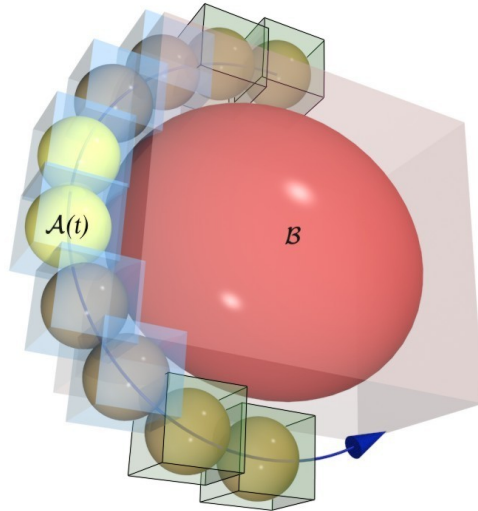


Figure 17: Experimental set-up corresponding to the results in Table 4: a sphere $\mathcal{A}(t)$ moves along a circular orbit around an ellipsoid \mathcal{B} . Bright spheres indicate collision. Framed boxes bounding $\mathcal{A}(t)$ are disjoint from the bounding box of \mathcal{B} .

The experiment is set up as follows (see Figure 17). A sphere $\mathcal{A}(t)$ of radius 1.0 orbits along a circular path around an ellipsoid \mathcal{B} . The half-lengths of the three principal axes of \mathcal{B} are 4.5, 4, and 2, respectively. The path of $\mathcal{A}(t)$ lies on the plane determined by the two longer principal axes of \mathcal{B} . $\mathcal{A}(t)$ and \mathcal{B} collide only near the two ends of the longest principal axis of \mathcal{B} , as shown by bright spheres in Figure 17. The central angle subtended by the displacement of $\mathcal{A}(t)$ between two consecutive frames is 0.01 radians. For each frame, the bounding boxes of $\mathcal{A}(t)$ and \mathcal{B} are first checked. If they do not intersect, separation is reported, shown by framed boxes bounding $\mathcal{A}(t)$ in the figure; otherwise, a collision detection procedure, either EECD or enhanced GJK, is called.

Out of a total of 13,746 frames, the bounding boxes of $\mathcal{A}(t)$ and \mathcal{B} intersect in 10,000 frames, and $\mathcal{A}(t)$ collides with \mathcal{B} in 1,157 of these frames. Collision detection for the 10,000 frames using EECD takes 0.0212 seconds. Among the 8,843 frames in which $\mathcal{A}(t)$

² The enhanced GJK routines are due to Stephen Cameron and are available at <http://users.comlab.ox.ac.uk/stephen.cameron/distances.html>.

and \mathcal{B} are disjoint, 8,020 frames are detected using separating planes. As a comparison, the NECD scheme, without using separating planes, takes 0.0763 seconds for the same set-up. Thus the use of separating planes reduces the computational time by a factor of more than 3 in this example. As explained earlier, this improvement is specific in this particular example, and varies with the orbit taken by $\mathcal{A}(t)$. In an extreme case, the two ellipsoids would collide in all frames and the separating planes may offer no benefit.

The enhanced GJK method was also run on the same set-up, with different numbers of vertices (n) used for approximating $\mathcal{A}(t)$ and \mathcal{B} . The results are summarized in Table 4.

	EECD	Enhanced GJK			
		$n=488$	$n=1,460$	$n=4,376$	$n=13,124$
Collision detected (frames)	1,157	0	435	971	1,111
Separation detected (frames)	8,843	10,000	9,565	9,029	8,889
Time per frame (μ s), averaged over 10,000 frames	2.12	15.2	23.6	36.8	54.3

Table 4: Computation time for our algorithm, EECD, and for the enhanced GJK method, with a varying number of vertices (n) used in the latter for polyhedral approximation of the ellipsoids.

The results show that EECD is about an order of magnitude quicker than the enhanced GJK method. When the ellipsoids are coarsely faceted, the enhanced GJK method, is also fast; however when $n = 488$, it incorrectly reports collision as separation in all frames. The number of misses reduces with increased resolution; when 13,124 vertices are used for polyhedral approximation, there are errors in only 46 frames, but the computation time is now increased from 0.152 seconds to 0.543 seconds. This trade-off between accuracy and efficiency is intrinsic to any collision detection methods based on polyhedral approximation.

4.5 AN ALTERNATIVE SOLUTION BY SUBRESULTANT SEQUENCE

We now present another method that involves only the evaluation of some expressions for determining whether two static ellipsoids are separate, touching externally or

overlap. Although this method requires more arithmetic operations, but it has its own value in solving the collision detection problem of two continuously moving ellipsoids or quadrics as we shall see in Chapter 6. To facilitate discussions in association to the classic theorems from the theory of equations, given two ellipsoids $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$ in \mathbb{R}^3 , we shall assume that the characteristic polynomial of \mathcal{A} and \mathcal{B} is given by

$$\hat{f}(\lambda) = \det(\lambda A + B).$$

Hence, we have the followings [70]:

- i. $\hat{f}(\lambda) = 0$ has at least two negative roots.
- ii. $\hat{f}(\lambda) = 0$ has two distinct positive roots if and only if \mathcal{A} and \mathcal{B} are separate.
- iii. $\hat{f}(\lambda) = 0$ has a positive double root if and only if \mathcal{A} and \mathcal{B} touch externally.

Rewrite $\hat{f}(\lambda)$ in monomial form, we have

$$\hat{f}(\lambda) = c_4 \lambda^4 + c_3 \lambda^3 + c_2 \lambda^2 + c_1 \lambda + c_0.$$

Since we assume that $\det(A) < 0$ and $\det(B) < 0$, we also have

$$c_4 < 0 \quad \text{and} \quad c_0 < 0.$$

We then consider the polynomial

$$f(\lambda) = \frac{\hat{f}(\lambda)}{c_4} = \lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d$$

with $d > 0$. Clearly, $f(\lambda)$ has the same zeros as $\hat{f}(\lambda)$.

4.5.1 Conditions for separation and external touching

The ellipsoids \mathcal{A} and \mathcal{B} are either separate or externally touching if and only if the following conditions are satisfied:

(I) $f(\lambda) = 0$ has 4 real roots, counted with multiplicity.

(II) $f(\lambda) = 0$ has 2 positive roots, counted with multiplicity.

We consider the signed subresultant sequence of f and f' [3]:

$$\mathcal{P} = f, f', sr_2(\lambda), sr_1(\lambda), \text{res}(f, f') \quad (4.2)$$

where

$$f = \lambda^4 + a\lambda^3 + b\lambda^2 + c\lambda + d,$$

$$f' = 4\lambda^3 + 3a\lambda^2 + 2b\lambda + c,$$

$$sr_2(\lambda) = (-8b + 3a^2)\lambda^2 + (2ab - 12c)\lambda + ac - 16d,$$

$$sr_1(\lambda) = (-6a^3c + 2a^2b^2 - 12a^2d + 28abc - 8b^3 - 36c^2 + 32bd)\lambda - 9a^3d \\ + a^2bc + 3ac^2 + 32abd - 4b^2c - 48cd,$$

$$\text{and } \text{res}(f, f') = -27a^4d^2 - 4a^3c^3 + 18a^3bcd - 4a^2b^3d - 6a^2c^2d + 144a^2bd^2 + a^2b^2c^2 \\ - 192acd^2 - 80ab^2cd + 18abc^3 + 144bc^2d - 128b^2d^2 + 256d^3 \\ - 27c^4 - 4b^3c^2 + 16b^4d$$

We also write $sr_2(\lambda) = sr_{22}\lambda^2 + sr_{12}\lambda + sr_{02}$, $sr_1(\lambda) = sr_{11}\lambda + sr_{01}$ and $\text{res} = \text{res}(f, f')$ for simplicity.

Now, condition (I) is satisfied, i.e., $f(\lambda) = 0$ has 4 real roots, counting multiplicity, if and only if any of the following three cases is true

(1) $f(\lambda) = 0$ has 4 distinct simple roots, i.e., $\text{res} \neq 0$ and $\text{Var}(\mathcal{P}; -\infty, +\infty) = 4$; or

(2) $f(\lambda) = 0$ has 1 double root and 2 simple roots, i.e., $\text{res} = 0$ and $sr_1(\lambda) \neq 0$ and $\text{Var}(\mathcal{P}; -\infty, +\infty) = 3$; or

(3) $f(\lambda) = 0$ has 2 double roots, i.e., $\text{res} = 0$ and $sr_1(\lambda) = 0$ and $sr_2(\lambda) \neq 0$ and $\text{Var}(\mathcal{P}; -\infty, +\infty) = 2$

The other cases in which $f(\lambda) = 0$ has a root of multiplicity greater than 2 are not considered, since $f(\lambda) = 0$ must not have any positive root (i.e., condition (II) must not be satisfied) as $f(\lambda) = 0$ has at least 2 negative roots and that $d > 0$.

In case (1), we have $\text{Var}(\mathcal{P}; -\infty, +\infty) = 4$ and hence $\text{Var}(\mathcal{P}; -\infty) = 4$ and $\text{Var}(\mathcal{P}; \infty) = 0$. We therefore have $\text{Var}(1, 4, sr_{22}, sr_{11}, \text{res}) = 0$, which implies that $sr_{22} > 0$, $sr_{11} > 0$ and $\text{res} > 0$. Using similar derivation, the above three cases are equivalently to

(1) $sr_{22} > 0$, $sr_{11} > 0$, $\text{res} > 0$; or

(2) $sr_{22} > 0$, $sr_{11} > 0$, $\text{res} = 0$; or

(3) $sr_{22} > 0$, $sr_{11} = 0$, $sr_{01} = 0$, $\text{res} = 0$

Furthermore, suppose now that (I) is satisfied, i.e., $f(\lambda) = 0$ has 4 real roots counting multiplicity. Then (II) is also satisfied, i.e., $f(\lambda) = 0$ has 2 positive roots, if and only if

$$\text{Var}(1, a, b, c, d) = 2$$

Since it is not possible to have $\text{Var}(1, a, b, c, d) = 4$ (which means all real roots of $f(\lambda) = 0$ are positive), we have

$$\text{Var}(1, a, b, c, d) = 2 \quad \text{iff} \quad a < 0 \quad \text{or} \quad b < 0 \quad \text{or} \quad c < 0.$$

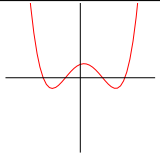
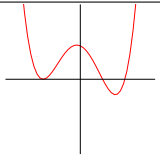
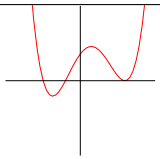
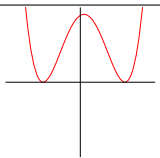
4.5.2 Differentiating the separation and touching cases

We now further distinguish between the cases of separation and external touching. Hence, for the three cases (1), (2) and (3), $f(\lambda) = 0$ also have 2 positive roots.

For case (1), $f(\lambda) = 0$ has 4 distinct simple roots and hence the two positive roots must be distinct. Therefore, this corresponds to two separating ellipsoids.

For case (2), $f(\lambda) = 0$ has 1 double root and 2 simple roots, and the double root λ_0 is given by $sr_1(\lambda)$ which is the gcd of f and f' . Given that $sr_{11} > 0$, if $sr_{01} > 0$, then $\lambda_0 < 0$ and $f(\lambda) = 0$ has two distinct positive roots corresponding to separation; if $sr_{01} < 0$, then $\lambda_0 < 0$ and $f(\lambda) = 0$ has a positive double root corresponding to external touching.

For case (3), $f(\lambda) = 0$ has 2 double roots. Given that there are 2 negative and 2 positive roots, $f(\lambda) = 0$ must have a positive double root which corresponds to two externally touching ellipsoids.

Case		f	sr ₂ (λ)		res	ellipsoid status	
		a/b/c	sr ₂₂	sr ₁₁			sr ₁₀
(1)		< 0	> 0	> 0	> 0	separate	
(2)		< 0	> 0	> 0	> 0	= 0	separate
		< 0	> 0	> 0	< 0	= 0	touching
(3)		< 0	> 0	= 0	= 0	= 0	touching

4.5.3 A summary

The algebraic condition for two separating ellipsoids is:

$$(a < 0 \text{ or } b < 0 \text{ or } c < 0), \text{ sr}_{22} > 0, \text{ sr}_{11} > 0, (\text{res} > 0 \text{ or } (\text{res} = 0 \text{ and } \text{sr}_{10} > 0))$$

The algebraic condition for two externally touching ellipsoids is:

$$(a < 0 \text{ or } b < 0 \text{ or } c < 0), \text{ sr}_{22} > 0, \text{ res} = 0, ((\text{sr}_{11} > 0, \text{sr}_{10} < 0) \text{ or } (\text{sr}_{11} = 0, \text{sr}_{10} = 0))$$

where

$$\begin{aligned}
\text{sr}_{22} &= -8b + 3a^2, \\
\text{sr}_{11} &= -6a^3c + 2a^2b^2 - 12a^2d + 28abc - 8b^3 - 36c^2 + 32bd, \\
\text{sr}_{10} &= -9a^3d + a^2bc + 3ac^2 + 32abd - 4b^2c - 48cd, \\
\text{res} &= -27a^4d^2 - 4a^3c^3 + 18a^3bcd - 4a^2b^3d - 6a^2c^2d + 144a^2bd^2 + a^2b^2c^2 - 192acd^2 \\
&\quad - 80ab^2cd + 18abc^3 + 144bc^2d - 128b^2d^2 + 256d^3 - 27c^4 - 4b^3c^2 + 16b^4d
\end{aligned}$$

4.5.4 Computation costs

The algebraic conditions involve the evaluation of four polynomials, namely, sr_{22} , sr_{11} , sr_{10} and res . By definition of the subresultants, these polynomials are the determinants of some matrices defined by the coefficients of $f(\lambda)$ and $f'(\lambda)$. These matrices share common submatrices and therefore, the evaluation of the polynomials could be simplified as follows:

$$\begin{aligned}
\text{sr}_{22} &= -8b + 3a^2 \\
E_1 &= -2ab + 12c \\
E_2 &= 4b^2 + 9ac \\
E_3 &= -b \cdot \text{sr}_{22} - aE_1 + E_2 \\
\text{sr}_{01} &= -3ad \cdot \text{sr}_{22} - 4dE_1 - cE_3 \\
E_4 &= -ac + 16d \\
E_5 &= -c \cdot \text{sr}_{22} - aE_4 - 2bc + 12ad \\
E_6 &= cE_1 - bE_4 - 3c^2 + 8bd \\
\text{sr}_{11} &= -2bE_3 + 3aE_5 - 4E_6 \\
E_7 &= cE_5 + d(2b \cdot \text{sr}_{22} + 4E_4) \\
E_8 &= cE_6 - d(2bE_1 - 3aE_4) \\
\text{res} &= d(-2b \cdot \text{sr}_{11} + 3a \cdot \text{sr}_{01} + 4E_7) - c(-c \cdot \text{sr}_{11} + b \cdot \text{sr}_{01} + aE_7 - E_8)
\end{aligned}$$

The above expressions can be computed efficiently using 42 multiplications and 27 additions using the following code:

```
t1 = -3*a;
t2 = -(b + b);
t3 = a*c;
sr22 = -t1*a - 8*b;
t4 = sr22*b;
E1 = a*t2 + 12*c;
E2 = 4*b*b + 9*t3;
E3 = -t4 - a*E1 + E2;
sr01 = -c*E3 + d*(sr22*t1 - 4*E1);
E4 = -t3 + 16*d;
E5 = (- sr22 + t2)*c + (-E4 + 12*d)*a;
E6 = (E1 - 3*c)*c + (-E4 + 8*d)*b;
sr11 = E3*t2 - t1*E5 - 4*E6;
E7 = c*E5 - d*(-2*t4 - 4*E4);
E8 = c*E6 - d*(-t2*E1 + E4*t1);
res = d*(sr11*t2 - sr01*t1 + t*E7) - c*(-c*sr11 + b*sr01 + a*E7 - E8);
```


5

MOVING ELLIPSOIDS WITH CONTINUOUS MOTION

The continuous collision detection for moving ellipsoids in 3D space is far more complex than that for moving ellipses in 2D plane. We discussed in Chapter 2 an algebraic approach in solving the CCD problem for moving ellipses, where a univariate polynomial is formulated whose roots correspond to the time instants at which the ellipses are in internal or external touch. For moving ellipsoids, however, the same approach of relying on detecting the roots of the univariate polynomial is infeasible. Unlike the situation with elliptic disks (see Corollary 2.8), in 3D the characteristic equation $f(\lambda) = 0$ may have a positive double root for a pair of separate ellipsoids (compare the characteristic equation of two separate spheres, which always has a double root $\lambda = 1$.) Thus, if we were to rely on detecting real zeros of the discriminant for collision detection between moving ellipsoids, our algorithm would certainly fail (i.e., have a false positive) because a zero of the discriminant can be caused by a positive double root, which does not correspond to an external contact between the ellipsoids.

Based on the algebraic condition of Wang et al. [70] for the separation of two stationary ellipsoids (cf. Theorem 4.1), we proposed a method that reduces the CCD problem for two moving ellipsoids to an analysis of the zero set of a bivariate polynomial equation which has high degree in the time parameter t . In this chapter we shall present a new efficient numerical method to solve the CCD problem in real-time, thus bringing ellipsoid-based continuous collision detection into the realm in computer graphics for time critical applications. This is achieved by exploring the special structure of the bivariate function under consideration and employing several novel and efficient search techniques. It is assumed throughout that the motions of

moving ellipsoids, either Euclidean or affine, are expressible as rational functions of the time parameter t .

5.1 3D RATIONAL EUCLIDEAN AND AFFINE MOTIONS

A rational Euclidean motion in \mathbb{E}^3 is given by

$$M(t) = \begin{pmatrix} R(t) & V(t) \\ \mathbf{0}^T & 1 \end{pmatrix}, \quad (5.1)$$

where $V(t) \in \mathbb{E}^3$, $R(t)$ a 3×3 orthogonal matrix, and t can be considered as a parameter of time. The motion is a composition of a rotation $R(t)$ acting upon a point in \mathbb{E}^3 , followed by a translation $V(t)$. All rational Euclidean motions can be represented in (5.1) with

$$V(t) = \left(\frac{v_0}{v_3}, \frac{v_1}{v_3}, \frac{v_2}{v_3} \right)^T, \quad \text{and}$$

$$R(t) = \frac{1}{E} \begin{pmatrix} e_0^2 + e_1^2 - e_2^2 - e_3^2 & 2e_1e_2 - 2e_0e_3 & 2e_0e_2 + 2e_1e_3 \\ 2e_0e_3 + 2e_1e_2 & e_0^2 - e_1^2 + e_2^2 - e_3^2 & 2e_2e_3 - 2e_0e_1 \\ -2e_0e_2 + 2e_1e_3 & 2e_0e_1 + 2e_2e_3 & e_0^2 - e_1^2 - e_2^2 + e_3^2 \end{pmatrix}$$

where $E = e_0^2 + e_1^2 + e_2^2 + e_3^2$ and $v_0, \dots, v_3, e_0, \dots, e_3$ are polynomials in t [30]. The Euler parameters e_0, e_1, e_2, e_3 describe a rotation about a vector in \mathbb{E}^3 and are called the *normalized Euler parameters* when $E = 1$. Readers are referred to [54] for a survey on rational motion design and [24, 29, 30] for interpolating a set of positions in \mathbb{E}^3 using piecewise B-spline motions.

When the entries of $V(t)$ and $R(t)$ are rational polynomials of maximal degree k , we called $M(t)$ a *rational motion of degree k* . An ellipsoid $\mathcal{A}(t)$ moving under a rational motion $M(t)$ is represented as $X^T A(t) X = 0$, where $A(t) = (M^{-1}(t))^T A M^{-1}(t)$.

Assume that the maximal degree of the entries in $R(t)$ and $V(t)$ are k_R and k_V , respectively. Then,

$$M^{-1}(t) = \begin{pmatrix} R(t)^T_{\langle k_R \rangle} & -R(t)^T V(t)_{\langle k_R+k_V \rangle} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

and

$$A(t) = \begin{pmatrix} P(t)_{\langle 2k_R \rangle} & U(t)_{\langle 2k_R+k_V \rangle} \\ U(t)^T_{\langle 2k_R+k_V \rangle} & s(t)_{\langle 2(k_R+k_V) \rangle} \end{pmatrix}$$

for some 3×3 matrix $P(t)$, 3-vector $U(t)$, and scalar function $s(t)$. Here, the bracketed subscript represents the maximal degree of the entries of the associated entity.

For a rational affine motion in \mathbb{E}^3 , the motion matrix $M(t)$ is formed by replacing $R(t)$ in (5.1) by a 3×3 non-singular matrix $L(t)$. The motion is then a composition of a linear transformation $L(t)$ acting upon a point in \mathbb{E}^3 , followed by a translation $V(t)$. Assume that the maximal degree of the entries in $L(t)$ and $V(t)$ are k_L and k_V , respectively. Here,

$$A(t) = \begin{pmatrix} P(t)_{\langle 6k_L \rangle} & U(t)_{\langle 6k_L+k_V \rangle} \\ U(t)^T_{\langle 6k_L+k_V \rangle} & s(t)_{\langle 6k_L+2k_V \rangle} \end{pmatrix}$$

for some 3×3 matrix $P(t)$, 3-vector $U(t)$, and scalar function $s(t)$.

5.2 AN AFFINE MOTION INTERPOLANT

Assume that an ellipsoid $\mathcal{A}(t)$ is under a motion $M_A(t)$, and another ellipsoid $\mathcal{B}(t)$ is similarly under a motion $M_B(t)$, for $t \in [t_0, t_1]$. The two ellipsoids $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide if and only if the standard ellipsoid \mathcal{A} collides with the moving ellipsoid $\hat{\mathcal{B}}(t)$

under a relative motion $\hat{M}_B(t) = M_A^{-1}(t)M_B(t)$, for $t \in [t_0, t_1]$. The ellipsoid $\hat{\mathcal{B}}(t_0)$ with its center \mathbf{c}_0 is represented as follows:

$$\hat{\mathcal{B}}(t_0) : (\mathbf{x} - \mathbf{c}_0)^T B_0 (\mathbf{x} - \mathbf{c}_0) = 1,$$

where B_0 is a symmetric positive definite matrix and $\mathbf{x} = (x, y, z)^T$ is a point on the ellipsoid $\hat{\mathcal{B}}(t_0)$. Similarly, the ellipsoid $\hat{\mathcal{B}}(t_1)$ with its center \mathbf{c}_1 is represented as follows:

$$\hat{\mathcal{B}}(t_1) : (\mathbf{x} - \mathbf{c}_1)^T B_1 (\mathbf{x} - \mathbf{c}_1) = 1,$$

where B_1 is a symmetric positive definite matrix. Now we consider an interpolation $\hat{\mathcal{B}}(t)$ between the two ellipsoids

$$\hat{\mathcal{B}}(t) : (\mathbf{x} - \mathbf{c}(t))^T B(t) (\mathbf{x} - \mathbf{c}(t)) = 1, \quad (5.2)$$

where

$$\mathbf{c}(t) = \frac{t_1 - t}{t_1 - t_0} \mathbf{c}_0 + \frac{t - t_0}{t_1 - t_0} \mathbf{c}_1,$$

$$B(t) = \frac{t_1 - t}{t_1 - t_0} B_0 + \frac{t - t_0}{t_1 - t_0} B_1.$$

Note that $B(t)$ is symmetric positive definite if B_0 and B_1 are both symmetric and positive definite. Thus $\hat{\mathcal{B}}(t)$ represents a moving ellipsoid under an affine motion, for $t_0 \leq t \leq t_1$.

By expanding the above representation of the ellipsoid $\hat{\mathcal{B}}(t)$, we get

$$\hat{\mathcal{B}}(t) : \mathbf{x}^T B(t) \mathbf{x} - 2\mathbf{x}^T B(t) \mathbf{c}(t) + \mathbf{c}(t)^T B(t) \mathbf{c}(t) = 1.$$

Using the homogeneous coordinates $X = (x, y, z, w)^T$, we can represent the ellipsoid $\hat{\mathcal{B}}(t)$ as follows

$$\hat{\mathcal{B}}(t) : X^T \hat{B}(t) X = 0,$$

where

$$\hat{B}(t) = \begin{pmatrix} B(t) & -B(t)\mathbf{c}(t) \\ -\mathbf{c}(t)^T B(t)^T & \mathbf{c}(t)^T B(t)\mathbf{c}(t) - 1 \end{pmatrix}. \quad (5.3)$$

From this formula, we can easily show that the characteristic polynomial $f(\lambda, t) = \det(\lambda A - \hat{B}(t))$ has degree 6 in t .

5.3 CCD EQUATION FOR MOVING ELLIPSOIDS

We now present an efficient algorithm for continuous collision detection (CCD) between two moving ellipsoids: $\mathcal{A}(t) : X^T A(t) X = 0$ and $\mathcal{B}(t) : X^T B(t) X = 0$. Here the ellipsoids may move under affine deformations, including the commonly used Euclidean rigid motions as a special case.

We first introduce the continuous collision detection (CCD) equation of the two moving ellipsoids $\mathcal{A}(t)$ and $\mathcal{B}(t)$. This CCD equation is simply the characteristic equation of $\mathcal{A}(t)$ and $\mathcal{B}(t)$, $f(\lambda; t) = \det(\lambda A(t) - B(t)) = 0$, $t \in [0, 1]$. We will call $f(\lambda; t)$ the *CCD function*. The graph of the typical CCD function is shown in Fig. 18(a) for two collision-free moving ellipsoids, and in Fig. 18(b) for two colliding moving ellipsoids. (Note that λ is replaced by by a function of u as discussed below.)

Our CCD algorithm exploits some special features of the zero set of the CCD equation. Consider a fixed time $t_0 \in [0, 1]$. If $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ are separate, according to the discussions in Section 4.1, $f(\lambda; t_0) = 0$ has two negative real roots, that is, the line $t = t_0$ has two intersection points with the zero set of $f(\lambda; t)$ in the half plane $\lambda < 0$. If $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ overlap, $f(\lambda; t_0) = 0$ has no negative real root, that is, the line $t = t_0$ has no intersection point with the zero set of $f(\lambda; t)$ in the infinite strip $(-\infty, 0] \times [0, 1]$. Finally, if $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ are externally tangent, the line $t = t_0$ has a tangential intersection (i.e., a double intersection point) with the zero set of $f(\lambda; t)$ in the half plane $\lambda < 0$.

To facilitate numerical processing, we use the reparameterization $\lambda = \frac{u-1}{u}$ to map the variable $\lambda \in (-\infty, 0]$ to $u \in (0, 1]$, therefore the infinite strip $(\lambda, t) \in (-\infty, 0] \times [0, 1]$

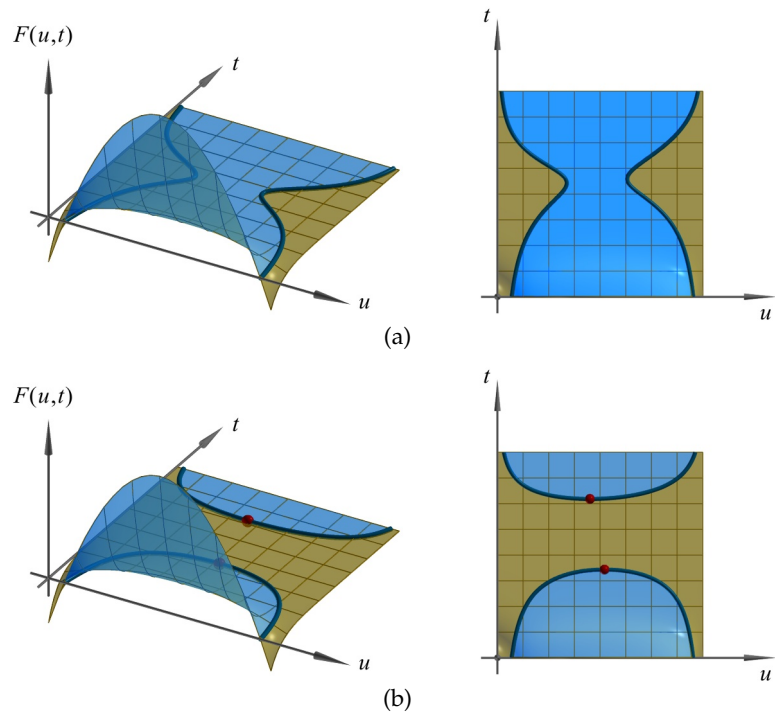


Figure 18: (a) A CCD function for two collision-free ellipsoids; (b) A CCD function for two colliding ellipsoids. The blue and yellow regions are where $F(u,t) > 0$ and $F(u,t) < 0$, respectively. The zero set $F(u,t) = 0$ is given by the dark blue curve. The red points in (b) represent the moments when the ellipsoids are in external contact.

is mapped to the region $(u, t) \in (0, 1] \times [0, 1]$. This mapping preserves the structure of $f(\lambda; t) = 0$ in the sense that the number of intersections between a horizontal line $t = t_0$ and the zero set of $f(\lambda; t) = 0$ is the same as that between the line $t = t_0$ and the zero set of $f(\lambda(u); t) = 0$. Clearly, the transformed characteristic equation $f(\lambda(u); t) = 0$ has the same zero set as the equation

$$\hat{F}(u, t) \equiv \det((u - 1)A(t) - uB(t)) = 0, \quad (u, t) \in (0, 1] \times [0, 1].$$

Recall that the elements of $A(t)$ and $B(t)$ are rational functions of t . Since we are only interested in the zero set of $\hat{F}(u, t)$, we use $F(u, t)$ to denote the bivariate polynomial after cleaning up the common denominator in $\hat{F}(u, t)$. Clearly, $F(u, t)$ and $\hat{F}(u, t)$ have the same zero set. Furthermore, to improve numerical robustness we represent $F(u, t)$ in the Bernstein form. From now on, we will also call $F(u, t) = 0$ the CCD equation.

Based on the preceding discussion and notation, we have the following theorems:

Theorem 5.1. *Any horizontal line $t = t_0 \in [0, 1]$ has at most two intersections with the zero set of $F(u, t)$ in the region $(0, 1] \times [0, 1]$. In particular,*

1. $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ are separate if and only if the line $t = t_0$ intersects the zero set of $F(u, t)$ in two distinct points in $(0, 1] \times [0, 1]$;
2. the interiors of $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ intersect if and only if the line $t = t_0$ does not intersect the zero set of $F(u, t)$ in $(0, 1] \times [0, 1]$;
3. $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ are externally tangent if and only if the line $t = t_0$ has a double intersection point with the zero set of $F(u, t)$ in $(0, 1] \times [0, 1]$.

The next theorem is fundamental to our CCD algorithm.

Theorem 5.2. *Let $\mathcal{A}(t)$ and $\mathcal{B}(t)$ be two moving ellipsoids in continuous motion in $t \in [0, 1]$. Suppose that at $t = 0$, the ellipsoids $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate. Then $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide in $t \in [0, 1]$ if and only if there exists a time t_0 in $[0, 1]$ such that the line $t = t_0$ has a double intersection point (u_0, t_0) with the zero set of $F(u, t)$ in the region $(0, 1] \times [0, 1]$.*

Proof. Suppose that there exists a time t_0 in $[0, 1]$ such that the line $t = t_0$ intersects the zero set of $F(t)$ at a double point in the region $[0, 1] \times [0, 1]$. Then, by Theorem

5.1, $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ touch each other externally. Therefore, $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide in $t \in [0, 1]$.

Now consider necessity. Suppose that $\mathcal{A}(t)$ and $\mathcal{B}(t)$ collide in $t \in [0, 1]$. Then either $\mathcal{A}(t)$ and $\mathcal{B}(t)$ touch each other externally at some time t_0 in $[0, 1]$ or $\mathcal{A}(t)$ and $\mathcal{B}(t)$ overlap with each other at some time $t_1 \in [0, 1]$. In the former case, we are done. In the latter case, since $\mathcal{A}(t)$ and $\mathcal{B}(t)$ are undergoing continuous motions and they are separate at $t = 0$, there exists time $t_0 \in [0, t_1]$ such that $\mathcal{A}(t)$ and $\mathcal{B}(t)$ touch each other externally at t_0 . The proof is completed. \square

Theorem 5.2 suggests how to detect whether two moving ellipsoids collide. First we may check if $\mathcal{A}(0)$ and $\mathcal{B}(0)$ are separate, using the procedure in Section 4.1. If not, we are done; if yes, we need to check if there exists a time t_0 in $[0, 1]$ such that the line $t = t_0$ has a double intersection point (u_0, t_0) with the zero set of $F(u, t)$ in $(0, 1] \times [0, 1]$. Clearly, such a point (u_0, t_0) is a solution of the equations $F(u, t) = F_u(u, t) = 0$, where $F_u(u, t)$ denotes $\partial F(u, t) / \partial u$. To find all the collision intervals, we note that whenever the collision status of two ellipsoids switches from separation to overlap (or vice versa), there must be a time instant at which the ellipsoids are in external contact; and hence the key task of our collision detection algorithm now is to detect all real solutions of $F(u, t) = F_u(u, t) = 0$ in the region $(u, t) \in (0, 1] \times [0, 1]$.

5.3.1 Solving the CCD equation

So far we have given an algebraic formulation of the problem under consideration. Now we shall present a numerical method based on this formulation. Given two moving ellipsoids over time $[0, 1]$, if they are separate throughout a time interval $(t_0, t_1) \subseteq [0, 1]$, then the interval (t_0, t_1) is called a *separation interval (SI)*. A separation interval (t_0, t_1) is called a *maximal separation interval* if (1) the two ellipsoids contact each other at t_0 or $t_0 = 0$; and (2) the two ellipsoids contact each other at t_1 or $t_1 = 1$. If the ellipsoids overlap throughout the interval (t_0, t_1) , then (t_0, t_1) is called an *overlapping interval (OI)*. Similarly, we can define the *maximal overlapping interval*. An interval $(t_0, t_1) \subset [0, 1]$ that is neither a separation interval nor an overlapping interval is called

a *mixed interval (MI)*. Our goal is to identify all the maximal separation intervals and maximal overlapping intervals.

By solving the CCD equation, we mean determining all *contact instants* at which the two ellipsoids are in external contact. Clearly, these instants define the endpoints of all the maximal separation intervals and maximal overlapping intervals. The contact instants correspond to the *critical points* in the zero set of the CCD equation—a solution (u^*, t^*) of $F(u, t) = 0$ is said to be a *critical point* if it further satisfies $F_u(u^*, t^*) = 0$. In this case the contact instant is t^* .

Basic idea

The idea of our algorithm is to subdivide recursively the motion interval $[0, 1]$ into a number of small intervals which can be confirmed to be either SI or OI. Then these intervals can be merged to form maximal separation intervals and maximal overlapping intervals.

During the process of our algorithm, for each interval (t_1, t_2) under consideration, we first determine the collision status of the two ellipsoids at the two endpoints of the interval. The interval (t_1, t_2) is temporarily labeled as a *candidate separation interval (CSI)* if the two ellipsoids are either separate or touching at t_1 and t_2 (Fig. 19(a)), since such an interval may be a separation interval in this case. Similarly, an interval (t_1, t_2) is temporarily labeled as a *candidate overlapping interval (COI)*, if the two ellipsoids are either overlap or touching at t_1 and t_2 (Fig. 19(b)). Further processing is needed to confirm whether a CSI is an SI, or a COI is an OI.

If the two moving ellipsoids have different collision status (either separate or collide) at t_1 and t_2 (Fig. 19(c)), then (t_1, t_2) is a mixed interval (MI). In this case, we will find a contact moment t^* in (t_1, t_2) , and use it to subdivide (t_1, t_2) into two intervals (t_1, t^*) and (t^*, t_2) . Evidently, one of the two intervals is a CSI and the other is a COI.

In the following we are going to devise robust tests to determine definitely whether a CSI (or COI) is a separation (or overlapping) interval. If the collision status over the entire interval is confirmed, we are done and the interval is labeled as an SI or OI. Otherwise, the interval will be subdivided at some contact time t^* so that we will work on the resulting smaller intervals in a recursive manner, until the collision status of the ellipsoids in all subintervals can be confirmed.

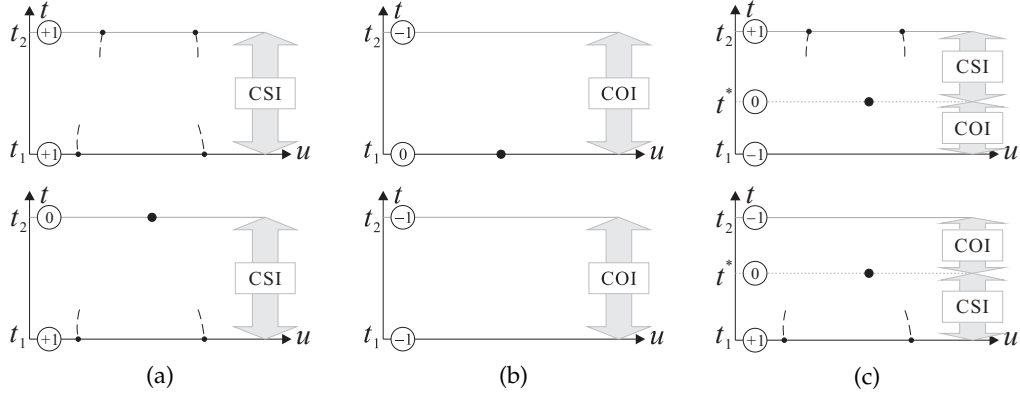


Figure 19: Examples of intervals classified as (a) a candidate separation interval (CSI); (b) a candidate overlapping interval (COI); and (c) mixed intervals with different collision status at t_1 and t_2 will be divided at a contact instant t^* and each of the two subintervals being classified as either a CSI or a COI. The circled numbers are the collision status at a particular time instant (+1 for separation, 0 for external contact and -1 for overlapping).

To determine the collision status of two ellipsoids at a particular time t_0 , we introduce the following *state function*:

$$\text{State}(t_0) = \begin{cases} +1 & \text{if } \max_u F(u, t_0) > 0, \text{ i.e., separate;} \\ 0 & \text{if } \max_u F(u, t_0) = 0, \text{ i.e., touching;} \\ -1 & \text{if } \max_u F(u, t_0) < 0, \text{ i.e., overlap.} \end{cases}$$

Instead of using the efficient method in Section 4.1.2, this function makes use of the sign of $\max_u F(u, t_0)$ to check the collision status of two static ellipsoids, whose value can be found by solving the cubic equation $F_u(u, t) = 0$ and can be reused in other steps of the algorithm, e.g., for the computation of contact time as discussed below. Here, $\text{State}(t_0) = 0$ if and only if (u_0, t_0) is a critical point for some $u_0 \in (0, 1]$.

We now describe our algorithm in details.

Initialization

We start by classifying the initial interval $[0, 1]$ as a CSI or a COI, depending on the collision status at $t = 0$ and $t = 1$. If the collision status at $t = 0$ and $t = 1$ are different, we compute a contact instant t^* (corresponding to a critical point (u^*, t^*)) where the ellipsoids are in external contact, using the following operation:

- **ContactTime:** This is to determine a contact instant in an interval $[t_1, t_2]$, when the collision status of the ellipsoids at t_1 and t_2 are different. It is done by a binary search in t to find $t^* \in [t_1, t_2]$ such that $\text{State}(t^*) = 0$. We then have $\text{ContactTime}(t_1, t_2) = t^*$. (In the binary search, we take into account the local maximum values $\max_u F(u, t_1)$ and $\max_u F(u, t_2)$. But we omit the details here.)

We then subdivide $[0, 1]$ into two smaller intervals $[0, t^*)$ and $(t^*, 1]$, and classify each as a CSI or a COI (as in Fig. 19(c) with $t_1 = 0$ and $t_2 = 1$). The procedures are given in the following algorithm:

Algorithm 1 Initialization

Input: $F(u, t)$ with $(u, t) \in (0, 1] \times [0, 1]$

```

if State(0) = +1 and State(1) = +1 then
  label [0, 1] as CSI
else if State(0) = -1 and State(1) = -1 then
  label [0, 1] as COI
else
   $t^* \leftarrow \text{ContactTime}(0, 1)$ 
  report the contact time  $t^*$ 
  if State(0) = -1 then
    label  $[0, t^*)$  as COI and  $(t^*, 1]$  as CSI
  else
    label  $[0, t^*)$  as CSI and  $(t^*, 1]$  as COI

```

Remark 5.3. For the sake of robustness, if $\text{State}(0) = 0$, we replace $\text{State}(0)$ by $\text{State}(\epsilon)$, where $\epsilon > 0$ is a sufficiently small constant. Similarly, if $\text{State}(1) = 0$, we replace $\text{State}(1)$ by $\text{State}(1 - \epsilon)$. Thus we assume that $\text{State}(0)$ and $\text{State}(1)$ can never be 0.

Processing candidate separation intervals

For a CSI (t_1, t_2) , we use the following operation, called *BézierShoot*, to either confirm that (t_1, t_2) is an SI or, if it is not, extract a separation interval which is a sub-interval of (t_1, t_2) .

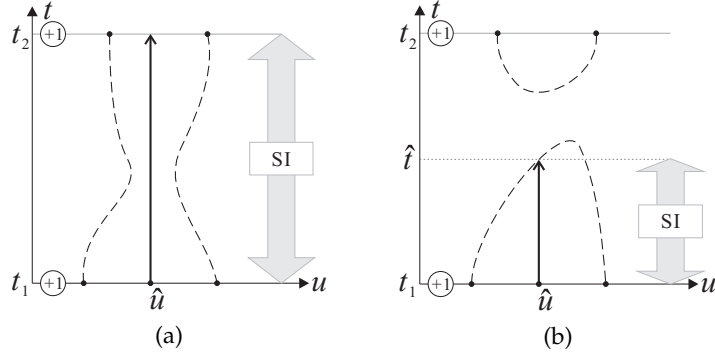


Figure 20: A Bézier shoot operation. (a) $F(\hat{u}, t) = 0$ has no real root in $[t_1, t_2]$; (b) \hat{t} is the smallest root of $F(\hat{u}, t) = 0$ in $[t_1, t_2]$.

- BézierShoot:** A Bézier shoot from t_1 to t_2 , denoted as $\text{BézierShoot}(t_1 \rightarrow t_2) = \hat{t}$, is to find a separation interval $(t_1, \hat{t}) \subseteq (t_1, t_2)$. It has two steps. In the first step, we find \hat{u} such that $F(\hat{u}, t_1) = \max_u F(u, t_1)$ (As discussed in Remark 5.3, to ensure robustness, t_1 is replaced by $t_1 + \epsilon$ if t_1 is a contact instant.) If $F(\hat{u}, t_1) \leq 0$, we conclude that no SI can be thus extracted (and set $\hat{t} = t_1$). Otherwise, we use in the second step the Bézier clipping search [45] from t_1 to t_2 to find the first root of $F(\hat{u}, t) = 0$ (an equation in t with \hat{u} being fixed), if there is one. This step either concludes that there is no real root of $F(\hat{u}, t) = 0$ in (t_1, t_2) (see Fig. 20(a)), which implies that (t_1, t_2) is an SI (and hence $\hat{t} = t_2$), or produces the smallest root \hat{t} of $F(\hat{u}, t) = 0$ in (t_1, t_2) (see Fig. 20(b)), which gives an SI $(t_1, \hat{t}) \subset (t_1, t_2)$, since a Bézier shoot ensures that $F(\hat{u}, t) > 0$ for all $t \in (t_1, \hat{t})$. A Bézier shoot from t_2 to t_1 , i.e., $\text{BézierShoot}(t_1 \leftarrow t_2)$, is defined similarly.

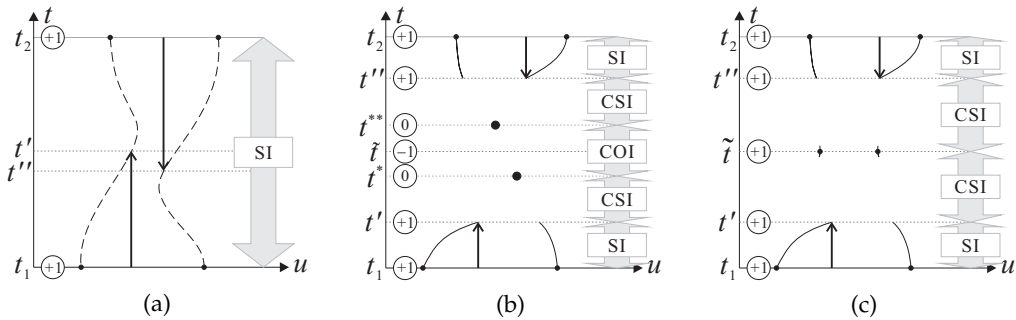


Figure 21: The handling of a candidate separation interval (CSI) in the algorithm for solving a CCD.

Given a CSI (t_1, t_2) , we perform two Bézier shoots from both ends of the interval to extract a separation interval from each end. This results in two possible cases: (1) the entire interval can be confirmed as an SI (Fig. 21(a)); or (2) two SIs (t_1, t') and (t'', t_2) are obtained, and depending on the collision status of the ellipsoids at $\tilde{t} = (t' + t'')/2$, the subdivided intervals from $[t', t'']$ are either labeled as CSIs or COIs (Fig. 21(b) & (c)) for further processing as in the following algorithm:

Algorithm 2 CSI handling

Input: A candidate separation interval (CSI) (t_1, t_2)

if interval width $t_2 - t_1$ is sufficiently small **then**

 report (t_1, t_2) as an OI (see Remark 5.4 below)

else

$t' \leftarrow \text{BézierShoot}(t_1 \rightarrow t_2)$

$t'' \leftarrow \text{BézierShoot}(t_1 \leftarrow t_2)$

if $t' > t''$ **then**

 report (t_1, t_2) as an SI

▷▷▷ Fig. 21(a)

else

 report (t_1, t') and (t'', t_2) as SIs

$\tilde{t} \leftarrow (t' + t'')/2$

if $\text{State}(\tilde{t}) = -1$ **then**

$t^* \leftarrow \text{ContactTime}(t', \tilde{t})$

▷▷▷ Fig. 21(b)

$t^{**} \leftarrow \text{ContactTime}(\tilde{t}, t'')$

 report contact time instants t^* and t^{**}

 label $[t', t^*)$, $(t^{**}, t'']$ as CSIs and

(t^*, t^{**}) as a COI

else

if $\text{State}(\tilde{t}) = 0$ **then**

 report contact time instant \tilde{t}

 label $[t', \tilde{t})$, $(\tilde{t}, t'']$ as CSIs

▷▷▷ Fig. 21(c)

Remark 5.4. One may wonder whether we should report (t_1, t_2) as an SI instead of an OI when the difference $t_2 - t_1$ is sufficiently small. But in this case we cannot

avoid some chance of having tiny loop(s) in the zero set of $F(u, t)$. Thus to be more conservative, we classify the interval as an OI.

Processing candidate overlapping intervals

For a candidate overlapping interval (COI) (t_1, t_2) , we aim to identify some overlapping intervals (OIs) within a COI, so that the remaining subintervals can be further processed. Given a COI (t_1, t_2) , if it contains any separation interval, then $F(u^*, t^*) > 0$ for some $t^* \in (t_1, t_2)$ and the zero set of $F(u, t) = 0$ contains some close loops in the strip $(u, t) \in (0, 1] \times (t_1, t_2)$. Hence, a COI can be confirmed as an OI if it does not contain any loop, and this can be checked as follows.

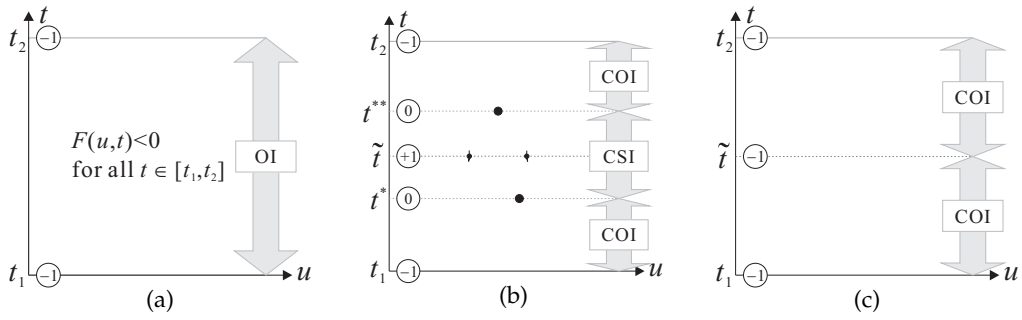


Figure 22: The handling of a candidate overlapping interval (COI) in the algorithm for solving a CCD.

We first consider the coefficients of the Bernstein form of $F(u, t)$. Using the convex hull property of the Bernstein form [15], if all the coefficients are negative, the interval (t_1, t_2) is an OI since we must have $F(u, t) < 0$ in this interval (Fig. 22(a)). If the coefficients have different signs, we will check the existence of a loop in the zero set of $F(u, t) = 0$. The existence of a loop in (t_1, t_2) implies that the derivative $F_t(u, t)$ cannot be of the same sign for all $(u, t) \in (0, 1] \times (t_1, t_2)$. For this, again using the convex hull property, we check whether the control coefficients of $F_t(u, t)$, expressed as a bivariate Bernstein function on $[0, 1] \times (t_1, t_2)$, have the same sign. To make the test more effective, we further limit this check only to the subregion in which $F(u, t)$ can possibly be positive for $t \in (t_1, t_2)$; this subregion is the maximum extent of intersection of the convex hull of the control polyhedron of $F(u, t)$ and the ut -plane. If all these coefficients of the Bernstein form of $F_t(u, t)$ are of the same sign, then the zero set

of $F(u, t)$ does not have a loop in the interval (t_1, t_2) , implying that (t_1, t_2) is an OI; otherwise, if these coefficients have different signs, (t_1, t_2) is still a COI.

If a COI remains so after the above filtering using the sign checking on the Bernstein coefficients of $F(u, t)$ and $F_t(u, t)$, we further process this interval by checking the collision status of the two ellipsoids at $\tilde{t} = (t_1 + t_2)/2$. If the two ellipsoids are separate at \tilde{t} , the two mixed interval (t_1, \tilde{t}) and (\tilde{t}, t_2) will be further processed (Fig. 22(b)). If the two ellipsoids are overlapping at \tilde{t} , we label the two subintervals (t_1, \tilde{t}) and (\tilde{t}, t_2) as COIs (Fig. 22(c)), and process them using the above coefficient filtering operation recursively. Algorithm 3 gives the detailed steps in handling a COI.

Algorithm 3 COI handling

Input: A candidate overlapping interval (COI) (t_1, t_2)

if interval width $t_2 - t_1$ is sufficiently small **then**

 report (t_1, t_2) as an OI

else

if $F(u, t) = 0$ has no loop in (t_1, t_2) **then**

 report (t_1, t_2) as an OI

▷▷▷ Fig. 22(a)

else

$\tilde{t} \leftarrow (t_1 + t_2)/2$

if $\text{State}(\tilde{t}) = +1$ **then**

$t^* \leftarrow \text{ContactTime}(t_1, \tilde{t})$

▷▷▷ Fig. 22(b)

$t^{**} \leftarrow \text{ContactTime}(\tilde{t}, t_2)$

 report contact time instants t^* and t^{**}

 label (t_1, t^*) , (t^{**}, t_2) as COIs and

(t^*, t^{**}) as a CSI

else

if $\text{State}(\tilde{t}) = 0$ **then**

 report contact time instant \tilde{t}

 label (t_1, \tilde{t}) , (\tilde{t}, t_2) as COIs

▷▷▷ Fig. 22(c)

5.3.2 Finding the first contact time only

Many real-time applications of collision detection require only the first contact time to be computed. Suppose that the two ellipsoids are separate at $t = 0$, i.e., $\text{State}(0) = +1$. We then apply Bézier shoots recursively from $t = 0$, until we encounter the first contact time. We shall show that this process has quadratic convergence (Section 5.3.3) and is efficient especially when the motion degree is low.

5.3.3 Quadratic convergence of recursive Bézier shoot

We now show that recursive Bézier shoot in search of a contact time, i.e., a regular solution (u^*, t^*) of $F(u, t) = F_u(u, t) = 0$, has quadratic convergence. Without loss of generality, we may assume that (u^*, t^*) is located at the origin $(0, 0)$, with $F(u, t) = 0$ and $F_u(u, t) = 0$ as shown in Fig. 23. Then, by the regularity assumption and Implicit Function Theorem, the solution of $F(u, t) = 0$ can be represented locally at $(0, 0)$ by Taylor expansion $t = \alpha u^2 + o(u^2)$, and the solution of $F_u(u, t) = 0$ by $u = kt + o(t)$. Now consider a Bézier shoot from t_0 . The solution of $F_u(u, t_0) = 0$ is $\hat{u} = kt_0 + o(t_0)$. So the first root of $F(\hat{u}, t) = 0$ is

$$t_1 = \alpha \hat{u}^2 + o(\hat{u}^2) = \alpha [kt_0 + o(t_0)]^2 + o(t_0^2) = \alpha k^2 t_0^2 + o(t_0^2).$$

It follows that $t_1/t_0^2 = \alpha k^2 + o(1)$. Hence, recursive Bézier shoot has quadratic convergence. But if (u^*, t^*) is a singular solution representing tangential contact of the two ellipsoids, then the convergence is in general linear.

5.4 EXPERIMENTAL RESULTS

We have tested our method in two applications to demonstrate its robustness and effectiveness. The first one features a human character animation in which two virtual human characters bounded by ellipsoids move in a sequence of frames. We determine the first contact instant of the characters in between every two consecutive frames. The

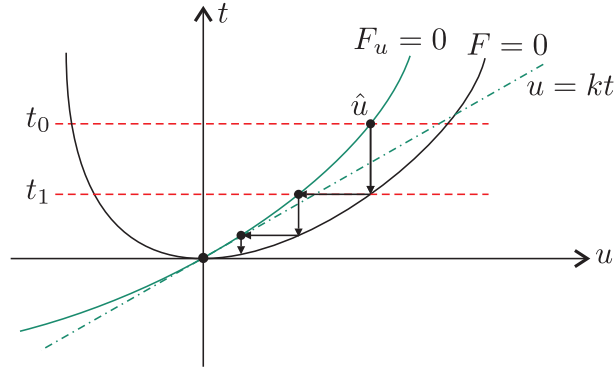


Figure 23: Quadratic convergence of recursive Bézier shoot.

motion of each ellipsoid is obtained by interpolating its orientations and positions at two consecutive frames. Both rigid and affine motion interpolation are tested and the performance in both cases are evaluated. In the second experiment we perform collision detection between a robotic arm moving with pre-specified rigid motion and a stationary obstacle. Continuous collision detection is applied among the bounding ellipsoids of the links of the robotic arm and the obstacle, and all collision time intervals are reported.

5.4.1 Test in human character animation

To test the efficiency of our method, we use two virtual boxers performing action in close proximity of each other, as shown in Fig. 24. The first contact instant in each time interval $[t_i, t_{i+1}]$ is to be determined, where the t_i are the time instants of each animation frame. Each character is bounded tightly by 20 ellipsoids, enclosing different body parts such as heads, limbs, etc. The motions of the two boxers are driven by motion capture data, together with a simple control mechanism. Between every two consecutive frames, the collision detection algorithm is applied to 400 pairs of ellipsoids, formed by picking one ellipsoid from each of the characters. We do not consider self-collision here, which can easily be dealt with by taking into account the pairwise CCD of non-adjacent ellipsoids of the same character.

Two fast and simple culling techniques are first used to quickly eliminate unlikely colliding pairs of ellipsoids. For each pair of moving ellipsoids, we first test whether

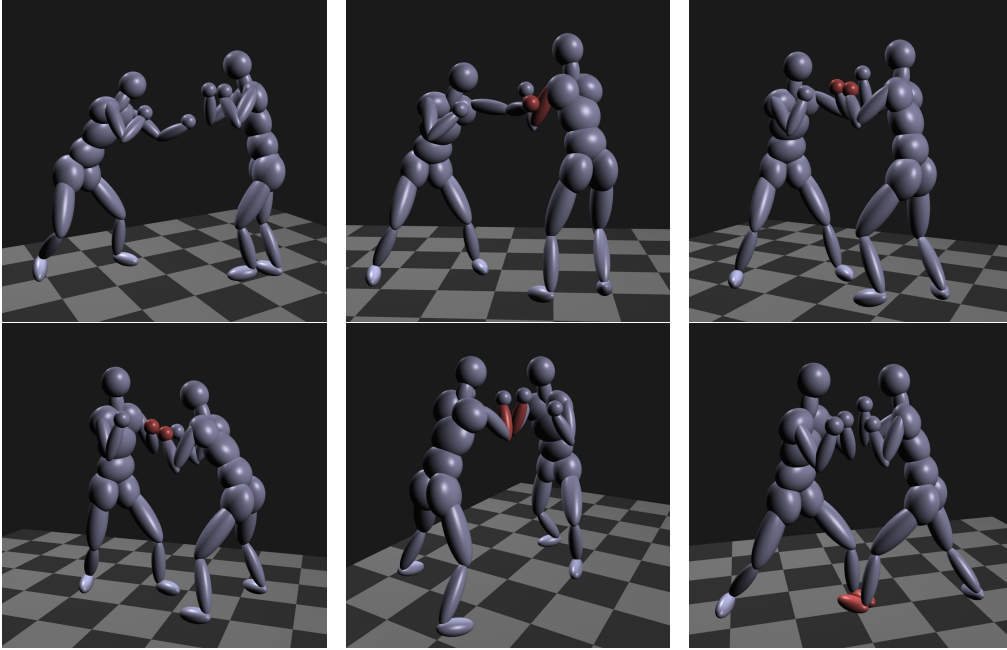


Figure 24: Real-time continuous collision detection in a boxing game. Ellipsoids in collision are highlighted in red.

their bounding spheres collide. The bounding spheres assume linear translation between the end positions of the ellipsoids. The moving spheres are guaranteed to bound the ellipsoids with the interpolating motions described in the next paragraph. To test whether the bounding spheres collide, we formulate a simple squared distance function $d(t) = |\mathbf{c}_1(t) - \mathbf{c}_2(t)|^2 - (r_1 + r_2)^2 < 0, t \in [0, 1]$, of two spheres, where $\mathbf{c}_1(t), \mathbf{c}_2(t)$ are the sphere centres and r_1, r_2 are the sphere radii. Then, two moving spheres are collision-free in $[0, 1]$ if and only if $d(0) > 0$ and $d(t)$ has no real roots in $[0, 1]$. The bounding spheres test is very efficient—it takes only $1.5 \mu\text{sec}$ per test and can filter out a large number of trivially non-collision cases, i.e., when the ellipsoids are far apart.

If the bounding spheres collide, we further apply a separating-plane method to further eliminate the remaining easy cases of non-colliding ellipsoids. We compute a plane that separates the two ellipsoids [73] at the beginning of the time frame, and then test whether the two moving ellipsoids are continuously separated by the plane during the whole frame period. We assume that the separating plane is under the same motion as one of the ellipsoid, say $\mathcal{B}(t)$, so that it is always separate from $\mathcal{B}(t)$ in

$[0, 1]$. The collision test is now between $\mathcal{A}(t)$ and the moving plane $\mathcal{P}(t)$, which are then transformed so that $\mathcal{A}(t)$ becomes the unit sphere at the origin and $\mathcal{P}(t)$ becomes $\mathcal{P}'(t)$; $\mathcal{A}(t)$ and $\mathcal{P}(t)$ collide if and only if the distance from the origin to $\mathcal{P}'(t)$ is less than 1, which can also be determined algebraically as in the bounding sphere test. The separating plane test involves also a static collision detection of the ellipsoids at $t = 0$, and hence can identify trivial collision cases where the ellipsoids overlap at $t = 0$.

A total of 1,000 frames are processed for the boxing sequence. A continuous rigid motion is used for interpolation between every two consecutive frames; the center positions of the ellipsoids are linearly interpolated and the orientations are interpolated by a linear quaternion curve, producing a rotation matrix of rational degree 2. As a result, 400,000 pairs of moving ellipsoids were tested, out of which 93.8% of the pairs were filtered out by the sphere test, and 34.1% and 62.6% of the remaining pairs were determined as colliding or collision-free at $t = 0$ using separating planes as witnesses. For the remaining 780 pairs (0.195%), we applied the algorithm from Section 5.3.2 that computes the first contact point in continuous collision detection. Of these, 742 were collision-free and 38 were in collision. Since only the first contact time is needed, we also maintain an upper bound, \bar{t} , on the contact time, which is the minimum of all the first contact time that have been computed so far. Subsequent CCD is only determined within the interval $[0, \bar{t}]$. Including all the above procedures and the generation of interpolating motions $M_A(t)$ and $M_B(t)$ for 40 ellipsoids, the average time for collision detection for each frame took 1.33 msec, in which 400 pairs of moving ellipsoids were handled. A total of 40 motion matrices were generated in 195 μ sec. The formulation of the bivariate function $F(u, t)$ takes considerable computation. However, this is needed only when the ellipsoids are in close proximity, when both the sphere test and separating plane test fail to declare separation. The first row of Table 5 summarizes the average and the worst case running time for all pairwise CCD tests. The performance for the close proximity cases is also presented.

Using a rigid motion of ration degree 2 as motion interpolant, the degree of the CCD equation $F(u, t)$ is 28 in t . In Section 5.2 we describe an affine motion interpolation which approximates the relative motion between two moving ellipsoids, and results in a CCD equation of degree 6 on t . In order to compare properly the performance of our CCD method with the two different motions, the separating plane test which

Table 5: Average CPU time taken for CCD of two virtual human characters in a boxing animation.

	τ_{frm} (ms)	τ_{mot} (ms)	τ_e (μs)	τ_f (μs)	τ_c (μs)	τ_w (μs)
Rigid motion, setup 1*	1.33	0.20				
over all 400K pairs			2.8	2.5	17.8	73.9
over 780 close pairs**			55.2	54.7	65.4	73.9
Rigid motion, setup 2*	1.90	0.20				
over all 400K pairs			4.2	4.0	13.8	100.1
Affine motion, setup 2*	1.19	0.13				
over all 400K pairs			2.6	2.5	6.6	39.4

τ_{frm} represents the average time per frame; τ_{mot} for constructing the interpolating motion; τ_e , τ_f , τ_c represent the average time for pairwise CCD of all ellipsoids, collision-free ellipsoids, and colliding ellipsoids, respectively; and τ_w is the worst case running time for all 400K pairwise CCDs.

*Setup 1—with sphere and separating plane tests, CCD over $[0, \bar{t}]$

Setup 2—with sphere test only, CCD over $[0, 1]$

**where both sphere and separating plane tests fail to declare separation

depends on the interpolating motion is not used and all CCD computations are carried out in the time interval $[0, 1]$, i.e., the upper bound \bar{t} of the first contact time is not maintained, since \bar{t} varies with different motions. The performance of our CCD method with the two motion interpolations is shown in the second and third rows of Table 5. The average time per frame has a significant 37.6% speed-up using the proposed affine motion interpolation, due to the more efficient motion construction and a CCD computation of a much lower degree. In our experiment, both motion interpolations gave the same collision result of whether a pair of ellipsoids collide or not. Not accounting those pairs with first contact at $t = 0$, the differences between the first contact time of the ellipsoids with affine motion interpolation and that with rigid motion interpolation have an average, standard deviation and maximum of 0.008, 0.03 and 0.49, respectively. We notice that the differences in the order of the maximum value occurs only in extreme cases; neglecting the maximum value gives an average, standard deviation and maximum of 0.006, 0.01 and 0.14, respectively. When using affine motion interpolation to achieve low degree polynomial computation and therefore a more efficient collision detection, significant deviation from the rigid motion may occur due to the affine approximation that varies the sizes of the ellipsoids.

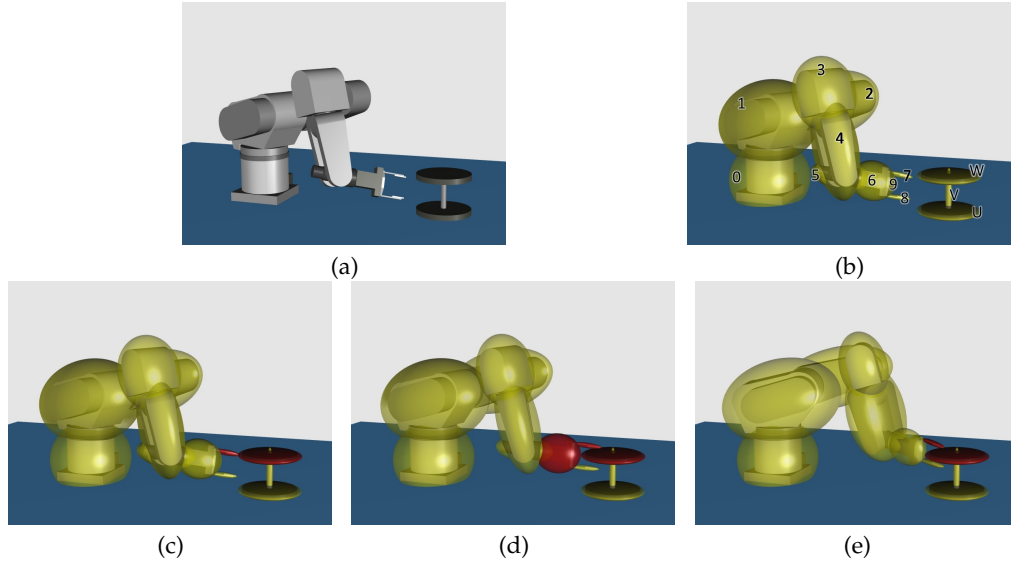


Figure 25: (a) An F3 robotic arm and an I-shaped obstacle; (b) the bounding ellipsoids; (c), (d) & (e) the robotic arm in motion with $t = 0.104, 0.311$ and 0.778 , respectively, and the colliding ellipsoids are shown in red.

5.4.2 Test in robotic arm movements

In our second experiment a CRS F3 robotic arm collides with an I-shaped obstacle. The robotic arm assumes a pre-defined rigid motion and is tightly bounded by 10 ellipsoids (0-9) and the obstacle by 3 ellipsoids (U,V,W) (Fig. 25). We perform 30 pairwise collision tests using our algorithm to find all the collision time intervals between the robotic arm and the obstacle. The motion of the robotic arm is designed in such a way that the three joints of the arm rotate with degree-2 rational motions and hence the fingers move with degree-6 rational motions. The total time for processing all 30 pairs of ellipsoids is 43.8 msec. Note that the time needed for collision detection in general depends on the motion degree as well as the complexity of the zero set of the CCD equation. The degree of $F(u, t)$ in t , the time taken for obtaining $F(u, t)$ and that for solving the CCD for each pair of ellipsoids are summarized in Table 6.

Table 6: Average CPU time taken for CCD of a robotic arm and an obstacle.

	Degree		Time (msec)			Collision intervals in $[0, 1]$
	Motion	$F(u, t)$ in t	Obtain $F(u, t)$	Solve CCD	Total	
U-0	0	0	0.045	0.077	0.122	-
U-1	2	32	0.272	0.125	0.397	-
U-2	2	32	0.272	0.130	0.402	-
U-3	2	32	0.272	0.126	0.398	-
U-4	4	64	0.813	0.838	1.651	-
U-5	4	64	0.812	0.256	1.068	-
U-6	6	96	1.612	1.642	3.253	-
U-7	6	96	1.676	0.338	2.014	-
U-8	6	96	1.680	0.356	2.036	-
U-9	6	96	1.684	1.498	3.181	-
V-0	0	0	0.045	0.076	0.122	-
V-1	2	32	0.272	0.126	0.397	-
V-2	2	32	0.272	0.131	0.402	-
V-3	2	32	0.272	0.125	0.397	-
V-4	4	64	0.813	0.860	1.673	-
V-5	4	64	0.812	0.881	1.693	-
V-6	6	96	1.612	1.591	3.203	-
V-7	6	96	1.676	0.052	1.729	-
V-8	6	96	1.680	1.505	3.186	-
V-9	6	96	1.683	1.545	3.228	-
W-0	0	0	0.045	0.077	0.122	-
W-1	2	32	0.272	0.127	0.399	-
W-2	2	32	0.272	0.130	0.402	-
W-3	2	32	0.272	0.126	0.398	-
W-4	4	64	0.812	0.859	1.671	-
W-5	4	64	0.813	0.827	1.641	-
W-6	6	96	1.612	0.075	1.687	$[0.311, 0.677]$
W-7	6	96	1.677	0.085	1.762	$[0.104, 0.323],$ $[0.778, 0.943]$
W-8	6	96	1.680	1.506	3.186	-
W-9	6	96	1.684	0.261	1.945	$[0.451, 0.538]$

5.4.3 Two further examples

We present two more examples to test the accuracy of our method and its efficiency in the case of general affine motions.

Example 5.1. Consider the two moving ellipsoids $\mathcal{A}(t) : \frac{x^2}{4} + \frac{y^2}{16} + \frac{z^2}{4} = 1$ and $\mathcal{B}(t) : x^2 + \frac{y^2}{9} + \frac{z^2}{16} = 1$ under rigid motions with the following degree-2 rotations ($R_A(t), R_B(t)$) and degree-3 translations ($T_A(t), T_B(t)$):

$$R_A(t) = \frac{1}{E_A(t)} \begin{pmatrix} -(8t^2 - 8t + 1) & -2(2t - 1) & 2(2t - 1) \\ 2(2t - 1) & 1 & 2(2t - 1)^2 \\ -2(2t - 1) & 2(2t - 1)^2 & 1 \end{pmatrix}$$

$$R_B(t) = \frac{1}{E_B(t)} \begin{pmatrix} \sqrt{2}(t - 1)(3t - 1) & 2t(2t - 1) & \sqrt{2}(t - 1)^2 \\ \sqrt{2}(2t - 1) & -2t(t - 1) & \sqrt{2}(2t - 1)^2 \\ -\sqrt{2}t(3t - 2) & 2(2t - 1)(t - 1) & \sqrt{2}t^2 \end{pmatrix}$$

where $E_A(t) = 8t^2 - 8t + 3$, $E_B(t) = -2(3t^2 - 3t + 1)$ and $T_A(t) = (-8t^3 + 24t^2 - 6t - 2, -24t^3 + 24t^2 + 6t - 6, -32t^3 + 48t^2 - 12t - 2)^T$, $T_B(t) = ((72 - 24\sqrt{2})t^3 + (-156 + 72\sqrt{2})t^2 + (114 - 72\sqrt{2})t - 27 + 24\sqrt{2}, 12t - 6, (88 - 24\sqrt{2})t^3 + (-168 + 72\sqrt{2})t^2 + (114 - 72\sqrt{2})t - 26 + 24\sqrt{2})^T$. These motions are designed so that the ellipsoids have their first contact at $t_0 = 1/2$. The degree of $F(u, t)$ in t is 34 and our algorithm reports contact at $t = 0.5$ with an error in the order of 10^{-8} . The whole computation took 0.7 msec and extracted two overlapping intervals.

Example 5.2. In Fig. 26, two ellipsoids are in motions of degree 4 with rather large affine deformations. Here, the degree of $F(u, t)$ in t is 48 and it took 2.7 msec to compute all the four overlapping intervals using the algorithm presented in Section 5.3.1. Detection of the first contact time takes 0.6 msec.

5.5 DISCUSSION

According to the operation counts, our approach requires about 20%~30% more arithmetic operations than the OBB overlap test [21] and even more operations than

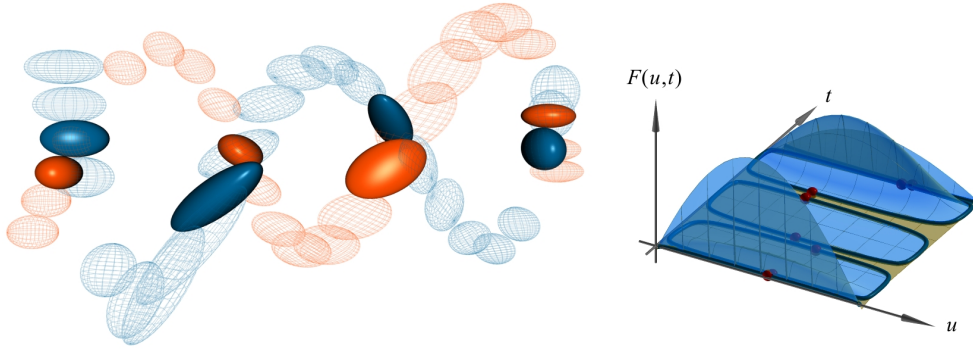


Figure 26: Two moving ellipsoids under degree-4 dependent motion with affine deformations, the CCD equation $F(u, t) = 0$ is of degree 48 in t and four overlapping intervals are detected.

other tests such as spheres, AABBs, k-DOPs, and LSSs. Thus the ellipsoidal CCD should be applied to special cases where ellipsoids provide tighter fit to freeform objects, possibly undergoing deformations that can locally be approximated by affine deformations. To this end, the recent trend in 3D modeling for the next generation GPU architecture [6, 34] is quite promising, where 3D shapes are directly represented using parametric surfaces to alleviate the bottleneck of bus bandwidth. As indicated by the Dupin indicatrix of a surface, convex parts of surfaces can be tightly fit with ellipsoids. Exact contact time and contact point of two ellipsoids would provide good initial solutions for further processing of the underlying parametric surfaces.

The other method based on the subresultant sequence of $f(\lambda)$ that we introduced in Section 4.5 for determining separation of two static ellipsoids can also be extended to deal with moving ellipsoids. In this case, the subresultant sequence is a set of polynomials in terms of the time parameter t . For a rigid motion of rational degree 2, the degree of the last polynomial in the sequence (4.2), i.e., the resultant of $f(\lambda)$ and $f'(\lambda)$, can be as high as 108. Hence, the approach described in this chapter is more robust in handling motions of higher degrees. The subresultant method, however, can be used to solve collision detection of moving composite quadric models (Chapter 6), in which internal tangencies of quadric surfaces will also be considered.

COMPOSITE QUADRIC MODELS

In this chapter, we develop algebraic methods for efficient and accurate continuous collision detection (CCD) of *composite quadric models*, or CQMs for short. A CQM may be produced with any surface modeling procedure; a commonly used one is the CSG (*Constructive Solid Geometry*) construction. A CSG construction is a binary tree in which every internal node is a Boolean expression $w = u \circ v$, where w is a CSG object resulting from a Boolean combination of its children u and v . Here, u, v are either subtrees representing intermediate CSG objects or leaf nodes representing CSG primitives. The three basic CSG operators are *union* (\cup), *intersection* (\cap), and *difference* (\setminus). All other boolean operations can be broken down into a composition of these basic operations. Figure 27 shows the collision between an ellipsoid and a CSG object which is the difference between an ellipsoid and a cylinder.

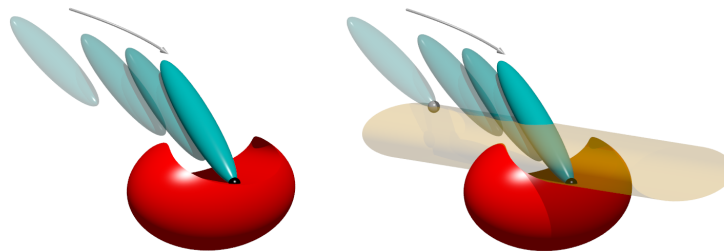


Figure 27: Collision detection of an ellipsoid and a CSG object.

A CQM is bounded by *boundary elements*; a *boundary element* of a CQM is a *face* (i.e., a planar or quadric surface patch), an *edge* (i.e., a curve segment) or a *vertex* (i.e., a point); here an edge is the intersection curve of two adjacent boundary surface patches. We

call the complete quadric surface or algebraic curve containing a boundary element an *extended boundary element*. To simplify our discussion, quadrics are assumed to include planes, and conics to include lines as a special case. The extended boundary element of a vertex is the vertex itself.

Two moving CQMs $Q_1(t)$ and $Q_2(t)$, where t is a time parameter in the interval $[t_0, t_1]$, are said to be *collision-free*, if the intersection of $Q_1(t)$ and $Q_2(t)$ is empty for all $t \in [t_0, t_1]$; otherwise, they are said to *collide*. Our goal is to determine whether the two CQMs are collision-free or not; if they collide, their first contact time instant in $[t_0, t_1]$ and the contact point will be computed.

Consider two moving CQMs that are separate initially. For collision detection, we need to determine the time when the two CQMs first have an external contact. Two CQMs may contact each other at a pair of boundary elements of the following basic types: (F, F) , (F, E) , (E, E) and (F, V) , where F, E and V stand for face, edge and vertex, respectively. Other contact types, such as (E, V) , are encompassed in the above four basic types. Therefore, we need to perform CCD of pairs of boundary elements of the four types in order to accomplish CCD of the two input CQMs. Clearly, the contact configurations for CQMs are more complex than, for example, the simple case of two ellipsoids, where a contact point is always of the type (F, F) .

Whereas the boundary curves of a CQM are in general degree four intersection curves of two quadrics, we shall focus on CCD of the important class of CQMs whose boundary curves are conics sections or straight lines, because such objects are widely used in practice and their special properties allow them to be processed for collision detection more efficiently than general CQMs. Hence, by assumption, we suppose for the moment that all edges, i.e., the boundary curves of a CQM, are conic segments, including straight line segments as a special case.

6.1 KEY IDEAS

Because CQMs are semi-algebraic varieties (i.e., defined by multiple *inequalities*) and their boundary elements are often just finite pieces on a quadric surface or a conic section, it is difficult to process them using algebraic methods in a direct manner. We

circumvent this difficulty using an approach consisting of the following two major steps: 1) *contact computation*; and 2) *contact verification*.

The first step uses algebraic methods to compute, for each pair of boundary elements, the contact point between their *extended boundary elements*, i.e., the complete containing quadrics or conics containing the boundary elements. If there is a contact detected between these extended boundary elements, then the contact point may or may not lie on the corresponding boundary elements of the input CQMs. See, for example, the two tangency points between the moving ellipsoid and the cylinder in Figure 27. Therefore we need in the second step to verify whether a contact point reported in the first step is indeed a contact point between the original CQMs.

Step 1 – Contact Computation: We use an example of two moving capped elliptic cylinders as shown in Figure 28 to illustrate the main idea. A capped elliptic cylinder Q is a CQM comprised of a cylinder \mathcal{C} and two elliptic disks at the two ends. The extended boundary elements of a capped cylinders include a cylinder, two planes and two ellipses. Given two moving capped elliptic cylinders $Q_A(t)$ and $Q_B(t)$, we may treat $Q_A(t)$ as if it is static, thus denoted as Q_A , and that $Q_B(t)$ moves relative to Q_A . For CCD of the two capped cylinders, taking all possible combinations of extended boundary elements, we need to consider 17 CCD subproblems, which are classified into three cases: (1) one CCD subproblem of a cylinder versus a cylinder (type (F, F) , Figure 28a); (2) twelve CCD subproblems of a cylinder versus an ellipse or an ellipse versus a plane (type (F, E) , Figure 28b & c); (3) four CCD subproblems of an ellipse versus an ellipse in 3D (type (E, E) , Figure 28d). There are some combinations of boundary elements for which we do not consider as subproblems, e.g., plane vs. plane, for reasons that will be discussed later in Section 6.2.1.

To explain the step of *contact computation*, we discuss briefly how some of the above cases can be handled. For case (1), suppose that the equations of the moving cylinders $\mathcal{C}_A(t)$ and $\mathcal{C}_B(t)$ are given by $X^T A(t) X = 0$ and $X^T B(t) X = 0$, respectively. The matrices $A(t)$ and $B(t)$ are singular and are of rank 3. The characteristic polynomial $f(\lambda; t) = \det(\lambda A - B(t))$ is therefore quadratic in λ . Similar to our work on ellipsoids, we can first establish the connection between the contact of the cylinders and the roots of $f(\lambda; t) = 0$ and then use it to devise a CCD algorithm for two cylinders. It can be

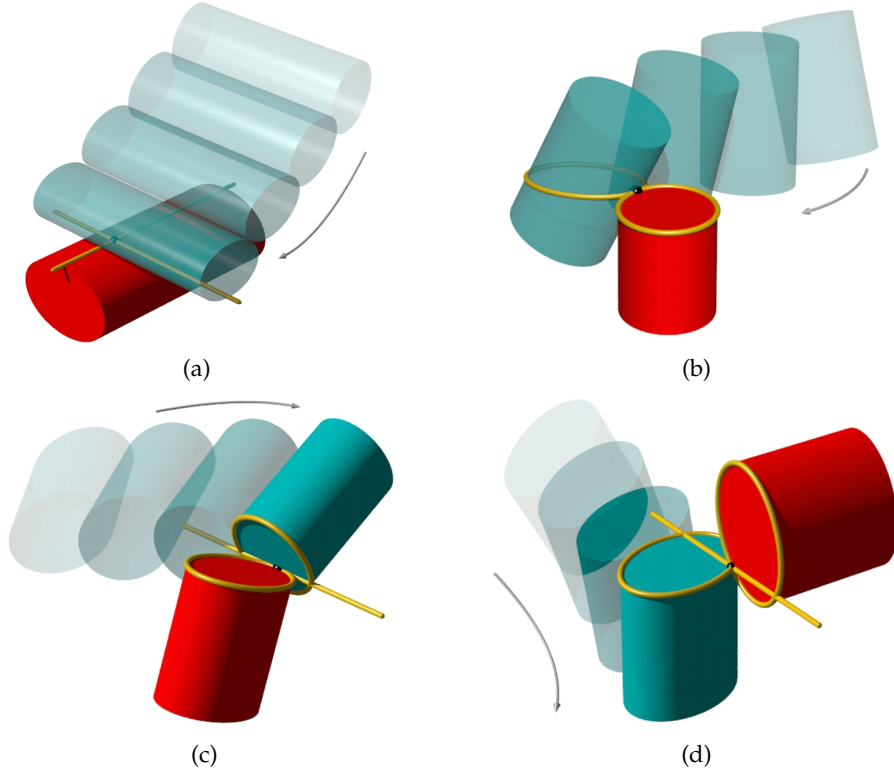


Figure 28: CCD subproblems for two capped cylinders.

shown that at the contact time t_0 the equation $f(\lambda; t_0) = 0$ has a multiple root λ_0 , and the contact point V_0 is given by a nontrivial solution of the equation $(\lambda_0 A - B(t_0))V = 0$, where by assumption the coefficient matrix $\lambda_0 A - B(t_0)$ is singular.

In case (2), let us only consider CCD of a cylinder and an ellipse. Without loss of generality, assume that only the cylinder $\mathcal{C}(t)$ moves and the ellipse \mathcal{E} is static. Let Π be the plane containing \mathcal{E} . Let $\mathcal{E}_\Pi(t)$ be the cross-section of $\mathcal{C}(t)$ in the plane Π . Then, the CCD problem is reduced to one between two ellipses \mathcal{E} and $\mathcal{E}_\Pi(t)$ in the 2D plane Π , which again can be solved with the algebraic approach.

Now, for case (3), if the two ellipses are contained in the same plane, then it is a CCD problem for two ellipses in 2D, and it is done. Otherwise, the containing planes of the two ellipses intersect in a line in 3D (projective space). Each ellipse cuts the line in an empty interval or one closed interval. Therefore, the CCD subproblem is reduced to one between two moving intervals in a line (i.e., a 1D space), which can be treated algebraically, by regarding the intervals as “low-dimensional” ellipses.

The basic idea depicted above can be extended and applied to more general CQMs with conic boundary curves. Essentially, we use a *dimension reduction technique* to reduce the CCD of extended boundary elements to CCD of quadrics or conics in different dimensions so that they can be resolved algebraically. Hence, we are able to find the contact points and contact times between *extended boundary elements* of two moving CQMs. We will discuss this step in more details in the next section.

Step 2 – Contact Verification: We will use the CSG construction tree to explain the idea of contact verification. Other boundary surface representation may entail different procedures for contact points verification, but the idea is essentially the same. Suppose that \mathbf{p} has been reported as a contact point between two extended boundary elements of two CQMs $\mathcal{Q}_A(t_0)$ and $\mathcal{Q}_B(t_0)$ at time t_0 . It suffices to consider the three basic Boolean operations, since all other CSG operations can be described as their compositions. Let $\overset{\circ}{u} = \text{“}\mathbf{p} \text{ is in the interior of } u\text{”}$ and $\partial u = \text{“}\mathbf{p} \text{ is on the boundary of } u\text{”}$ be two Boolean predicates. For each internal node associated with a CSG object w , we will evaluate $\overset{\circ}{w}$ and ∂w recursively using the following rules (see Figure 29 for the 2D analogy):

$$\begin{array}{ll}
\text{Case } w = u \cup v : & \overset{\circ}{w} \leftrightarrow \overset{\circ}{u} \vee \overset{\circ}{v}, \\
& \partial w \leftrightarrow (\partial u \wedge \partial v) \vee (\partial u \wedge \neg \overset{\circ}{v}) \vee (\neg \overset{\circ}{u} \wedge \partial v) \\
\text{Case } w = u \cap v : & \overset{\circ}{w} \leftrightarrow \overset{\circ}{u} \wedge \overset{\circ}{v}, \\
& \partial w \leftrightarrow (\partial u \wedge \partial v) \vee (\overset{\circ}{u} \wedge \partial v) \vee (\partial u \wedge \overset{\circ}{v}) \\
\text{Case } w = u \setminus v : & \overset{\circ}{w} \leftrightarrow \overset{\circ}{u} \wedge \neg(\overset{\circ}{v} \vee \partial v), \\
& \partial w \leftrightarrow (\partial u \wedge \partial v) \vee (\partial u \wedge \neg \overset{\circ}{v}) \vee (\overset{\circ}{u} \wedge \partial v)
\end{array}$$

Whether or not a point is inside or on the boundary of a quadric CSG primitive can be checked using the equation of the quadric. The answer to whether \mathbf{p} is on the boundary surface of $\mathcal{Q}_A(t_0)$ is then given by the truth value of the predicate $\partial \mathcal{Q}_A(t_0)$ evaluated at the root node of the CSG tree of $\mathcal{Q}_A(t_0)$. Similarly, we can check if \mathbf{p} is on $\partial \mathcal{Q}_B(t_0)$.

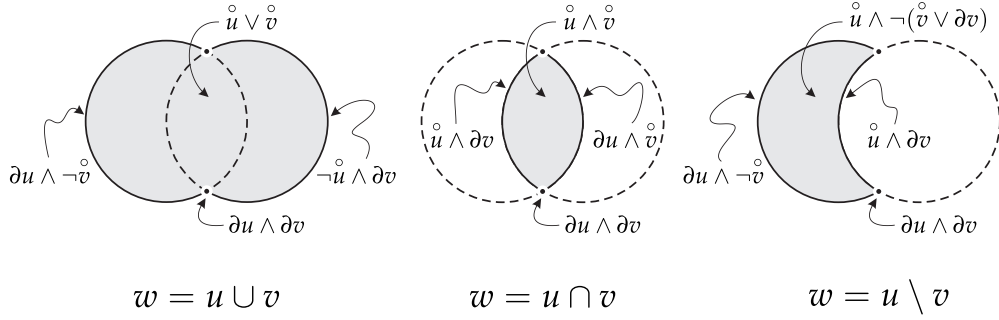


Figure 29: Contact verification for CSG objects. Given two objects u and v , and the result w of applying a boolean operation to u and v , the boundary and interior of w , denoted by ∂w and $\overset{\circ}{w}$, respectively, can be determined from ∂u , $\overset{\circ}{u}$, ∂v and $\overset{\circ}{v}$ accordingly.

6.2 THE ALGORITHM

Given two moving CQMs $\mathcal{A}(t)$ and $\mathcal{B}(t)$, $t \in [t_0, t_1]$, we shall solve the CCD of the CQMs as follows:

1. Identify the subproblems, each of which is a CCD between two extended boundary element, one from each of $\mathcal{A}(t)$ and $\mathcal{B}(t)$.
2. For each subproblem,
 - a) Depending on the subproblem type, solve CCD of the extended boundary element pair using appropriate algebraic techniques, and thereby obtain the candidate contact time instants (*contact computation*).
 - b) Find the first *valid* contact time instant with a corresponding contact point for the pair (*contact verification*), if there are any.
3. Based on the CCD results of all the subproblems, determine the first contact time instant and the corresponding contact point for $\mathcal{A}(t)$ and $\mathcal{B}(t)$.

Contact verification has been described in Section 6.1. The details for the steps for identifying subproblems and CCD by dimension reduction will be discussed in the following subsections.

6.2.1 Subproblems identification

In general, we will need to solve the CCD problem of each pair of boundary elements, each from one of the two CQMs \mathcal{A} and \mathcal{B} , if the pair should fall into one of these types: (F, F) , (F, E) , (E, E) and (F, V) , where F, E and V stand for face, edge and vertex, respectively. We notice that while a CQM is formed by boolean operations of primitive objects (which are convex for quadric primitives,) the difference operator may result in boundary elements that are *concave* (for example, the surface element resulting from an ellipsoid minus a cylinder in Fig. 27.) We have the following definition for a concave boundary element of a CQM:

Definition 6.1. A face of a CQM is said to be concave if it possesses a negative principal curvature everywhere on the face. An edge of a CQM is said to be concave if the edge e lies on a planar face f and any line connecting any two points of e is not contained in f .

Example 6.1. Figure 30 shows two CQMs in which (a) is a ring constructed by subtracting a cylinder from an ellipsoid; and (b) is a wedge formed by subtracting a half space and a circular cylinder from an elliptic cylinder. The faces $F_{A,2}$ and $F_{B,2}$ are concave faces, while the edges $E_{B,3}$ and $E_{B,4}$ are concave edges.

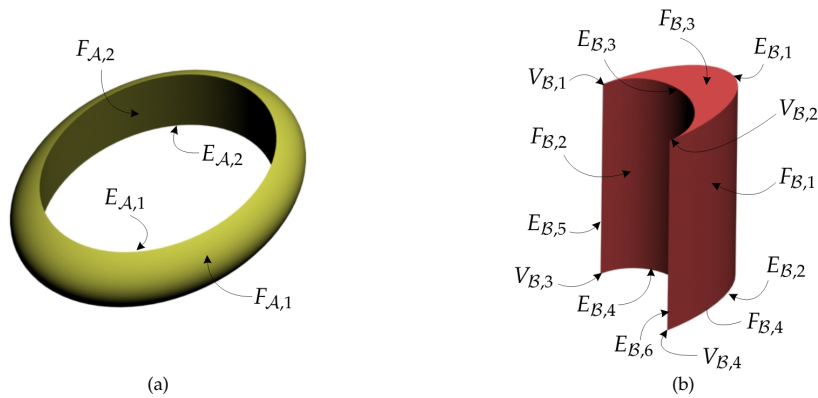


Figure 30: Two CQM objects.

Now with the concave face and concave edge of a CQM defined, we may eliminate some pairwise CCD which certainly will not capture any valid contact point of the

CQMs in question. The rules are applied as follows:

Subproblem exclusion rule

For F-F pair: Pairwise tests between concave/planar face and concave/planar face are excluded. Obviously, concave face and concave/planar face cannot contain any valid touching point. On the other hand, a plane of a CQM must be delimited by a boundary curve of type E. Hence, a plane/plane touching could be found by CCD of a plane and the boundary curve of the other plane.

For F-E pair: Pairwise tests between concave face and concave edge are excluded. Again, this boundary element pair should contain no valid touching point.

Example 6.2. The CCD subproblems that need to be considered for the two CQM objects in Fig. 30 are as follows:

- F-F — $\{F_{A,i}\} \times \{F_{B,j}\}, i = 1, 2, j = 1, \dots, 4$, excluding $(F_{A,2}, F_{B,2}), (F_{A,2}, F_{B,3})$ and $(F_{A,2}, F_{B,4})$.
- F-E — $\{F_{A,i}\} \times \{E_{B,j}\}, i = 1, 2, j = 1, \dots, 6$, excluding $(F_{A,2}, E_{B,3}), (F_{A,2}, E_{B,4})$; and $\{E_{A,i}\} \times \{F_{B,j}\}, i = 1, 2, j = 1, \dots, 4$.
- F-V — $\{F_{A,i}\} \times \{V_{B,j}\}, i = 1, 2, j = 1, \dots, 4$.
- E-E — $\{E_{A,i}\} \times \{E_{B,j}\}, i = 1, 2, j = 1, \dots, 6$.

6.2.2 Contact computation

For each pair of extended boundary elements $\mathcal{A}_i(t)$ and $\mathcal{B}_j(t)$ of the CQMs $\mathcal{A}(t)$ and $\mathcal{B}(t)$, respectively, our next step is to solve CCD and compute the first contact time instant and the corresponding contact point. As mentioned in Section 6.1, we adopted a *dimension reduction technique* to deal with CCD of boundary elements of different types. First of all, we introduce the *direct substitution method* to deal with CCD of two extended boundary elements, when one element is a quadric or a conic, and the other is a line or vertex. Then, we will proceed to describe how the subproblems of different types could be solved.

Direct substitution method

The direct substitution method checks if a line touches a quadric or a conic, or a vertex is on a quadric or a conic (both elements being in \mathbb{R}^3 or \mathbb{R}^2), by substituting the line (expressed in parametric form) and the vertex into the quadric/conic equation.

Suppose $\mathcal{Q}(t) : X^T Q(t) X = 0$ is a quadric surface or a conic curve and $\mathcal{L}(s; t)$ is a line in \mathbb{R}^3 . We simply substitute $\mathcal{L}(s; t)$ into $\mathcal{Q}(t)$ and obtain $g(s; t) = L(s; t)^T Q(t) L(s; t)$ which is a quadratic polynomial in s . The contact time instants of $\mathcal{Q}(t)$ and $\mathcal{L}(s; t)$ are then given by the roots of the discriminant $\Delta_g(t)$ of $g(s; t)$.

For CCD of $\mathcal{Q}(t)$ with a vertex, we obtain the equation $\mathbf{v}^T(t) Q(t) \mathbf{v}(t) = 0$ whose roots correspond to the contact time instants. This is also applicable when $\mathcal{Q}(t)$ is a plane.

When the two elements are a plane and a line, the direct substitution method is not appropriate since it will determine an intersection of a plane and a line, while for the purpose of CCD, we seek a touching configuration where a line should lie on the plane. In this case, a different handling will be needed.

Face vs. face (F-F)

- *Quadric vs. Quadric*

Since all quadric surfaces are projectively equivalent and that the multiplicity of roots of the characteristic polynomial is projectively invariant, Theorem 4.1 regarding the correspondence between a double root and tangency between two ellipsoids can easily be extended to apply to other quadric surfaces as well. However, unlike ellipsoids, the interior of a general quadric surface $\mathcal{Q} : X^T Q X = 0$ may not be well-defined. Here, we adopt the convention that the “interior” of \mathcal{Q} is defined with respect to the CQM object to which \mathcal{Q} belongs, so that a point X_0 in the interior of the CQM should give $X_0^T Q X_0 < 0$. Then, we seek the external tangency of $\mathcal{A}_i(t)$ and $\mathcal{B}_j(t)$ with respect to the CQM which always results in a negative double root of the characteristic equation $f(\lambda; t) = 0$. This applies also to the case of CCD between two conics, by extending Theorem 2.6 regarding the external tangency of two ellipses with a projective consideration.

As we have explained earlier, there is always a double root in the characteristic equation $f(\lambda)$ of two spheres corresponding to an imaginary contact point. Now, consider the case when we are dealing with the CCD of two spheres, both being the extended boundary elements of some CQMs. Then, the discriminant $\Delta_f(t)$ will be a zero function throughout $t \in [0, 1]$, and we cannot extract the possible genuine touching of the two spheres merely from $\Delta_f(t) = 0$. In fact, this happens not only for two spheres, but also for any two quadrics whose intersection is reducible. Hence, we make use of the subresultant sequence to detect possible touching moments of two quadrics.

Given two quadric surfaces $\mathcal{A}_i(t) : X^T A_i(t) X = 0$ and $\mathcal{B}_j(t) : X^T B_j(t) X = 0$ in \mathbb{R}^3 , we set up the characteristic polynomial $f(\lambda; t) = \det(\lambda A_i(t) - B_j(t))$. We then compute the subresultant sequence $\mathcal{P}(t)$ similarly as in (4.2) described in Section 4.5, which is given by

$$\mathcal{P}(t) = f(\lambda; t), f'(\lambda; t), \text{sr}_2(\lambda; t), \text{sr}_1(\lambda; t), \text{res}(f(\lambda; t), f'(\lambda; t)) \quad (6.1)$$

For degenerate quadrics such as cones and cylinders, the degree of $f(\lambda; t)$ is smaller than 4, and the subresultant sequence $\mathcal{P}(t)$ consists of fewer polynomials.

We first look for tangency (both internal and external) of $\mathcal{A}_i(t)$ and $\mathcal{B}_j(t)$, by determining the zeros of the first non-zero function in $\mathcal{P}(t)$ in reverse order. For example, if $\text{res}(f(\lambda; t), f'(\lambda; t)) \not\equiv 0$, its zeros become the candidate contact moments \underline{t} 's. Otherwise, we compute the zeros of $\text{sr}_1(\lambda; t) \not\equiv 0$ as the required \underline{t} 's. Next, those \underline{t} 's such that $f(\lambda; \underline{t}) = 0$ have no negative multiple roots correspond to internal touching and are therefore rejected. For each of the remaining \underline{t} 's, we determine a candidate contact point \mathbf{p} which is the solution of $(\lambda A_i(\underline{t}) - B_j(\underline{t}))X = 0$. The point \mathbf{p} is where $\mathcal{A}_i(\underline{t})$ and $\mathcal{B}_j(\underline{t})$ share a common tangent plane, and it is not necessarily a ‘‘genuine’’ touching point for the purpose of collision detection. Hence, we should also check the contact configuration of $\mathcal{A}_i(\underline{t})$ and $\mathcal{B}_j(\underline{t})$. Furthermore, two quadrics may have internal tangency at a single point, two distinct points, or along a conic curve. We shall further discuss these issues in relation to CCD of general quadrics in section 6.3.

- *Quadric vs. Plane*

A plane $\mathcal{P}(t) : P(t)^T X = 0$ will be treated as a degenerate quadrics $\mathcal{Q}(t) : X^T Q(t) X = 0$ where $Q(t) = P(t)P(t)^T$, and the CCD in this case can be solved in the same way as that between two quadrics described above.

Face vs. edge (F-E)

- *Quadric vs. conic*

Given a quadric surface $\mathcal{Q}(t) : X^T Q(t) X = 0$ and a conic curve $\mathcal{C}(t) : X^T C(t) X = 0$ in \mathbb{R}^3 with a containing plane $\Pi_{\mathcal{C}}(t)$, we reduce the CCD problem of $\mathcal{Q}(t)$ and $\mathcal{C}(t)$ to a CCD problem in \mathbb{R}^2 . This is done by considering the conic curve $\mathcal{U}(t)$ which is the intersection of $\mathcal{Q}(t)$ with $\Pi_{\mathcal{C}}(t)$. The CCD is then carried out between $\mathcal{C}(t)$ and $\mathcal{U}(t)$ in \mathbb{R}^2 . The characteristic polynomial and discriminant function are set up, and the roots of $\Delta(t) = 0$ are computed. Each root t_0 will then be checked to see if $\mathcal{U}(t_0)$ is imaginary, or that t_0 corresponds to an internal contact; in both cases t_0 will be rendered invalid.

- *Plane vs. conic*

Given a plane $\mathcal{P}(t) : P(t)^T X = 0$ and a conic curve $\mathcal{C}(t) : X^T C(t) X = 0$ in \mathbb{R}^3 with a containing plane $\Pi_{\mathcal{C}}(t)$, we consider the line $\mathcal{L}(s;t)$ (in parametric form) which is the intersection of $\mathcal{P}(t)$ and $\Pi_{\mathcal{C}}(t)$. Then we solve the CCD of $\mathcal{L}(s;t)$ and $\mathcal{C}(t)$ in \mathbb{R}^2 using direct substitution. Again, a root t_i which gives an imaginary line \mathcal{L} (or equivalently $\mathcal{P} \parallel \Pi_{\mathcal{C}}(t)$) will be rejected.

- *Quadric vs. line*

This is solved using the direct substitution method.

- *Plane vs. line*

Given a plane $\mathcal{P}(t) : P(t)^T X = 0$ and a line $\mathcal{L}(s;t)$ in \mathbb{R}^3 , we determine, for the purpose of CCD of two CQMs, the time instants at which $\mathcal{L}(s;t)$ lies on $\mathcal{P}(t)$. We substitute $\mathcal{L}(s;t)$ into $\mathcal{P}(t)$ and obtain $g(s;t) = L(s;t)^T P(t)$ which is a linear polynomial in s . Let $g(s;t) = \alpha_1(t)s + \alpha_0(t)$. The line $\mathcal{L}(s;t)$ lies on $\mathcal{P}(t)$ if and only if $g(s;t) \equiv 0$. Hence, the contact time instants are the common roots of $\alpha_0(t)$ and $\alpha_1(t)$, which can be computed by solving both $\alpha_0(t)$ and $\alpha_1(t)$ directly.

We may also set up the Sylvester resultant [64], $\text{Res}_t(\alpha_0, \alpha_1)$, of $\alpha_0(t)$ and $\alpha_1(t)$ and solve for the roots of $\text{Res}_t(\alpha_0, \alpha_1)$. This latter method, however, requires the treatment of the Sylvester resultant which is a polynomial of much higher degree in t .

Face vs. vertex (F-V)

- *Quadric surface / plane vs. vertex*

This is done by direct substitution.

Edge vs. edge (E-E)

- *Conic curve vs. conic curve*

In this case, we will reduce the CCD problem of two conic curves in \mathbb{R}^3 to a 1D CCD problem in \mathbb{R} as in the case for solving CCD of ellipses moving in the space as described in Chapter 3.

- *Conic curve vs. line*

This is solved by direct substitution.

- *Line vs. line*

Two lines $\mathcal{L}_1(s; t) = \mathbf{u}_1(t) + s\mathbf{v}_1(t)$ and $\mathcal{L}_2(w; t) = \mathbf{u}_2(t) + w\mathbf{v}_2(t)$ intersect if and only if $\mathbf{v}_1(t)$, $\mathbf{v}_2(t)$ and $\mathbf{u}_2(t) - \mathbf{u}_1(t)$ are coplanar. Hence, we solve for the roots of $\det[\mathbf{v}_1(t), \mathbf{v}_2(t), \mathbf{u}_2(t) - \mathbf{u}_1(t)] = 0$, which corresponds to the contact time instants of the two lines.

6.3 ISSUES FOR CCD OF GENERAL QUADRICS

6.3.1 Contact configuration

Given two moving quadric surfaces $\mathcal{A}(t)$ and $\mathcal{B}(t)$, let $g(t)$ be the first non-zero function in the subresultant sequence $\mathcal{P}(t)$ of their characteristic polynomial $f(\lambda; t)$ and its derivative as given by (6.1). A root t_0 of $g(t)$ does not always correspond to a

real touching point. Hence, we will need to further check if $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ indeed have real tangency.

We will first solve for the multiple root λ_0 of the characteristic equation $f(\lambda) = \det(\lambda A(t_0) - B(t_0)) = 0$. If $\lambda_0 \geq 0$, t_0 will be rejected since it corresponds to an internal contact of the two surfaces with respect to the CQMs. Otherwise, we shall solve for the system $(\lambda_0 A(t_0) - B(t_0))X = 0$, whose solution X_0 is the contact point of $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$.

When two quadric surfaces touch each other (no matter whether it is an internal touch or an external touch,) the tangency could happen either at a single point, two distinct points, a straight line or a conic section in \mathbb{C}^3 . We shall show that these cases can be distinguished by inspecting the rank of $(\lambda_0 A(t_0) - B(t_0))$.

Lemma 6.2. *Let λ_0 be a multiple root of $f(\lambda) = \det(\lambda A(t_0) - B(t_0)) = 0$, the characteristic polynomial of two quadric surfaces \mathcal{A} and \mathcal{B} . We have the following cases¹:*

1. *If $\text{rank}(\lambda_0 I - A^{-1}B) = 3$, \mathcal{A} and \mathcal{B} have a single real tangency point in \mathbb{R}^3 .*
2. *If $\text{rank}(\lambda_0 I - A^{-1}B) = 2$, \mathcal{A} and \mathcal{B} either have two distinct tangency point, or that they have tangency along a straight line in \mathbb{C}^3 .*
3. *If $\text{rank}(\lambda_0 I - A^{-1}B) = 1$, \mathcal{A} and \mathcal{B} are tangent along a conic curve in \mathbb{C}^3 .*

Proof. Case 1: If $\text{rank}(\lambda_0 I - A^{-1}B) = 3$, λ_0 is associated with only 1 Jordan block of size 3×3 in the Jordan canonical form of $A^{-1}B$. Then there are a real eigenvector X_0 and a generalized eigenvector X_1 of $A^{-1}B$ such that

$$(\lambda_0 I - A^{-1}B)X_0 = 0 \quad \text{and} \quad (\lambda_0 I - A^{-1}B)X_1 = X_0,$$

and hence $(\lambda_0 I - A^{-1}B)^2 X_1 = 0$. Now,

$$\begin{aligned} X_0^T A X_0 &= [(\lambda_0 I - A^{-1}B)X_1]^T A [(\lambda_0 I - A^{-1}B)X_1] \\ &= X_1^T (\lambda_0 I - A^{-1}B) A (\lambda_0 I - A^{-1}B) X_1 \\ &= X_1^T A (\lambda_0 I - A^{-1}B)^2 X_1 \\ &= 0, \end{aligned}$$

¹ Note that $\text{rank}(\lambda_0 I - A^{-1}B) = \text{rank}(\lambda_0 A - B)$

i.e., X_0 is a point on \mathcal{A} . Similarly, we may show that X_0 is a point on \mathcal{B} . Since $(\lambda_0 I - A^{-1}B)X_0 = 0$, we have $\lambda_0 AX_0 = BX_0$, and therefore the tangent planes, $X^T AX_0 = 0$ and $X^T BX_0 = 0$ of \mathcal{A} and \mathcal{B} at X_0 , are identical. Hence, \mathcal{A} and \mathcal{B} have a real single tangency point X_0 which is the unique solution of $(\lambda I - A^{-1}B)X = 0$.

Case 2: Suppose that $\text{rank}(\lambda_0 I - A^{-1}B) = 2$, then the null space of $A^{-1}B$ has dimension $2(= 4 - 2)$ and λ_0 is associated with two linearly independent eigenvectors X_0 and X_1 , such that

$$(\lambda_0 I - A^{-1}B)X_0 = 0 \quad \text{and} \quad (\lambda_0 I - A^{-1}B)X_1 = 0.$$

Consider the line $X(u; v) = uX_0 + vX_1$, $u, v \in \mathbb{R}$. A line in general intersects a quadric in two points (counting multiplicity) in \mathbb{C}^3 , or that the line lies on the quadrics. We first assume that $X(u; v)$ intersects \mathcal{A} at two points. Let $X(\underline{u}; \underline{v})$ for some $\underline{u}, \underline{v} \in \mathbb{R}$ be such an intersection point on \mathcal{A} . Since $(\lambda_0 A - B)X_0 = 0$ and $(\lambda_0 A - B)X_1 = 0$, we have

$$\begin{aligned} 0 &= \underline{u}(\lambda_0 A - B)X_0 + \underline{v}(\lambda_0 A - B)X_1 \\ &= (\lambda_0 A - B)X(\underline{u}; \underline{v}) \\ &= X(\underline{u}; \underline{v})^T (\lambda_0 A - B)X(\underline{u}; \underline{v}) \\ &= \lambda_0 X(\underline{u}; \underline{v})^T AX(\underline{u}; \underline{v}) - X(\underline{u}; \underline{v})^T BX(\underline{u}; \underline{v}) \\ &= -X(\underline{u}; \underline{v})^T BX(\underline{u}; \underline{v}) \end{aligned}$$

The last equality holds since $X(\underline{u}; \underline{v})$ is on \mathcal{A} . Hence, we have shown that $X(\underline{u}; \underline{v})$ is also on \mathcal{B} . Moreover, since $\lambda_0 AX_0 = BX_0$ and $\lambda_0 AX_1 = BX_1$, we have

$$\begin{aligned} \lambda_0 AX(\underline{u}; \underline{v}) &= \lambda_0 A(\underline{u}X_0 + \underline{v}X_1) \\ &= \underline{u}\lambda_0 AX_0 + \underline{v}\lambda_0 AX_1 \\ &= \underline{u}BX_0 + \underline{v}BX_1 \\ &= BX(\underline{u}; \underline{v}). \end{aligned}$$

Hence, the tangent planes, $X^T A X(\underline{u}; \underline{v}) = 0$ and $X^T B X(\underline{u}; \underline{v}) = 0$ of \mathcal{A} and \mathcal{B} at $X(\underline{u}; \underline{v})$, are identical, which means that $X(\underline{u}; \underline{v})$ is a tangency point of \mathcal{A} and \mathcal{B} . On the other hand, if $X(u; v)$ lies on the quadric surfaces, we may show similarly that every point on $X(u; v)$ is a tangency point of \mathcal{A} and \mathcal{B} .

Case 3: If $\text{rank}(\lambda_0 I - A^{-1}B) = 1$, then the null space of $A^{-1}B$ has dimension 3 and λ_0 is associated with three linearly independent eigenvectors X_0, X_1, X_2 such that

$$(\lambda_0 I - A^{-1}B)X_0 = 0, \quad (\lambda_0 I - A^{-1}B)X_1 = 0, \quad \text{and} \quad (\lambda_0 I - A^{-1}B)X_2 = 0.$$

A plane in general intersects a quadrics in a conic section in \mathbb{C}^3 . Now let $X(u; v; w) = uX_0 + vX_1 + sX_2$, $u, v, s \in \mathbb{R}$ be the plane spanned by the eigenvectors X_0, X_1, X_2 . Also, let $X(\underline{u}; \underline{v}; \underline{w})$ for some $\underline{u}, \underline{v}, \underline{w} \in \mathbb{R}$ be a point on the intersection conic curve of $X(u; v; w)$ and \mathcal{A} . Then, since $(\lambda_0 A - B)X_0 = (\lambda_0 A - B)X_1 = (\lambda_0 A - B)X_2 = 0$, we have

$$\begin{aligned} 0 &= \underline{u}(\lambda_0 A - B)X_0 + \underline{v}(\lambda_0 A - B)X_1 + \underline{w}(\lambda_0 A - B)X_2 \\ &= (\lambda_0 A - B)X(\underline{u}; \underline{v}; \underline{w}) \\ &= X(\underline{u}; \underline{v}; \underline{w})^T (\lambda_0 A - B)X(\underline{u}; \underline{v}; \underline{w}) \\ &= \lambda_0 X(\underline{u}; \underline{v}; \underline{w})^T A X(\underline{u}; \underline{v}; \underline{w}) - X(\underline{u}; \underline{v}; \underline{w})^T B X(\underline{u}; \underline{v}; \underline{w}) \\ &= -X(\underline{u}; \underline{v}; \underline{w})^T B X(\underline{u}; \underline{v}; \underline{w}). \end{aligned}$$

The last equality holds as $X(\underline{u}; \underline{v}; \underline{w})$ is on \mathcal{A} . Hence, $X(\underline{u}; \underline{v}; \underline{w})$ is also on \mathcal{B} . It means that \mathcal{A} and \mathcal{B} share a common intersection curve with the plane $X(u; v; w)$. Now, we also have

$$\begin{aligned} \lambda_0 A X(\underline{u}; \underline{v}; \underline{w}) &= \lambda_0 A (\underline{u}X_0 + \underline{v}X_1 + \underline{w}X_2) \\ &= \underline{u}\lambda_0 A X_0 + \underline{v}\lambda_0 A X_1 + \underline{w}\lambda_0 A X_2 \\ &= \underline{u}B X_0 + \underline{v}B X_1 + \underline{w}B X_2 \\ &= B X(\underline{u}; \underline{v}; \underline{w}). \end{aligned}$$

Then, the tangent planes of \mathcal{A} and \mathcal{B} , $X^T \lambda_0 A X(\underline{u}; \underline{v}; \underline{w})$ and $X^T \lambda_0 B X(\underline{u}; \underline{v}; \underline{w})$ are identical at every point $X(\underline{u}; \underline{v}; \underline{w})$ on the intersection curve. Hence, the two quadrics \mathcal{A} and \mathcal{B} are tangent at a conic curve. \square

Note that Lemma 6.2 gives the touching configuration of \mathcal{A} and \mathcal{B} in \mathbb{C}^3 , if their characteristic equation has a multiple root. For CCD of \mathcal{A} and \mathcal{B} , one should further check if the contact is in the real space \mathbb{R}^3 .

6.3.2 Tangency occurring on real intersection curves

There are cases in which two quadric surfaces have tangency at some point \mathbf{p} in \mathbb{R}^3 , but however intersect locally in the neighbourhood of \mathbf{p} . This happens when the tangency point lies on some real intersection curve of the quadric surfaces. Figure 31a shows an example in the case of two cylinders. Now, imagine the case when one of the quadrics contributes to a concave surface in a CQM, then this touching point is not valid in the sense that it is not an external contact point for the two CQMs that the quadrics belong to, since the CQMs in this case intersect locally in \mathbb{R}^3 in the neighbourhood of the touching point. This compares to the situation where a touching point is not on any of the real intersection curve of the quadrics; when one of the quadrics is considered concave, the touching point will be a valid external contact (Figure 31b).

Definition 6.3. A contact point \mathbf{p} of two quadrics \mathcal{A} and \mathcal{B} is called a *genuine contact point* if and only if \mathbf{p} does not lie on any real intersection curve between \mathcal{A} and \mathcal{B} ; or in other words, \mathcal{A} and \mathcal{B} does not intersect locally in the neighbourhood of \mathbf{p} .

By merely looking into the characteristic equation which characterizes the pencil of two quadrics, we cannot differentiate whether a contact point is genuine or not, i.e, whether it lies on any real intersection curve of the two quadrics. In this regard, we make use of the signature sequence [66] to classify the morphology of the intersection curve. Table 7, 8 and 9 are taken from [66] which show the classification of 35 different morphologies for the intersection curve of two quadrics (QSIC) in $\mathbb{P}\mathbb{R}^3$, using the index sequence and the signature sequence. The illustration of each case shows how a QSIC looks like: a solid line and a dotted line represents a real and an imaginary intersection component, respectively, while a tangency point is marked with a circular point.

Table 7: Classification of QSiC that contain non-planar components in $\mathbb{P}R^3$













$[\text{Segre}]_r$ $r = \text{the \#}$ of real roots	Index Sequence	Signature Sequence	Illustration	Representative Quadric Pair	Real Contact Point
[1111] ₄	1 $\langle 1 2 1 2 3 \rangle$	$(1,(1,2),2,(1,2),1,(1,2),2,(2,1),3)$		$\mathcal{A}: x^2 + y^2 + z^2 - w^2 = 0$ $\mathcal{B}: 2x^2 + 4y^2 - w^2 = 0$	
	2 $\langle 0 1 2 3 4 \rangle$	$(0,(0,3),1,(1,2),2,(2,1),3,(3,0),4)$		$\mathcal{A}: x^2 + y^2 + z^2 - w^2 = 0$ $\mathcal{B}: 2x^2 + 4y^2 + 3z^2 - w^2 = 0$	
[1111] ₂	3 $\langle 1 2 3 \rangle$	$(1,(1,2),2,(2,1),3)$		$\mathcal{A}: 2xy + z^2 + w^2 = 0$ $\mathcal{B}: -x^2 + y^2 + z^2 + 2w^2 = 0$	
[1111] ₀	4 $\langle 2 \rangle$	(2)		$\mathcal{A}: xy + zw = 0$ $\mathcal{B}: -x^2 + y^2 - 2z^2 + zw + 2w^2 = 0$	
[211] ₃	5 $\langle 2\ell_-2 3 2 \rangle$ $\langle 2\ell_+2 3 2 \rangle$	$(2,((2,1)),2,(2,1),3,(2,1),2)$ $(2,((1,2)),2,(2,1),3,(2,1),2)$		$\mathcal{A}: x^2 - y^2 + z^2 + 4yw = 0$ $\mathcal{B}: -3x^2 + y^2 + z^2 = 0$	✓
	6 $\langle 1\ell_-1 2 3 \rangle$	$(1,((1,2)),1,(1,2),2,(2,1),3)$		$\mathcal{A}: -x^2 - z^2 + 2yw = 0$ $\mathcal{B}: -3x^2 + y^2 - z^2 = 0$	✓
	7 $\langle 1\ell_+1 2 3 \rangle$	$(1,((0,3)),1,(1,2),2,(2,1),3)$		$\mathcal{A}: x^2 + z^2 + 2yw = 0$ $\mathcal{B}: 3x^2 + y^2 + z^2 = 0$	✓
[211] ₁	8 $\langle 2\ell_-2 \rangle$	$(2,((2,1)),2)$		$\mathcal{A}: xy + zw = 0$ $\mathcal{B}: 2xy + y^2 - z^2 + w^2 = 0$	✓
[22] ₂	9 $\langle 2\ell_-2\ell_-2 \rangle$ $\langle 2\ell_-2\ell_+2 \rangle$	$(2,((2,1)),2,((2,1)),2)$ $(2,((2,1)),2,((1,2)),2)$		$\mathcal{A}: xy + zw = 0$ $\mathcal{B}: y^2 + 2zw + w^2 = 0$	✓
[22] ₀	10 $\langle 2 \rangle$	(2)		$\mathcal{A}: xw + yz = 0$ $\mathcal{B}: xz - yw = 0$	
[31] ₂	11 $\langle 1\ell\ell_+2 3 \rangle$	$(1,(((1,2))),2,(2,1),3)$		$\mathcal{A}: y^2 + 2xz + w^2 = 0$ $\mathcal{B}: 2yz + w^2 = 0$	✓
[4] ₁	12 $\langle 2\ell\ell\ell_-2 \rangle$	$(2,(((2,1))),2)$		$\mathcal{A}: xw + yz = 0$ $\mathcal{B}: z^2 + 2yw = 0$	✓

Table 8: Classification of QSIC with only planar components in $\mathbb{P}\mathbb{R}^3$ - Part I

$[\text{Segre}]_r$ $r = \text{the \#}$ of real roots	Index Sequence	Signature Sequence	Illustration	Representative Quadric Pair	Real Contact Point
[[11]11] ₃	<u>13</u> $\langle 2 2 1 2 \rangle$	$(2,((1,1)),2,(1,2),1,(1,2),2)$		$\mathcal{A}: x^2 - y^2 + z^2 - w^2 = 0$ $\mathcal{B}: x^2 - 2y^2 = 0$	✓
	<u>14</u> $\langle 1 3 2 3 \rangle$	$(1,((1,1)),3,(2,1),2,(2,1),3)$		$\mathcal{A}: -x^2 + y^2 + z^2 + w^2 = 0$ $\mathcal{B}: -x^2 + 2y^2 = 0$	
	<u>15</u> $\langle 1 1 2 3 \rangle$	$(1,((0,2)),1,(1,2),2,(2,1),3)$		$\mathcal{A}: x^2 + y^2 + z^2 - w^2 = 0$ $\mathcal{B}: x^2 + 2y^2 = 0$	✓
	<u>16</u> $\langle 0 2 3 4 \rangle$ $\langle 1 3 4 3 \rangle$	$(0,((0,2)),2,(2,1),3,(3,0),4)$ $(1,((1,1)),3,(3,0),4,(3,0),3)$		$\mathcal{A}: x^2 + y^2 - z^2 - w^2 = 0$ $\mathcal{B}: x^2 + 2y^2 = 0$	
[[11]11] ₁	<u>17</u> $\langle 1 3 \rangle$	$(1,((1,1)),3)$		$\mathcal{A}: x^2 + y^2 + 2zw = 0$ $\mathcal{B}: -z^2 + w^2 + 2zw = 0$	
	<u>18</u> $\langle 2 2 \rangle$	$(2,((1,1)),2)$		$\mathcal{A}: x^2 - y^2 - 2zw = 0$ $\mathcal{B}: -z^2 + w^2 + 2zw = 0$	✓
[[111]1] ₂	<u>19</u> $\langle 1 2 3 \rangle$	$(1,(((0,1))),2,(2,1),3)$		$\mathcal{A}: y^2 + z^2 - w^2 = 0$ $\mathcal{B}: x^2 = 0$	
	<u>20</u> $\langle 0 3 4 \rangle$	$(0,(((0,1))),3,(3,0),4)$		$\mathcal{A}: y^2 + z^2 + w^2 = 0$ $\mathcal{B}: x^2 = 0$	
[[21]1] ₂	<u>21</u> $\langle 1\lambda_- 2 3 \rangle$	$(1,(((1,1))),2,(2,1),3)$		$\mathcal{A}: y^2 - z^2 + 2zw = 0$ $\mathcal{B}: -x^2 + z^2 = 0$	✓
	<u>22</u> $\langle 1\lambda_+ 2 3 \rangle$	$(1,(((0,2))),2,(2,1),3)$		$\mathcal{A}: y^2 - z^2 + 2zw = 0$ $\mathcal{B}: x^2 + z^2 = 0$	✓

Table 9: Classification of QSIC with only planar components in $\mathbb{P}R^3$ - Part II

$[\text{Segre}]_r$ $r = \text{the \# of real roots}$	Index Sequence	Signature Sequence	Illustration	Representative Quadric Pair	Real Contact Point
[2(11)] ₂	<u>23</u> $\langle 2\hat{\imath}_- \hat{\imath}_- 2 \rangle$	$(2, ((2,1)), 2, ((1,1)), 2)$		$\mathcal{A}: 2xy - y^2 = 0$ $\mathcal{B}: y^2 + z^2 - w^2 = 0$	✓
	<u>24</u> $\langle 1\hat{\imath}_- \hat{\imath}_- 3 \rangle$	$(1, ((1,2)), 1, ((1,1)), 3)$		$\mathcal{A}: 2xy - y^2 = 0$ $\mathcal{B}: y^2 - z^2 - w^2 = 0$	✓
	<u>25</u> $\langle 1\hat{\imath}_+ \hat{\imath}_+ 3 \rangle$	$(1, ((0,3)), 1, ((1,1)), 3)$		$\mathcal{A}: 2xy - y^2 = 0$ $\mathcal{B}: y^2 + z^2 + w^2 = 0$	✓
[(31)] ₁	<u>26</u> $\langle 2\hat{\imath}_- 2 \rangle$	$(2, (((1,1))), 2)$		$\mathcal{A}: y^2 + 2xz - w^2 = 0$ $\mathcal{B}: yz = 0$	✓
	<u>27</u> $\langle 1\hat{\imath}_+ 3 \rangle$	$(1, (((1,1))), 3)$		$\mathcal{A}: y^2 + 2xz + w^2 = 0$ $\mathcal{B}: yz = 0$	✓
[(11)(11)] ₂	<u>28</u> $\langle 2\hat{\imath}_- 2 \rangle$	$(2, ((1,1)), 2, ((1,1)), 2)$		$\mathcal{A}: x^2 - y^2 = 0$ $\mathcal{B}: z^2 - w^2 = 0$	✓
	<u>29</u> $\langle 0\hat{\imath}_- 4 \rangle$	$(0, ((0,2)), 2, ((2,0)), 4)$		$\mathcal{A}: x^2 + y^2 = 0$ $\mathcal{B}: z^2 + w^2 = 0$	
	<u>30</u> $\langle 1\hat{\imath}_- 3 \rangle$	$(1, ((0,2)), 1, ((1,1)), 3)$		$\mathcal{A}: x^2 + y^2 = 0$ $\mathcal{B}: z^2 - w^2 = 0$	✓
[(11)(11)] ₀	<u>31</u> $\langle 2 \rangle$	$\langle 2 \rangle$		$\mathcal{A}: xy + zw = 0$ $\mathcal{B}: -x^2 + y^2 - z^2 + w^2 = 0$	
[(211)] ₁	<u>32</u> $\langle 2\hat{\imath}_- 2 \rangle$	$(2, (((1,0))), 2)$		$\mathcal{A}: x^2 - y^2 + 2zw = 0$ $\mathcal{B}: z^2 = 0$	✓
	<u>33</u> $\langle 1\hat{\imath}_- 3 \rangle$	$(1, (((1,0))), 3)$		$\mathcal{A}: x^2 + y^2 + 2zw = 0$ $\mathcal{B}: z^2 = 0$	✓
[(22)] ₁	<u>34</u> $\langle 2\hat{\imath}_- \hat{\imath}_- 2 \rangle$	$(2, (((2,0))), 2)$		$\mathcal{A}: xy + zw = 0$ $\mathcal{B}: y^2 + w^2 = 0$	
	<u>35</u> $\langle 2\hat{\imath}_- \hat{\imath}_+ 2 \rangle$	$(2, (((1,1))), 2)$		$\mathcal{A}: xy - zw = 0$ $\mathcal{B}: y^2 - w^2 = 0$	✓

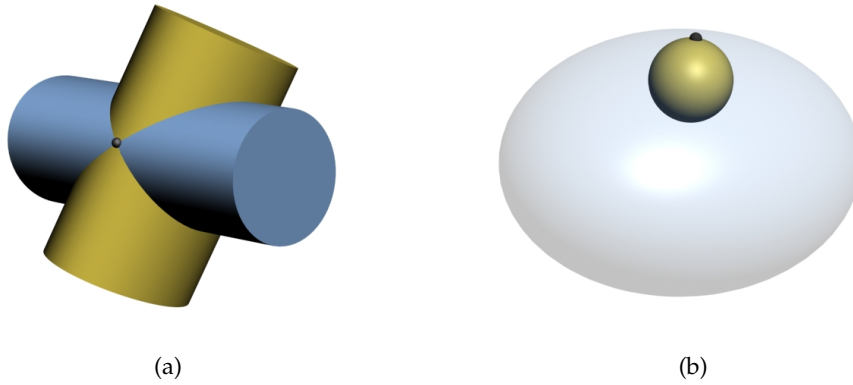


Figure 31: Tangency of quadrics with different intersection configuration in the neighbourhood. (a) Touching point lying on the real intersection curves of the two quadrics. (b) There is no intersection locally at the touching point between the quadrics. Only (b) will be considered a *genuine touching point* for CCD of two quadrics.

We are only interested in those cases where the two quadrics have real contact (i.e., the 22 cases marked by the last column in the three tables). The index sequence and the signature sequence are the root characterization of the characteristic equation $f(\lambda) = \det(\lambda A - B) = 0$. To facilitate our discussions, we shall first show how these two sequences should be interpreted ²:

1. Each multiple root λ_0 of $f(\lambda) = 0$ is associated with a sequence of consecutive symbols $|$ or \wr in the index sequence. The multiplicity of λ_0 equals the number of consecutive symbols. In the signature sequence, there is a corresponding pair of numbers enclosed by a pair of bracket. The number of pairs of brackets used equals the multiplicity of λ_0 .
2. The roots of $f(\lambda) = 0$ are given by the eigenvalues of $A^{-1}B$ and each eigenvalue is associated with some Jordan blocks in the Jordan normal form of $A^{-1}B$. Then the symbol $|$ denotes a real root associated with a 1×1 Jordan block, while \wr is used for p consecutive times to denote a real root associated with a $p \times p$ Jordan block. Hence, if λ_0 is associated with the symbols $\wr \wr \wr |$, it has two corresponding Jordan blocks, one is 3×3 and the other is 1×1 .

² Here, we try to simplify the interpretation of the two sequences and extract only those important properties that are related to our later analysis.

3. The number before the symbols for λ_0 characterizes the quadric surface $Q(\lambda_0 - \epsilon) : X^T((\lambda_0 - \epsilon)A - B)X = 0$ using the signs of the eigenvalues of $(\lambda_0 - \epsilon)A - B$, where $\epsilon > 0$ is a sufficiently small constants. An even number means $Q(\lambda_0 - \epsilon)$ is a hyperboloid, and an odd number means $Q(\lambda_0 - \epsilon)$ is an ellipsoid. Similarly, we may characterize $Q(\lambda_0 + \epsilon)$ using the number that comes after the symbols corresponding to λ_0 .
4. The two numbers (p, n) within the brackets corresponding to λ_0 in the signature sequence tell the number of positive and negative non-zero eigenvalues of $\lambda_0 I - A^{-1}B$, respectively. Hence, if λ_0 is associated with $((1, 1))$, then λ_0 is a triple root and that $\lambda_0 I - A^{-1}B$ has one non-zero positive and one non-zero negative eigenvalues.

Consider two quadric surfaces \mathcal{A} and \mathcal{B} which are in tangency, i.e., $f(\lambda) = \det(\lambda A - B) = 0$ with a multiple root λ_0 . Based on Table 7, 8 and 9 characterizing the index sequence and signature sequence of \mathcal{A} and \mathcal{B} , the following procedure determines whether \mathcal{A} and \mathcal{B} have genuine contact.

Procedure:

Input: Two quadric surfaces \mathcal{A} and \mathcal{B} which are in tangency, with $f(\lambda) = \det(\lambda A - B) = 0$ having a multiple root λ_0 .

Output: Whether \mathcal{A} and \mathcal{B} HAVE or DO NOT HAVE genuine contact.

Step 1: Check the index sequence of \mathcal{A} and \mathcal{B} .

Let $Q(\lambda)$ be $X^T(\lambda A - B)X = 0$.

- If $Q(\lambda_0 - \epsilon)$ and $Q(\lambda_0 + \epsilon)$ are both ellipsoids for sufficiently small $\epsilon > 0$, i.e., the numbers before and after the symbols corresponding to λ_0 in the index sequence are both odd,
 - If λ_0 is a quadruple root,

report: DO NOT HAVE genuine contact Case 27 (Table 10)
 - Else

report: HAVE genuine contact. Cases 6,7,15,24,25,30,33

- If $\mathcal{Q}(\lambda_0 - \epsilon)$ and $\mathcal{Q}(\lambda_0 + \epsilon)$ are both hyperboloids for sufficiently small $\epsilon > 0$, i.e., the numbers before and after the symbols corresponding to λ_0 in the index sequence are both even,

– *report*: DO NOT HAVE genuine contact Cases 5,8,9,12,13,18,23,26,28,32,35

- If one of $\mathcal{Q}(\lambda_0 - \epsilon)$ and $\mathcal{Q}(\lambda_0 + \epsilon)$ is an ellipsoid and the other a hyperboloid for sufficiently small $\epsilon > 0$, go to Step 2.

Step 2: Now that one of $\mathcal{Q}(\lambda_0 - \epsilon)$ and $\mathcal{Q}(\lambda_0 + \epsilon)$ is an ellipsoid and the other a hyperboloid, for sufficiently small $\epsilon > 0$;

if $\text{rank}(\lambda_0 I - A^{-1}B) = 3$,

report: DO NOT HAVE a genuine contact Case 11

Otherwise, check the signature sequence:

- If the non-zero eigenvalues of $\lambda_0 I - A^{-1}B$ are of different signs,
report: DO NOT HAVE genuine contact. Case 21
- If the non-zero eigenvalues of $\lambda_0 I - A^{-1}B$ are of the same sign,
report: HAVE genuine contact. Case 22

Step 2 serves to differentiate cases 11, 21 and 22 whose properties are shown in Table 10.

Table 10: Cases 11, 21, 22 and 27 of the QSIC classification that need special handling for determining whether a contact point is genuine or not.

Index Sequence	Signature Sequence	Illustration	Differentiation
<u>11</u> $\langle 1 1 _+ 2 3 \rangle$	$(1, (((1,2))), 2, (2,1), 3)$		$\text{rank}(\lambda_0 I - A^{-1}B) = 3$
<u>21</u> $\langle 1 1 _- 2 3 \rangle$	$(1, (((1,1))), 2, (2,1), 3)$		$\text{rank}(\lambda_0 I - A^{-1}B) = 2$ non-zero eigenvalues of $\lambda_0 I - A^{-1}B$ are of different signs
<u>22</u> $\langle 1 1 _+ 2 3 \rangle$	$(1, (((0,2))), 2, (2,1), 3)$		$\text{rank}(\lambda_0 I - A^{-1}B) = 2$ non-zero eigenvalues of $\lambda_0 I - A^{-1}B$ are of the same sign
<u>27</u> $\langle 1 1 1 _+ 3 \rangle$	$(1, (((1,1))), 3)$		quadruple root

The above procedure requires the testing of whether $\mathcal{Q}(\lambda_0 \pm \epsilon)$ are ellipsoids or hyperboloids, which might require determining the eigenvalues of the coefficient

matrices. It turns out that we have a simple computation scheme, based on the fact that $f(\lambda') < 0$ if $\mathcal{Q}(\lambda')$ is an ellipsoid; and $f(\lambda') > 0$ if $\mathcal{Q}(\lambda')$ is a hyperboloid. We may then use the N^{th} derivative test of $f(\lambda)$ at λ_0 , and conclude the nature of $\mathcal{Q}(\lambda_0 \pm \epsilon)$ as follows. Suppose the $f^n(\lambda_0) \neq 0$ is the first nonzero derivative at λ_0 :

- If n is even, and that $f^n(\lambda_0) < 0$, then $\mathcal{Q}(\lambda_0 \pm \epsilon)$ are both ellipsoids.
- If n is even, and that $f^n(\lambda_0) > 0$, then $\mathcal{Q}(\lambda_0 \pm \epsilon)$ are both hyperboloids.
- If n is odd, then one of $\mathcal{Q}(\lambda_0 \pm \epsilon)$ is an ellipsoid and the other a hyperboloid.

Example 6.3. The two cylinders $\mathcal{A} : x^2 + z^2 = 1$ and $\mathcal{B} : y^2 + z^2 = 1$ have contact configuration as in Figure 31a. Their characteristic equation is $f(\lambda) = \lambda(\lambda - 1)^2$ and hence has a double root 1. Now, $\text{rank}(\lambda_0 A - B) = 2$ and $(\lambda_0 A - B)X = 0$ has two linearly independent solutions $X_0 = (0, 0, 1, 0)^T$ and $X_1 = (0, 0, 0, 1)^T$. Intersecting the line $uX_0 + vX_1$, $u, v \in \mathbb{R}$, with the cylinders yields the two distinct contact points at $(0, 0, \pm 1)^T$ in \mathbb{R}^3 . Furthermore, $f'(1) = 0$ and $f''(1) = 2$. Hence, $\mathcal{Q}(1 \pm \epsilon)$ are both hyperboloids, and by Step 1 of the above procedure, \mathcal{A} and \mathcal{B} do not have genuine contact.

6.4 TWO WORKING EXAMPLES

Example 6.4. We now solve the CCD of two capped elliptic cylinders $\mathcal{A}(t)$ and $\mathcal{B}(t)$, both of the same size (Figure 32). The boundary elements of the cylinders are:

- Face $F_{\mathcal{A},1}, F_{\mathcal{B},1}$: a cylinder $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z \in [-5, 5]$.
- Face $F_{\mathcal{A},2}, F_{\mathcal{B},2}$: a plane $z = -5$.
- Face $F_{\mathcal{A},3}, F_{\mathcal{B},3}$: a plane $z = 5$.
- Edge $E_{\mathcal{A},1}, E_{\mathcal{B},1}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z = -5$.
- Edge $E_{\mathcal{A},2}, E_{\mathcal{B},2}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z = 5$.

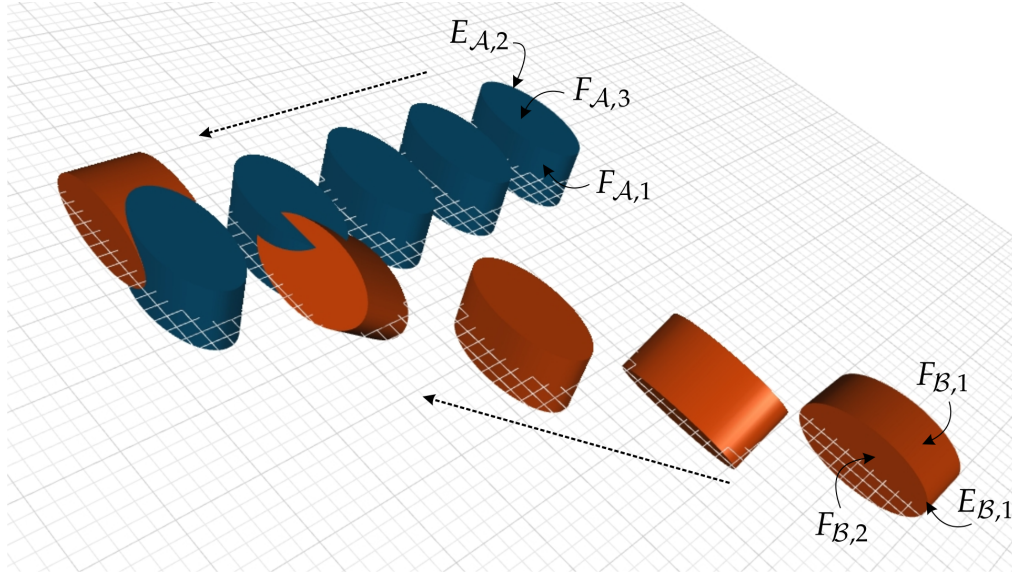


Figure 32: CCD of two capped cylinders.

The motions of \mathcal{A} and \mathcal{B} are

$$M_A(t) = \begin{pmatrix} 1 & 0 & 0 & -60t + 30 \\ 0 & 1 & 0 & 20 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{and}$$

$$M_B(t) = \begin{pmatrix} -2t^2 + 2t & 0 & -2t + 1 & -120t^3 + 180t^2 - 120t + 30 \\ 0 & 2t^2 - 2t + 1 & 0 & 160t^3 - 260t^2 + 180t - 50 \\ 2t - 1 & 0 & -2t^2 + 2t & 0 \\ 0 & 0 & 0 & 2t^2 - 2t + 1 \end{pmatrix},$$

respectively. The subproblems that we need to consider are:

- F-F — $(F_{A,1}, F_{B,1})$
- F-E — $(F_{A,1}, E_{B,1}), (F_{A,1}, E_{B,2}), (F_{B,1}, E_{A,1}), (F_{B,1}, E_{A,2}), (F_{A,2}, E_{B,1}), (F_{A,2}, E_{B,2}), (F_{A,3}, E_{B,1}), (F_{A,3}, E_{B,2}), (F_{B,2}, E_{A,1}), (F_{B,2}, E_{A,2}), (F_{B,3}, E_{A,1}), (F_{B,3}, E_{A,2})$
- E-E — $(E_{A,1}, E_{B,1}), (E_{A,1}, E_{B,2}), (E_{A,2}, E_{B,1}), (E_{A,2}, E_{B,2})$

Note that the subproblems for cylinder vs. plane, and plane vs. plane are excluded, as they are encompassed by the other subproblems. We will now illustrate how CCD of four of the above subproblems (corresponding to the four cases in Figure 28) is solved.

F-F: $(F_{A,1}, F_{B,1})$ —cylinder vs. cylinder

Here, the characteristic polynomial $f(\lambda; t) = \det(\lambda A(t) - B(t))$ is quadratic in λ and $\text{res}(f(\lambda; t), f'(\lambda; t))$ is of degree 32 in t . They are omitted because of the long expressions. The roots of $\text{res}(f(\lambda; t), f'(\lambda; t))$ is found to be $t_0 = 0.5, 0.625, 0.875$.

- For $t_0 = 0.5$, we have $f(\lambda; t_0) = 0$ and the axes of the cylinders are parallel. Hence, we transform $\mathcal{A}(t_0)$ and $\mathcal{B}(t_0)$ by $M_A^{-1}(t_0)$ so that the axes are orthogonal to the xy -plane, and check whether the cross-section ellipses touch each other on the xy -plane. Now, the characteristic polynomial $f(\lambda; t_0) = -0.0004\lambda^3 - 0.00015\lambda^2 + 0.9375 \times 10^{-5}\lambda + 0.9766 \times 10^{-7}$ of the cross-section ellipses has zeros $-0.4284, -0.00912, 0.0625$. The ellipses are separate and hence the cylinders are separate at $t = 0.5$. Therefore, $t = 0.5$ is invalid.
- For $t_0 = 0.625$, $f(\lambda; t_0) = 0.2822 \times 10^{-6}\lambda^2 + 0.4496 \times 10^{-7}\lambda + 0.1791 \times 10^{-8}$ which has a negative double root $\lambda_0 = -0.0796$. Hence, the cylinders are in external touch. The contact point is found to be $(-7.5, 10, 0)^T$ (i.e., the solution of $(\lambda A(t_0) - B(t_0))X = 0$) and is verified to be a point on both capped cylinders. Hence, a valid first contact at $t = 0.625$ is found for the cylinders and we may skip the other larger roots of $\Delta_f(t) = 0$.

F-E: $(F_{A,1}, E_{B,1})$ —cylinder vs. ellipse

Both $(F_{A,1}, E_{B,1})$ are transformed to the object space of $E_{B,1}$ on the xy -plane. The intersection of the transformed $F_{A,1}$ and the xy -plane is an ellipse \mathcal{E} whose coefficients are polynomials in t of degree as high as 9. The discriminant of the characteristic polynomial of \mathcal{E} and $E_{B,1}$ is of degree 32 in t , and the roots are found to be $t_0 = 0, 0.6341, 1$.

- For $t_0 = 0$, there is no negative double root of $f(\lambda; t_0) = 0$ and hence t_0 will be rejected. There is, however, a double root equals zero; since \mathcal{E} is a line that does not touch $E_{B,1}$.

- For $t_0 = 0.6341$, $f(\lambda; t_0) = 0$ has a negative double root $\lambda_0 = -0.2843$. Solving the system $(\lambda_0 A - B)X = 0$ yields a single contact point $(-6.623, 10.414, -4.95)^T$, which can be verified to lie on both truncated cylinders.

F-E: $(F_{A,3}, E_{B,1})$ —plane vs. ellipse

Both $F_{A,3}(= \mathcal{P}(t))$, $E_{B,1}(= \mathcal{E}(t))$ are transformed to the object space of $\mathcal{E}(t)$ on the xy -plane. The plane $\mathcal{P}(t)$, after transformation, is not parallel to nor equivalent to the xy -plane for all t ; it intersects the xy -plane in the line $\mathcal{L}(s; t) = (10t - 5, s(1 - 2t), 0, 4t^2 - 4t + 1)^T$. Substituting \mathcal{L} into the ellipse equation yields $h(s; t) = (\frac{1}{25}t^2 - \frac{1}{25}t + \frac{1}{100})s^2 - 16t^4 + 32t^3 - 20t^2 + 4t$. Solving for the discriminant $\Delta_h(t)$ gives the roots $t_0 = 0, 0.5, 1$.

- For $t_0 = 0$, $\mathcal{P}(t_0)$ is not parallel to nor equivalent to the xy -plane. Solving $h(s; t_0) = 0$ gives $s_0 = 0$, and the contact point is then given by $X_0 = (25, -50, 5)^T$. Next, we check if X_0 is in the elliptic disk on $\mathcal{P}(t)$: since $X_0^T E(t) X_0 = 49 > 0$, X_0 is not in the elliptic disk, and hence $t_0 = 0$ is rejected.
- For $t_0 = 0.5$, $\mathcal{P}(t_0)$ is parallel to the xy -plane, and t_0 is therefore rejected.
- For $t_0 = 1$, $\mathcal{P}(t_0)$ is not parallel to nor equivalent to the xy -plane. The contact point is found to be $X_0 = (-25, 30, 5)^T$; however, it is not within the elliptic disk and $t_0 = 1$ is rejected.
- Hence, $\mathcal{P}(t_0)$ and $\mathcal{E}(t_0)$ are collision-free.

E-E: $(E_{A,1}, E_{B,1})$ —ellipse vs. ellipse

We transform both $E_{A,1}(= \mathcal{E}_A(t))$ and $E_{B,1}(= \mathcal{E}_B(t))$ to the object space of $\mathcal{E}_B(t)$ on the xy -plane. The containing plane of $\mathcal{E}_A(t)$ does not equal to the xy -plane for all t and we proceed to compute the contact. Two 1D ellipses are formed and the discriminant of their characteristic equation is of degree 24 in t , and the roots are found to be $t_0 = 0.5, 0.6342, 0.875, 0.9658$.

- For $t_0 = 0.5$, the containing plane of $\mathcal{E}_A(t_0)$ equals the xy -plane; hence, we check if the two static ellipses $\mathcal{E}_1(t_0) : x^2/25 - y^2/100 - 2y/5 + 3 = 0$ and $\mathcal{E}_2(t_0) : x^2/400 + y^2/1600 + y/80 = 0$ are touching at $t=0.5$. The

characteristic equation for $\mathcal{E}_1(t_0)$ and $\mathcal{E}_2(t_0)$ have two distinct negative roots, and therefore, they are separate. The time $t_0 = 0.5$ is thus rejected.

- For $t_0 = 0.6342$, the containing plane of $\mathcal{E}_A(t_0)$ does not equal the xy -plane. The characteristic equation $f(\lambda) = -0.2860 \times 10^{-6}\lambda^2 - 0.1991 \times 10^{-5}\lambda - 0.3464 \times 10^{-5}$ has a negative double root $\lambda_0 = -3.481$ and the contact point is found to be $(-6.7084, 10.3668, -5)^T$.

Final result: Combining the results from the other subproblems, the two capped cylinders are found to have the first contact at time $t = 0.625$ between $(F_{A,1}, F_{B,1})$ at $(-7.5, 10, 0)^T$ (Fig. 33).

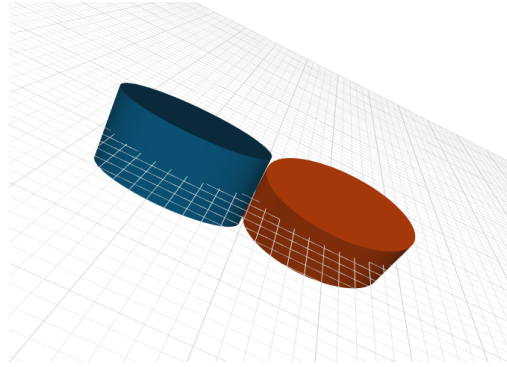


Figure 33: First contact of the two cylinders in example 6.4.

Example 6.5. In this example, we solve the CCD for the two CQMs as shown in Figure 30.

The extended boundary elements of the ring (Figure 30a) are:

- Face $F_{A,1}$: an ellipsoid $\frac{x^2}{20^2} + \frac{y^2}{15^2} + \frac{z^2}{8^2} = 1$.
- Face $F_{A,2}$: a cylinder (concave) $\frac{x^2}{16^2} + \frac{y^2}{12^2} = 1, z = [-24/5, 24/5]$.
- Edge $E_{A,1}$: an ellipse $\frac{x^2}{16^2} + \frac{y^2}{12^2} = 1, z = -24/5$.
- Edge $E_{A,2}$: an ellipse $\frac{x^2}{16^2} + \frac{y^2}{12^2} = 1, z = 24/5$.

The extended boundary elements of the wedge (Figure 30b) are:

- Face $F_{B,1}$: a cylinder $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z = [-10, 10]$.
- Face $F_{B,2}$: a cylinder (concave) $\frac{x^2}{5^2} + \frac{y^2}{5^2} = 1, z = [-10, 10]$.
- Face $F_{B,3}$: a plane $z = 10$.

- Face $F_{B,4}$: a plane $z = -10$.
- Edge $E_{B,1}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z = 10$.
- Edge $E_{B,2}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{10^2} = 1, z = -10$.
- Edge $E_{B,3}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{5^2} = 1, z = 10$.
- Edge $E_{B,4}$: an ellipse $\frac{x^2}{5^2} + \frac{y^2}{5^2} = 1, z = -10$.
- Edge $E_{B,5}$: a line $x = 5, y = 0, z = [-10, 10]$.
- Edge $E_{B,6}$: a line $x = -5, y = 0, z = [-10, 10]$.
- Vertices $V_{B,1} = (-5, 0, 10), V_{B,2} = (5, 0, 10), V_{B,3} = (-5, 0, -10), V_{B,4} = (5, 0, -10)$.

We only consider the parts with $y > 0$ for all elements.

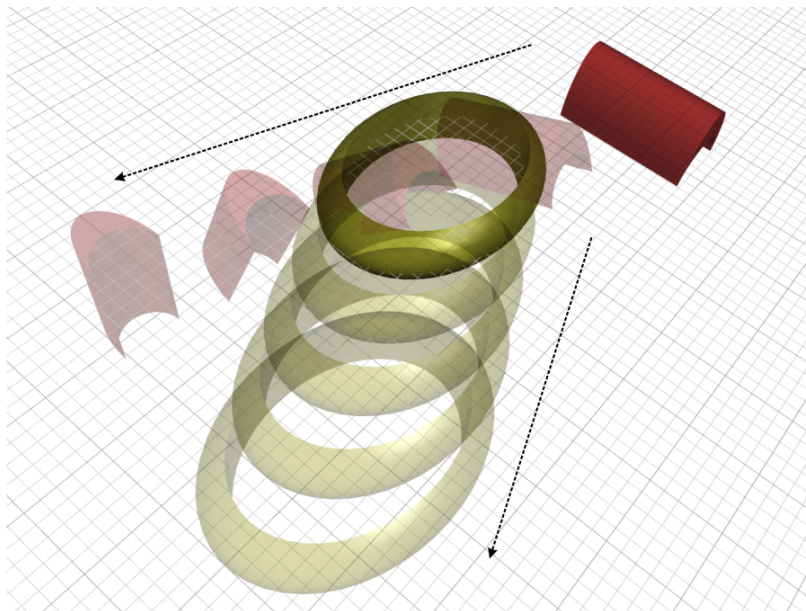


Figure 34: CCD of two general CQMs.

The motions of \mathcal{A} and \mathcal{B} are

$$M_A(t) = \begin{pmatrix} 1 & 0 & 0 & -40t + 10 \\ 0 & 1 & 0 & -20t + 10 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{and}$$

$$M_B(t) = \begin{pmatrix} \alpha t^2 + \alpha t + 1 & 0 & 0 & v_0(t) \\ 0 & \beta t^2 + \sqrt{2}t & -\beta t^2 + \alpha t - 1 & v_1(t) \\ 0 & \beta t^2 - \alpha t + 1 & \beta t^2 + \sqrt{2}t & v_2(t) \\ 0 & 0 & 0 & \alpha t^2 - \alpha t + 1 \end{pmatrix},$$

respectively, where $\alpha = 2 - \sqrt{2}$, $\beta = 1 - \sqrt{2}$, and

$$v_0(t) = (-120 + 60\sqrt{2})t^3 + (180 - 90\sqrt{2})t^2 - (120 - 30\sqrt{2})t + 30,$$

$$v_1(t) = (80 - 40\sqrt{2})t^3 - (100 - 50\sqrt{2})t^2 + (60 - 10\sqrt{2})t - 10,$$

$$v_2(t) = (-40 + 20\sqrt{2})t^3 + (60 - 30\sqrt{2})t^2 - (40 - 10\sqrt{2})t + 10.$$

The subproblems that we need to consider are given in Example 6.2.

The two CQMs have first contact at $t = 0.2723$ with the contact point $(11.9080, 11.7540, 4.8)^T$ between the elements $E_{\mathcal{A},1}$ and $E_{\mathcal{B},4}$ (Figure 35).

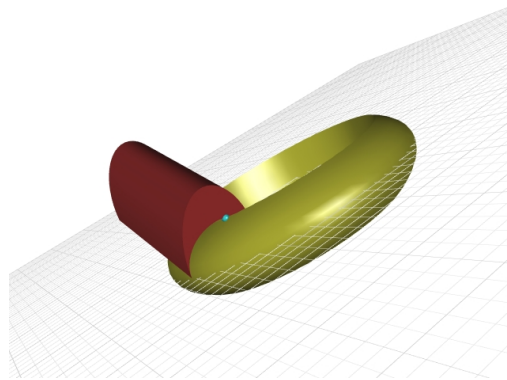


Figure 35: First contact of the two CQMs in example 6.5.

Although we do not deal with CCD of CQMs with general boundary curve, we shall give a brief idea of how this can be solved algebraically. A boundary edge of a general CQM may not be a conic section, but rather the general degree four intersection curve of two boundary quadrics (see, for example, Figure 27). First consider the CCD of an edge and a face respectively from two general moving CQMs. Let a boundary curve \mathcal{I} be the intersection of two quadric surfaces $\mathcal{A} : X^T A X = 0$ and $\mathcal{B} : X^T B X = 0$. The curve \mathcal{I} touches (i.e., has a tangential contact point with) another quadric $\mathcal{C} : X^T C X = 0$ if and only if the three quadrics \mathcal{A} , \mathcal{B} and \mathcal{C} intersect at a multiple point. Hence, the problem is reduced to a study on the intersection of three quadrics in 3D. Our preliminary investigation suggests that the multiple intersection of \mathcal{A} , \mathcal{B} and \mathcal{C} is indicated by that the quartic curve $G(\alpha, \beta, \gamma) \equiv \det(\alpha A + \beta B + \gamma C) = 0$ has a singular point. Hence, to resolve CCD of one moving boundary curve and a moving quadric, we need to develop methods to detect the time t_0 at which the moving planar quartic curve $G(\alpha, \beta, \gamma) = 0$ has a singular point.

The case of CCD of two edges can be treated similarly, but reduced to the study of the intersection of four quadrics, i.e., the two pairs of quadrics defining the two extended boundary curves. This will lead to the study of the quadric net $\alpha A + \beta B + \gamma C + \delta D$ or the quartic surface $H(\alpha, \beta, \gamma, \delta) \equiv \det(\alpha A + \beta B + \gamma C + \delta D) = 0$ with one parameter, the motion time t . Hence, for *contact computation* of general CQMs, techniques based on elimination theory for polynomials will be developed to compute the singularity of the surface $H = 0$. Once the contact points between extended boundary elements are computed, the subsequent step of *contact verification* can be done in the same way as before.

CONCLUSION

We have presented a collection of work related to collision detection for quadrics or conics moving with continuous motions. We adopt an algebraic treatment and our general approach is to characterize a separation condition of two quadrics and conics, based on the roots of their characteristic equation. For two static ellipsoids or two static ellipses, we showed that they are separate if and only if their characteristic equation has two distinct negative roots. We further develop efficient algorithms to deal with CCD based on the separation condition.

For two ellipses, we solve their CCD by finding the zeros of a univariate polynomial. Each root then corresponds to a real touching time instant between the two ellipses. In particular, if the two ellipses are separate initially at $t = 0$, the first contact time will correspond to an external contact. We devise efficient and robust numerical polynomial computation to roots finding. For ellipses moving in the 3D space, we reduce CCD to a 1D problem which can again be solved by using algebraic treatments.

For two ellipsoids, a computation procedure with minimal arithmetic operations is devised to determine whether they are separate or not. In the case where the ellipsoids are moving with on-the-fly motions, we devised a method to compute a separating plane between a pair of separate ellipsoids and use it to accelerate collision detection by exploiting temporal and geometric coherence of the ellipsoids. When the ellipsoids are under pre-specified continuous motions, we formulate a bivariate characteristic function and analyse its zero set. Significant speed-up was realized by developing an efficient scheme to quickly compute the critical points of the zero set of the bivariate

CCD equation, which correspond to the contact time instants of two ellipsoids, and determine whether the ellipsoids are overlapping or separate within a time interval.

Our algebraic formulation based on roots characterization of the characteristic equation is projective invariant, and therefore the above methods on ellipses and ellipsoids can be extended to other conics and quadrics with little adaptation. We have therefore developed also a framework for CCD of the composite quadric models (or CQMs) whose edges are conic segments or line segments.

We have also shown with examples and experiments that our CCD methods are robust and efficient, and are practical to apply to interactive applications.

We believe that there are many other interesting properties of our algebraic condition, which should lead to more efficient geometric algorithms for dealing with ellipsoids (or the other quadrics) and affine deformations. Further applications of the computational tools reported in this thesis should therefore be investigated. The quadrics is the class of degree two implicit surfaces. It would also be interesting to check the applicability of our approach to the other more general implicit surfaces.

BIBLIOGRAPHY

- [1] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press Inc., New York, USA, 1983. ISBN 0-12-049820-0. Transl. by J. Rokne from the original German 'Einführung In Die Intervallrechnung'. (Cited on page 33.)
- [2] David Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. In Forest Baskett, editor, *SIGGRAPH*, pages 19–28. ACM, 1990. ISBN 0-201-50933-4. (Cited on pages 4 and 50.)
- [3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 3540330984. (Cited on page 61.)
- [4] M. Berger. *Geometry*, volume II. Springer-Verlag, Berlin, 1987. (Cited on pages 6 and 14.)
- [5] S. Bischoff and L. Kobbelt. Ellipsoid decomposition of 3D-models. In *Proc. of 1st Int'l Symp. on 3D Data Processing, Visualization and Transmission (3DPVT 2002), Padova, Italy, June 19–21*, pages 480–488, 2002. (Cited on page 2.)
- [6] David Blythe. The direct3d 10 system. *ACM Trans. Graph.*, 25(3):724–734, 2006. (Cited on page 90.)
- [7] Thomas John I'A Bromwich. *Quadratic forms and their classification by means of invariant-factors*, volume 3 of *Cambridge tracts in mathematics and mathematical physics*. Hafner, New York, 1906. (Cited on page 6.)
- [8] Stephen Cameron. Collision detection by four-dimensional intersectin testing. *IEEE Trans. Robot. Autom.*, 6(3):291–302, 1990. (Cited on page 4.)
- [9] Jonathan D. Cohen, Ming C. Lin, Dinesh Manocha, and Madhav K. Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *SI3D*, pages 189–196, 218, 1995. (Cited on page 3.)

- [10] V. Coppola and J. Woodburn. Determination of close approaches based on ellipsoidal threat volumes. In *Proc. of the AAS/AIAA Space Flight Mechanics Meeting*, pages 1013–1023, US, Feb. 7–10 1999. (Cited on page 4.)
- [11] Leonard Eugene Dickson. *Elementary Theory of Equations*. Wiley, New York, 1914. (Cited on pages 16 and 18.)
- [12] Aleksandar Donev, Salvatore Torquato, and Frank H. Stillinger. Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles ii: Applications to ellipses and ellipsoids. *Journal of Computational Physics*, 202: 765–793, 2005. (Cited on page 4.)
- [13] D. H. Eberly. *3D Game Engine Design*. Academic Press, 2001. (Cited on page 1.)
- [14] D. H. Eberly. *Game Physics*. Morgan Kaufmann, 2003. (Cited on page 1.)
- [15] G. Farin. *Curves and Surfaces in Computer Aided Geometric Design*. Academic Press, San Diego, 3rd ed. edition, 1993. (Cited on page 80.)
- [16] Rida T. Farouki. On the stability of transformations between power and Bernstein polynomial forms. *Comput. Aided Geom. Des.*, 8(1):29–36, 1991. (Cited on pages 27 and 33.)
- [17] Rida T. Farouki and V. T. Rajan. On the numerical condition of polynomials in Bernstein form. *Comput. Aided Geom. Des.*, 4(3):191–216, 1987. (Cited on pages 27 and 33.)
- [18] E. Ferré and Jean-Paul Laumond. An iterative diffusion algorithm for part disassembly. In *Proc. IEEE Conf. Robot. Autom.*, pages 3149–3154, New Orleans, USA, April 2004. (Cited on page 4.)
- [19] Elmer G. Gilbert and C.P. Foo. Computing the distance between general convex objects in three-dimensional space. *IEEE J. Robot. Automat.*, 6(1):53–61, 1990. (Cited on page 4.)
- [20] Elmer G. Gilbert, D. W. Johnson, and S. Sathiyaraj. A fast procedure for computing the distance between objects in three-dimensional space. *IEEE J. Robot. Automat.*, 4:193–203, 1988. (Cited on pages 3 and 56.)

- [21] Stefan Gottschalk, Ming C. Lin, and Dinesh Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *SIGGRAPH*, pages 171–180, 1996. (Cited on pages 1, 2, and 89.)
- [22] Naga K. Govindaraju, Ilknur Kabul, Ming C. Lin, and Dinesh Manocha. Fast continuous collision detection among deformable models using graphics processors. *Computers & Graphics*, 31(1):5–14, 2007. (Cited on pages 4 and 5.)
- [23] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Evaluation of collision detection methods for virtual reality fly-throughs. In *Proc. of the 7th Canadian Conference on Computational Geometry*, pages 205–210, 1995. (Cited on page 2.)
- [24] Thomas Horsch and Bert Jüttler. Cartesian spline interpolation for industrial robots. *Comput. Aided Des.*, 30(3):217–224, 1998. (Cited on pages 21 and 68.)
- [25] Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179–210, 1996. (Cited on pages 1 and 2.)
- [26] D.-E. Hyun, S.-H. Yoon, M.-S. Kim, and B. Jüttler. Modeling and deformation of arms and legs based on ellipsoidal sweeping. In *Proc. of Pacific Graphics 2003*, pages 204–212, 2003. (Cited on page 2.)
- [27] Pablo Jiménez, Federico Thomas, and Carme Torras. 3D collision detection: a survey. *Comput. Graph.*, 25(2):269–285, 2001. (Cited on page 5.)
- [28] M. Ju, J. Liu, S. Shiang, Y. Chien, K. Hwang, and W. Lee. A novel collision detection method based on enclosed ellipsoid. In *Proceedings of 2001 IEEE Conference on Robotics and Automation*, pages 21–26, 2001. (Cited on page 2.)
- [29] Bert Jüttler and Michael G. Wagner. Computer-aided design with spatial rational B-spline motions. *ASME J. Mech. Des.*, 118(2):193–201, 1996. (Cited on pages 21 and 68.)
- [30] Bert Jüttler and Michael G. Wagner. Kinematics and animation. In M.S. Kim G. Farin, J. Hoschek, editor, *Handbook of Computer Aided Geometric Design*, pages 723–748. Elsevier, 2002. (Cited on pages 21 and 68.)

- [31] Ku-Jin Kim, Myung-Soo Kim, and Kyungho Oh. Torus/sphere intersection based on a configuration space approach. *Graphical Models and Image Processing*, 60(1): 77–92, 1998. (Cited on page 7.)
- [32] Young J. Kim, Stephane Redon, Ming C. Lin, Dinesh Manocha, and Jim Templeman. Interactive continuous collision detection using swept volume for avatars. *Presence*, 16(2):206–223, 2007. (Cited on pages 4 and 5.)
- [33] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k -dops. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):21–36, 1998. (Cited on page 2.)
- [34] Matt Lee. Next-generation graphics programming on xbox 360. Game Developers Conference, March 2006. URL http://download.microsoft.com/download/d/3/0/d30d58cd-87a2-41d5-bb53-baf560aa2373/Next_Generation_Graphics_Programming_on_Xbox_360.ppt. (Cited on page 90.)
- [35] Christian Lennerz and Elmar Schömer. Efficient distance computation for quadratic curves and surfaces. In *Geometric Modeling and Processing (GMP 2002), Theory and Applications*, pages 60–69, Wako, Saitama, Japan, 10–12 July 2002. (Cited on page 3.)
- [36] J.Z. Levin. Mathematical models for determining the intersections of quadric surfaces. *Comput. Graph. Image Process.*, 1:73–87, 1979. (Cited on page 6.)
- [37] Harry Levy. *Projective and Related Geometries*. Macmillan, New York, 1964. (Cited on pages 11 and 14.)
- [38] Ming C. Lin and Dinesh Manocha. Collision and proximity queries. In J.E. Goodman and eds. J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, 2nd ed. edition, 2004. (Cited on page 5.)
- [39] X. Lin and T.T. Ng. Contact detection algorithms for three-dimensional ellipsoids in discrete element modeling. *International Journal for Numerical and Analytical Methods in Geomechanics*, 19:653–659, 1995. (Cited on page 4.)

- [40] Lin Lu, Yi-King Choi, Wenping Wang, and Myung-Soo Kim. Variational 3d shape segmentation for bounding volume computation. *Computer Graphics Forum*, 26(3): 329–338, 2007. (Cited on pages [xv](#), [2](#), and [3](#).)
- [41] James R. Miller. Geometric approaches to nonplanar quadric surface intersection curves. *ACM Trans. Graph.*, 6(4):274–307, 1987. (Cited on page [6](#).)
- [42] James R. Miller and Ronald N. Goldman. Geometric algorithms for detecting and calculating all conic sections in the intersection of any two natural quadric surfaces. *CVGIP: Graphical Model and Image Processing*, 57(1):55–66, 1995. (Cited on page [7](#).)
- [43] Brian Mirtich. V-clip: Fast and robust polyhedral collision detection. *ACM Trans. Graph.*, 17(3):177–208, 1998. (Cited on pages [3](#) and [57](#).)
- [44] H. Michael Möller. Counting zeros of polynomials by their Bézier ordinates. Ergebnisbericht Nr. 251, Universität Dortmund, 2004. (Cited on page [28](#).)
- [45] T. Nishita, T. W. Sederberg, and M. Kakimoto. Ray tracing trimmed rational surface patches. In *Proc. of ACM SIGGRAPH'90*, pages 337–345, 1990. (Cited on page [78](#).)
- [46] C. Ó Dúnlaing and Chee K. Yap. A retraction method for planning the motion of a disc. *J. Algorithms.*, 6:104–111, 1985. (Cited on page [5](#).)
- [47] Ian J. Palmer and Richard L. Grimsdale. Collision detection for animation using sphere-trees. *Comput. Graph. Forum*, 14(2):105–116, 1995. (Cited on page [1](#).)
- [48] J. Perram, J. Rasmussen, E. Prastaard, and J. Lebowitz. Ellipsoids contact potential: theory and relation to overlap potentials. *Physics Review E*, 54(6):6565–6572, 1996. (Cited on page [4](#).)
- [49] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985. (Cited on page [5](#).)
- [50] Stephane Redon, Abderrahmane Kheddar, and Sabine Coquillart. Fast continuous collision detection between rigid bodies. *Comput. Graph. Forum*, 21(3):279–288, 2002. (Cited on pages [4](#) and [5](#).)

- [51] Stephane Redon, Young J. Kim, Ming C. Lin, Dinesh Manocha, and Jim Templeman. Interactive and continuous collision detection for avatars in virtual environments. In *Proc. IEEE Virt. Reality 2004*, pages 117–124, Chicago, USA, March 2004. IEEE Computer Society. ISBN 0-7803-8415-6. (Cited on pages 4 and 5.)
- [52] Stephane Redon, Ming C. Lin, Dinesh Manocha, and Young J. Kim. Fast continuous collision detection for articulated models. *ASME J. Comput. Inf. Sci. Eng.*, 5(2):126–137, 2005. (Cited on pages 4 and 5.)
- [53] E. Rimon and S.P. Boyd. Obstacle collision detection using best ellipsoid fit. *Journal of Intelligent and Robotic Systems*, 18:105–126, 1997. (Cited on pages 2 and 3.)
- [54] Otto Röschel. Rational motion design - a survey. *Comput. Aided Des.*, 30(3):169–178, 1998. (Cited on pages 21 and 68.)
- [55] P. Schneider and D.H. Eberly. *Geometric Tools for Computer Graphic*. Morgan Kaufmann, San Francisco, 2003. (Cited on page 1.)
- [56] F. Schwarzer, M. Saha, and J.C. Latombe. Exact collision checking of robot paths. In J.D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Algorithmic Foundations of Robotics V*, Springer Tracts in Advanced Robotics, pages 25–41. Springer, 2004. (Cited on page 4.)
- [57] Thomas W. Sederberg. Applications to computer aided geometric design. In D. Cox and B. Sturmfels, editors, *Applications of Computational Algebraic Geometry*, volume 53, pages 67–89. AMS Proc. Symp. Appl. Math., 1998. (Cited on pages 27 and 33.)
- [58] J.G. Semple and G.K. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, London, 1952. (Cited on pages 6, 51, and 53.)
- [59] Michael Ian Shamos and Dan Hoey. Geometric intersection problems. In *Proc. 17th Ann. IEEE Sympos. Found. Comput. Sci.*, pages 208–215, 1976. (Cited on page 5.)
- [60] Micha Sharir. Intersection and closest-pair problems for a set of planar discs. *SIAM J. Comput.*, 14(2):448–468, 1985. (Cited on page 5.)

- [61] S.P. Shiang, J.S. Liu, and Y.R. Chien. Estimate of minimum distance between convex polyhedra based on enclosed ellipsoids. In *Proceedings of the IEEE/RST International Conference on Intelligent Robots and Systems, Takamatsu, Japan, October*, pages 739–744, 2000. (Cited on page 2.)
- [62] Kyung-Ah Sohn, Bert Jüttler, Myung-Soo Kim, and Wenping Wang. Computing distances between surfaces using line geometry. In *Pacific Conference on Computer Graphics and Applications*, pages 236–245. IEEE Computer Society, 2002. ISBN 0-7695-1784-6. (Cited on page 3.)
- [63] D.M.Y. Sommerville. *Analytical Geometry of Three Dimensions*. Cambridge University Press, 1947. (Cited on pages 6 and 51.)
- [64] J.J. Sylvester. On a general method of determining by mere inspection the derivations from two equations of any degree. *Philosophical Magazine*, 16:132–135, 1840. (Cited on page 102.)
- [65] Changhe Tu, Wenping Wang, and Jiaye Wang. Classifying the nonsingular intersection curve of two quadric surfaces. In *Geometric Modeling and Processing (GMP 2002), Theory and Applications*, pages 23–32, Wako, Saitama, Japan, 10-12 July 2002. (Cited on pages 6 and 7.)
- [66] Changhe Tu, Wenping Wang, Bernard Mourrain, and Jiaye Wang. Signature sequence of intersection curve of two quadrics for exact morphological classification. Technical Report TR-2005-09, The University of Hong Kong, 2005. (Cited on page 106.)
- [67] G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997. (Cited on page 2.)
- [68] Michael G. Wagner. Planar rational B-spline motions. *Comput. Aided Des.*, 27(2): 129–137, 1995. (Cited on pages 21, 23, and 33.)
- [69] Wenping Wang. Modelling and processing with quadric surfaces. In M.S. Kim G. Farin, J. Hoschek, editor, *Handbook of Computer Aided Geometric Design*, pages 777–795. Elsevier, 2002. (Cited on page 6.)

- [70] Wenping Wang, Jiaye Wang, and Myung-Soo Kim. An algebraic condition for the separation of two ellipsoids. *Comput. Aided Geom. Des.*, 18(6):531–539, 2001. (Cited on pages 9, 43, 44, 50, 51, 60, and 67.)
- [71] Wenping Wang, Barry Joe, and Ronald N. Goldman. Computing quadric surface intersections based on an analysis of plane cubic curves. *Graphical Models*, 64(6):335–367, 2002. (Cited on pages 6 and 7.)
- [72] Wenping Wang, Ronald N. Goldman, and Changhe Tu. Enhancing levin’s method for computing quadric-surface intersections. *Computer Aided Geometric Design*, 20(7):401–422, 2003. (Cited on pages 6 and 7.)
- [73] Wenping Wang, Yi-King Choi, Bin Chan, Myung-Soo Kim, and Jiaye Wang. Efficient collision detection for moving ellipsoids using separating planes. *Computing*, 72(1-2):235–246, 2004. (Cited on page 84.)
- [74] Itzhak Wilf and Yehuda Manor. Quadric-surface intersection curves: shape and structure. *Computer-Aided Design*, 25(10):633–643, 1993. (Cited on page 6.)
- [75] C.J. Wu. On the representation and collision detection of robots. *Journal of Intelligent and Robotic Systems*, 16:151–168, 1996. (Cited on page 2.)
- [76] Xinyu Zhang, Stephane Redon, Minkyong Lee, and Young J. Kim. Continuous collision detection for articulated models using taylor models and temporal culling. *ACM Trans. Graph.*, 26(3):15, 2007. (Cited on page 5.)