Collision Detection for Curved Rigid Objects in the Context of Dynamics Simulations

Dissertation

zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

von

Thomas Warken

Saarbrücken
12.07.2004

Datum des Kolloquiums: 12.07.2004

Dekan der naturwissenschaftlich-technischen Fakultät I:
Professor Dr. Jörg Eschmeier, Universität des Saarlandes, Saarbrücken

Vorsitzender der Prüfungskommission:
Professor Dr. Reinhard Wilhelm, Universität des Saarlandes, Saarbrücken

Gutachter:
Professor Dr. Elmar Schömer, Johannes Gutenberg Universität, Mainz
Professor Dr. Kurt Mehlhorn, MPI für Informatik, Saarbrücken

akademischer Beisitzer:
Dr. Ernst Althaus, MPI für Informatik, Saarbrücken

Contents

Kurzzusammenfassung

In dieser Doktorarbeit stellen wir Verfahren vor, mit deren Hilfe man Kollisionen zwischen sich bewegenden starren Körpern erkennen kann, deren Oberflächen sich aus gekrümmten Flächen- und Kurvenstücken zusammensetzen. Als Flächentypen betrachten wir dabei algebraische Flächen der Ordnungen eins und zwei (Quadriken) sowie den Torus. Als Kurven erlauben wir Kegelschnittkurven und Schnittkurven zwischen Quadriken. Wir unterscheiden zwei verschiedene Arten der Kollisionserkennung. Die *statische Kollisionserkennung* entscheidet, ob sich zwei stationäre Objekte überlappen, während die *dynamische Kollisionserkennung* für sich bewegende Objekte feststellt, ob diese während eines gegebenen Zeitintervalls kollidieren. Um die Anwendbarkeit dieser Verfahren bei interaktiven Dynamiksimulationen zu gewährleisten, spielen sowohl die Effizienz als auch die numerische Robustheit der Algorithmen eine entscheidende Rolle. Um dem Rechnung zu tragen, führen wir alle Berechnungen darauf zurück, die Nullstellen von Polynomen in einer Variablen zu bestimmen. Für dieses Problem sind effiziente und robuste Algorithmen bekannt. Da deren Laufzeitverhalten und numerische Stabilität stark von den Graden der betrachteten Polynome abhängt, halten wir diese so gering wie möglich. So zeigen wir zum Beispiel für zwei spezielle Klassen von Objekten, dass das Problem der statischen Kollisionserkennung darauf zurückgeführt werden kann, die Nullstellen von Polynomen zu berechnen, deren Grade höchstens vier sind. Ein weiterer Aspekt den wir in dieser Arbeit behandeln ist die Simulation der Dynamik starrer Körper. In diesem Zusammenhang leiten wir ein Verfahren her, mit dem der Vorgang des Rollens eines Körpers auf einer beliebigen Oberfläche simuliert werden kann.

Abstract

In this thesis we present methods for detecting collisions between moving rigid bodies whose boundaries are composed of segments of curved surfaces and curves. The surface types that we consider are algebraic surfaces of degree one and two (quadrics) as well as the torus. The curves that we allow are conic sections and intersection curves between quadrics. We distinguish two different kinds of collision detection. The *static collision detection* decides whether two stationary objects overlap whereas the *dynamic collision detection* checks for two moving objects whether they collide during a given time interval. To make these methods applicable for interactive dynamics simulations, both the efficiency and the robustness of the algorithms play a decisive role. In order to meet these requirements we reduce all computations to the problem of finding the roots of polynomials in one variable. There are efficient and robust algorithms for this task. Since their running time and numerical stability depend heavily on the degrees of the polynomials, we keep these as low as possible. For instance, we show for two special types of objects that the problem of static collision detection can be reduced to determining the roots of polynomials whose degrees are at most four. This thesis also addresses the simulation of the dynamic behaviour of rigid bodies. In this context we develop an approach to simulate the rolling motion of an object on an arbitrary surface.

Ausführliche Zusammenfassung

In der vorliegenden Arbeit behandeln wir das Problem, für zwei sich bewegende starre Körper festzustellen, ob diese miteinander kollidieren oder nicht. Entgegen herkömmlichen Kollisionserkennungsverfahren ist unser Ansatz nicht auf polyederförmige Objekte beschränkt, sondern erlaubt die Verwendung von Körpern, deren Oberflächen sich aus gekrümmten Flächen- und Kurvenstücken zusammensetzen. Die Flächen, die wir dabei zulassen, sind Ebenen, Quadriken (algebraische Flächen der Ordnung zwei) und der Torus. Als Kurven erlauben wir Kegelschnittkurven sowie Schnittkurven zwischen Quadriken. Wir betrachten verschiedene Klassen von Objekten, die sich darin unterscheiden, welche dieser Flächen- und Kurventypen bei der Beschreibung ihrer Oberfläche zugelassen sind. Wir unterscheiden zwei verschiedene Arten der Kollisionserkennung.

- Die *statische Kollisionserkennung* entscheidet, ob sich zwei stationäre Objekte überlappen.

- Die *dynamische Kollisionserkennung* testet, ob zwei sich bewegende Objekte in einem gegebenen Zeitintervall miteinander kollidieren.

Die statische Kollisionserkennung kann in einem dynamischen Szenario benutzt werden, indem man sie zu kurz aufeinander folgenden Zeitpunkten aufruft. Jedoch kann es dabei passieren, dass Kollisionen zwischen sich sehr schnell bewegenden Objekten nicht erkannt werden. Die dynamische Kollisionserkennung vermeidet dies, indem sie nicht nur die Lagen und Orientierungen der Objekte zu festen Zeitpunkten betrachtet, sondern auch deren Bewegungen mit in Betracht zieht.

Kollisionserkennungsalgorithmen kommen in allen Anwendungen zum Einsatz, in denen Objekte sich in einer virtuellen Umgebung bewegen, wobei die Invariante der Durchdringungsfreiheit erhalten bleiben muss. Als Beispiele hierfür lassen sich 3D-Computerspiele nennen, aber auch Anwendungen zur Simulation des dynamischen Verhaltens von Körpern in einem dreidimensionalen virtuellen Szenario. Beispiele für letzteres sind interaktive Montagesimulationen, in denen der Anwender den Zusammenbau mechanischer Teile in einer Virtual-Reality Umgebung simulieren kann. Solche Anwendungen erfordern häufig eine hohe Genauigkeit bei der Repräsentation der Objekte. Werden zur Kollisionserkennung Verfahren verwendet, die mit Polyedern arbeiten, bedeutet dies, dass gekrümmte Oberflächen mittels sehr vieler Polygone approximiert werden müssen. Da die Laufzeit einer Kollisionserkennung stark von der Komplexität der Oberflächenbeschreibung der Objekte abhängt, ist dies insbesondere bei interaktiven Anwendungen problematisch. Daher liegt es nahe, Algorithmen zu entwickeln, die direkt mit den gekrümmten Oberflächen arbeiten können ohne diese polygonal zu approximieren.

Beim Einsatz in Echtzeitanwendungen ist die Effizienz der Kollisionserkennung sehr wichtig. Darüber hinaus kommt es in interaktiven Anwendungen wie der Montagesimulation häufig zu degenerierten Kontaktsituationen. Aus diesem Grund ist auch die numerische Robustheit der Algorithmen von großer Bedeutung. Daher ist es unser

Ansatz, alle nötigen Berechnungen darauf zurückzuführen, die Nullstellen von Polynomen in einer Variablen zu bestimmen. Für dieses Problem kennt man effiziente und robuste Algorithmen. Da deren Laufzeit und numerische Stabilität stark von den Graden der betrachteten Polynome abhängt, halten wir diese möglichst gering. So zeigen wir beispielsweise für zwei der betrachteten Objektklassen, dass das Problem der statischen Kollisionserkennung darauf zurückgeführt werden kann, die Nullstellen von Polynomen höchstens vierten Grades zu bestimmen. Diese können sehr effizient und robust mit Hilfe der Formeln von Cardano und Ferrari bestimmt werden.

Eine prototypische Implementierung unserer statischen Kollisionserkennung für eine der betrachteten Objektklassen und ein Vergleich dieser mit einem Kollisionserkennungssystem für Polyeder zeigt, dass es möglich ist, unsere Verfahren derart zu implementieren, dass die Laufzeiten mit denen herkömmlicher Methoden konkurrieren können. Darüber hinaus zeigt sich bei der Verwendung genauerer Approximationen für den polyederbasierten Algorithmus, dass das Arbeiten mit gekrümmten Oberflächen wie erwartet zu besseren Laufzeiten führen kann.

Ein weiteres Thema, das wir in dieser Arbeit aufgreifen, ist die Simulation des dynamischen Verhaltens starrer Körper. Darunter versteht man das Bewegungsverhalten unter dem Einfluss von Kräften und gegenseitigen Kontakten. Wir beschreiben zwei bekannte Ansätze, dieses Verhalten zu simulieren. Beim *impulsbasierten Ansatz* werden die Bewegungsänderungen, die durch Kollisionen hervorgerufen werden, durch Anwendung der Stoßgesetze in den Kontaktpunkten ermittelt. Dieser Ansatz ermöglicht die Simulation sowohl plastischer als auch elastischer Stöße unter Einbeziehung von Reibung. Ein Nachteil dieses Verfahrens ist, dass nur infinitesimal kurze Kontakte mit nur einem Kontaktpunkt simuliert werden können. Mehrfachkontakte sowie permanente Kontakte müssen durch so genannte Mikrokollisionen modelliert werden. Beim *zwangsbasierten Ansatz* werden Zwangsbedingungen formuliert, die zusammen mit den Newton-Euler Bewegungsgleichungen das Verhalten der Objekte beschreiben. Beispiele für solche Bedingungen sind die Forderung nach Durchdringungsfreiheit in den Kontaktpunkten oder das Verbot von Gleitbewegungen im Falle einer Rollsimulation. Eine bekannte Anwendung des zwangsbasierten Ansatzes ist die Simulation durch Kontaktkräfte. Dort werden die Kräfte ermittelt, die in den Kontaktpunkten wirken, um Durchdringungen zu verhindern. Diese Methode erlaubt das Einbeziehen von Reibung sowie die Simulation von Mehrfach- und Permanentkontakten. Jedoch kann man damit lediglich plastische Stöße simulieren.

Wir entwickeln eine zwangsbasierte Methode zur Simulation des Rollens eines Objektes auf einer beliebigen Oberfläche. Die Zwangsbedingungen, die wir formulieren, fordern, dass in den Kontaktpunkten keine Gleitbewegungen stattfinden dürfen. Zusammen mit den Bewegungsgleichungen leiten wir daraus ein Differentialgleichungssystem her, das die Rollbewegung beschreibt. Wir beweisen, dass dieses System bei gegebenen intitialen Geschwindigkeiten eine eindeutige Lösung hat. Mit dieser Methode waren wir in der Lage, das Hinabrollen eines "Oloid" genannten Objektes auf einer schiefen Ebene in Echtzeit zu simulieren.

Danksagungen

An dieser Stelle möchte ich mich bei Herrn Prof. Dr. Elmar Schömer für die Betreuung dieser Arbeit bedanken. Die vielen Diskussionen mit ihm und seine Anregungen trugen maßgeblich zu ihrer Entstehung bei. Mein Dank gilt auch Herrn Prof. Dr. Kurt Mehlhorn für die Möglichkeit, in seiner Gruppe am Max-Planck-Institut für Informatik in angenehmer und freundlicher Atmosphäre zu arbeiten.

Weiterhin möchte ich mich bei meinen Freunden und Kollegen am MPII für ihre Unterstützung in Form von fruchtbaren Diskussionen und freundschaftlicher Zusammenarbeit bedanken. Insbesondere sind hier Christian Lennerz, Sven Thiel, Nicola Wolpert, Ernst Althaus und die 11:30 Mittagessensgruppe zu erwähnen, durch die die Zeit am MPII eine schöne Zeit geworden ist.

Ein besonderer Dank geht auch an Herrn Prof. Dr. Günter Hotz, der meine Diplomarbeit mit dem Titel "Berechnung von Kontaktkräften für eine zwangsbasierte Dynamiksimulation" betreut und mich damit an die Thematik dieser Arbeit herangeführt hat.

Schließlich danke ich meiner Mutter und ihrem Lebensgefährten, die durch ihre uneingeschränkte Unterstützung erheblich zur Entstehung dieser Arbeit beigetragen haben.

1 Introduction

1.1 Problem Definition

In this thesis we present methods to solve the collision detection problem for objects with curved surfaces in the context of dynamics simulations for rigid bodies. We describe known approaches to simulate the dynamic behaviour of rigid objects and derive a new method for the simulation of rolling motions.

Collision Detection

The collision detection problem that we will solve can be formulated as follows. Given a scene with moving rigid objects whose boundaries consist of curved surface patches, detect the collisions between them. These methods are intended to be used in interactive virtual reality applications and should be applicable for the simulation of the dynamic behaviour of rigid objects. Therefore they must be

- *efficient*, since interactive applications require real-time performance and
- *robust*, since in many interactive dynamics simulations (e.g. assembly simulations) it is very likely that situations are geometrically degenerate.

We consider different classes of objects. These differ in the curve and surface types the boundaries of the objects consist of. The surface types that we use are quadrics (with the plane as a special case) and the torus. As curve types we allow conics (with the straight line as a special case) and intersection curves between quadrics. We present two different kinds of collision detection algorithms.

- The *static collision detection* decides for two stationary objects whether they overlap or not.
- The *dynamic collision detection* decides whether two moving objects will interpenetrate during a given time interval.

The static collision detection algorithm can be used in a scene with moving objects by applying it at discrete points in time. But in order to ensure that no collisions are missed one has to take care that the time steps between these discrete points are sufficiently small. The dynamic collision detection is a possibility to avoid this difficulty.

In order to meet the requirements for efficiency and robustness, we reduce all computations to the problem of finding the roots of univariate polynomials. For this problem there are well known algorithms that are efficient and robust. In addition to detailed derivations of these polynomials we give careful analyses of their degrees. Since the running time as well as the accuracy of the root finding algorithms depend highly on these degrees we will keep them as low as possible.

Dynamics Simulation

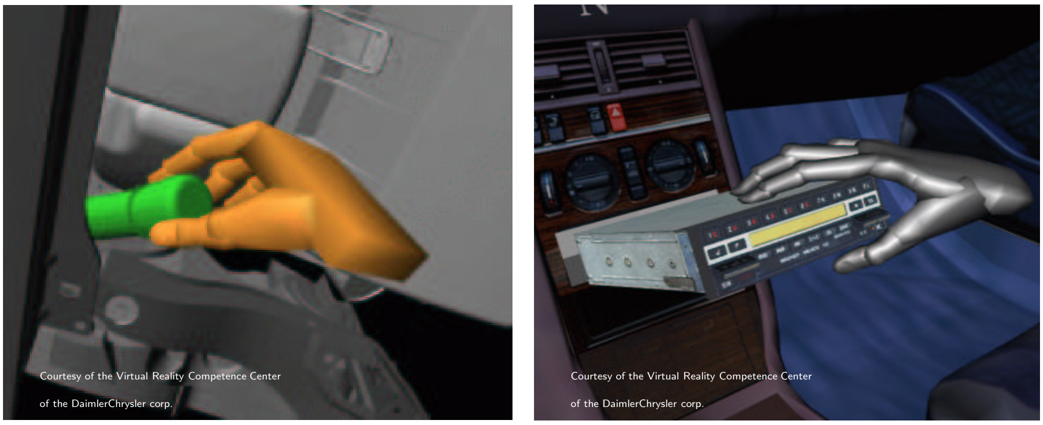
Simulating the dynamics of rigid objects means simulating their motions under the influence of forces and mutual contacts. We describe two classical simulation methods, namely the impulse based and the constraint based method. The former one computes the reactions to the collisions by applying the laws of impact in the contact points and the latter one by formulating geometric and dynamic constraints in addition to the equations of motion. We derive a new constraint based approach for the simulation of an object with curved boundary that rolls on an arbitrary surface. This means we are able to simulate the motion of an object that is in contact with a surface under the assumption that the sticking friction in the contact points forbids sliding.

1.2 Motivation

In a variety of areas simulation techniques are used to create virtual environments that behave in some sense realistic. The film industry and the computer game industry, for instance, make use of these computer simulated environments. Movie scenes that are too difficult or even impossible to shoot in the classical way are produced using such methods. In 3D computer games the players navigate through a virtual world and interact with each other and with their environment. But not only the entertainment industry benefits from these methods. Also the manufacturing industry uses simulation techniques to reduce the costs during the development phase of a product. Simulation software can be applied to test mechanical parts both for assemblability and functionality. The automotive industry also uses such techniques to perform virtual crash tests. With this *virtual prototyping*, errors in the product design can be detected in an early stage of the development phase before real prototypes are constructed. In this way the duration as well as the expenses of this phase can be drastically reduced.

In all applications where virtual objects are moved and the invariant that no objects overlap has to be maintained, collision detection algorithms are necessary. Such applications include 3D computer games as well as the just mentioned simulations for virtual prototyping. In many cases the objects in the virtual environment are manipulated interactively. Hence, the collision detection must be capable of performing in real-time.

In most simulation applications it does not suffice to just detect collisions between objects. Imagine a scenario as shown in figure 1.1. These images show screenshots from two simulations that were performed in the Virtual Reality Competence Center of the DaimlerChrysler corporation. The left image shows the changing of a light bulb in the



Courtesy of the Virtual Reality Competence Center of the DaimlerChrysler corp.

Courtesy of the Virtual Reality Competence Center of the DaimlerChrysler corp.

Figure 1.1: Changing a light bulb (left) and mounting a car radio (right) in a virtual environment.

head-light of a car and the right image shows the mounting of a car radio. In both scenarios it was possible for an engineer to move a virtual object, namely a light bulb and a radio, interactively in a virtual world. Collisions between the moving object and its environment were recognized by the software. But if the movement of the objects would have just stopped whenever a collision occurred it would have been a very difficult task to move the light bulb out of its socket or to insert the radio into the slot. However, in a real environment these tasks are not that difficult since due to the collisions the motions of the objects change and thus they slip into the right direction. Hence, a simulation software must also be able to compute physically correct reactions to the collisions. This is the subject of chapter 4 of this thesis.

Currently used collision detection algorithms are based on the assumption that the involved objects are polyhedra. This assumption can be justified by the fact that any curved object can be approximated arbitrarily well by a polyhedron. The basic tests that are performed by a collision detection algorithm are collision tests between single faces. Since the faces of polyhedra are polygons, the basic tests can be performed very efficiently for these objects. However, there are applications that require a very high accuracy. This means that the objects have to be approximated very well, i.e. by a large number of polygons. Since the running time of a collision detection algorithm between two objects depends highly on the number of boundary elements (faces, edges and vertices), higher accuracy implies a longer response time. This is particularly problematic if real-time performance is required, e.g. in a scenario where the objects are manipulated interactively. Hence, it is desirable to work with the curved objects directly instead of approximating them. Therefore, we present methods for detecting collisions between objects with curved surfaces.

1.3 Previous Work

1.3.1 Static Collision Detection for Polyhedra

There are many publications dealing with the problem of static collision detection for polyhedra. Since the basic tests are rather simple for polygonal faces, these publications mainly concentrate on acceleration techniques. These techniques either reduce the number of pairs of faces for the basic tests or the number of pairs of objects that have to be tested in a large scale scene. The number of pairs of faces is usually reduced using bounding volume hierarchies. Different types of bounding volumes are used. These include spheres ([Qui94, Hub95, Hub96, PG95]), oriented boxes ([GLM96]) and fixed direction hulls ([HKM$^+$96]) with the special case of axis-oriented boxes ([ZF95, Zac97]). Most of these techniques can also be used for objects with curved surfaces. In fact, we will describe methods published in [Rei01] to compute smallest bounding volumes for sets of curved surface patches. In [EL01] a hierarchical convex decomposition of the boundaries is used to compute a hierarchy of bounding volumes. This is not possible for curved objects, since their boundaries do not necessarily have a convex decomposition. For the reduction of the number of objects to be tested in a scene, [Mir97] describes a hierarchical space partitioning approach which is based on [Ove92]. We will see that this approach can be used for curved objects, as well. In [CLMP95], an approach based on projecting axis aligned bounding boxes onto the coordinate axes and sorting the resulting intervals is used to find the pairs of objects that possibly collide. A disadvantage of this approach is that the axis aligned boxes have to be recomputed whenever an object rotates. For curved objects, this computation is rather expensive in general. An algorithm for static collision detection (as well as various other queries) between polyhedra using graphics hardware acceleration can be found in [HZLM02]. Their method is to use first a bounding volume hierarchy to localize the potential collision regions. These regions are then point-sampled uniformly. The graphics hardware pixel framebuffer is used as a 2D slice of this sample and the tests are performed slice-wise. In this way all queries become pixel operations that can be performed very efficiently by the graphics hardware.

1.3.2 Dynamic Collision Detection for Polyhedra

Publications on dynamic collision detection for polyhedral objects include [Eck99] and [ES99]. They approximate the volume swept by the bounding volumes in order to use the hierarchy in a dynamic scenario. For the basic tests, a simple interpolation scheme is used to find the roots of signed distance functions between vertices and faces or between edges. For curved objects it is not clear how to define these signed distances. In [RKC01] it is proposed to use an interval arithmetic based approach to check the bounding volumes for collision. In that publication, also the basic tests are performed using interval arithmetic.

There are also theoretical results for the dynamic collision detection problem for two triangulated polyhedra with a total of n triangles. In [Sch94] it is shown that the collision test can be performed in time $O(\log^2 n)$ if both polyhedra are convex and their motions are translations. If only one polyhedron is convex it is shown that a running time of $O(n \log n)$ is possible. In [ST95a] it was proven that for two non-convex polyhedra a running time of $O(n^{1.6+\varepsilon})$ in the case of translational motions and $O(n^{1.\overline{6}+\varepsilon})$ in the case of rotational motions is possible. For the case that the motion is described by polynomial trajectories it was shown in [ST96] that still a sub-quadratic running time of $o(n^2)$ can be obtained. In [SSW95] the bit complexity of the collision detection problem for polyhedra was analyzed. This is important if one wants to use exact arithmetic. It was shown that for the case that all input data are given by L-bit numbers the collision test can be performed using at most $14L + 22$ bits.

1.3.3 Collision Detection for Curved Surfaces

There is also previous work dealing with the static collision detection problem for curved objects or the computation of intersections between such objects. For parametric surfaces there are publications about subdivision methods (e.g. [Sny92, Dok97]). These work by recursively subdividing the parameter domains of two surface patches. These methods are intrinsically approximate and if they are used for high precision they suffer from data proliferation and are therefore time consuming. There are also subdivision methods for the dynamic collision detection between time dependent parametric surfaces ([vHBZ90, SWF$^+$93]). In [MC91] and [KM97] a method based on computations with bivariate matrix polynomials has been published for the intersection between two general parametric surfaces. An implicit form of one of these surfaces is represented as the determinant of a matrix. The parameterization of the other surface is inserted into this matrix. In our case we know parametric as well as implicit representations of the surfaces. Hence, we do not have to perform these time consuming matrix computations. In [LM95] algebraic surfaces are tested for intersection by looking for real roots of a system of three equations in four unknowns. A configuration space approach to compute the intersections between a torus and natural quadrics has been published in [Kim98]. We also use this approach in our static collision detection algorithm.

1.3.4 Dynamics Simulation

In the field of dynamics simulations there are two fundamental approaches, namely the impulse based approach and the constraint based approach. The impulse based method computes the collision reactions by applying the laws of impact in the contact points. Publications dealing with this approach include [MC95], [Mir96b] and [Len00]. An example for the constraint based method is the computation of so-called contact forces. These are the forces that act in the collision points and prevent the objects from interpenetrating. In [SKL98] these are modelled as spring forces. A different way to compute them is by formulating geometric constraints in the contact points. This method goes back to [Bar94] and was refined by later works including [BS98] and [War99]. The publications [ST95b], [SS98] and [Sau03] extend the approach by including friction.

1.3.5 Software

Finally, we want to mention software packages that have been developed in the field of collision detection and dynamics simulation for polyhedral objects. Pure collision detection packages include I-COLLIDE, which is presented in [CLMP95] and RAPID, which is based on the results presented in [GLM96]. The package SWIFT++ which is based on [EL01] is also able to compute closest features, approximate and exact distances and can perform tolerance verifications, i.e. decide whether two objects are closer than a given tolerance. The IGOR system (see [Sch94]) is a system for assembly planning and simulation. It can assist an engineer to develop a scheme for the chronological and spacial coordination of an assembly process. Based on this, the practicability of a (robot aided) assembly process can be proven by a simulation. Whereas this system considers only the geometry and kinematics, the main focus of the SiLVIA library (see [HKL$^+$99]) is the real-time simulation of the dynamic behaviour of colliding rigid objects. Finally, we mention the virtual reality platform of the DaimlerChrysler corporation which is called DBVIEW. It contains a real-time collision detection (see [Eck99]) and a module for the computation of contact forces (see [Buc99]) and was used to perform the simulations shown in figure 1.1.

1.4 Our Contributions

- We present a generic algorithm for the static collision detection problem for rigid objects with curved boundaries. We describe specializations of this algorithm for certain classes of objects. We reduce all computations to the problem of finding the roots of polynomials in one variable and prove upper bounds for the degrees of these polynomials. In particular we prove that these degrees are at most four in the cases that the boundaries of the objects consist of

  - patches of quadrics and segments of conics and
  - patches of planes, spheres circular cones, circular cylinders and tori and segments of straight lines and circles.

  This means that there exist closed form solutions for the roots and hence, they can be computed very efficiently and accurately. We implemented our approach prototypically for the so called natural quadratic complexes. These are objects whose faces are embedded on planes, spheres, circular cones and circular cylinders and whose edges are embedded on circles and straight lines. By comparison of our implementation with the collision detection system SWIFT++ that works on polyhedra we could verify that it indeed makes sense to work with curved objects instead of approximating them.

- We present a generic algorithm for the dynamic collision detection problem for curved rigid objects and describe specializations for certain classes of objects. To our knowledge, this is the first time that a dynamic collision detection algorithm for complex classes of objects with curved surfaces is published. We reduce all our computations to root finding problems for univariate polynomials. We prove upper bounds for the degrees of these polynomials in the cases of pure translational motions, pure rotational motions and superpositions of translations and rotations. We give a thorough case distinction on the types of curves and surfaces involved in the collision test in order to keep the degrees as low as possible.

- A subproblem that has to be solved in both the static and the dynamic collision test is to decide whether a given point on a surface lies inside or outside a face that is embedded on that surface. Typically, this test is performed in the parameter space of the surface. A disadvantage of this approach is that the edges that bound the face have to be mapped to the parameter space. But simple algebraic curves in 3-space are usually much more complicated in the parameter space of a surface. We present a method to perform this test directly in 3-space. The crucial operation that our method uses is computing the points of intersection between the edges bounding the face and a plane. In this way we exploit the low algebraic degrees of the curves containing these edges.

- A subproblem of the dynamic collision test is the following. Given two objects that are in touch at a point $\mathbf{p}$, decide whether there is a penetration locally at $\mathbf{p}$ in the immediate future. We show that this penetration test can be formulated as the problem to decide whether a point belongs to the boundary of a full-dimensional cell in a semi-algebraic set in $\mathbb{R}^4$. It is an open question how this problem can be solved efficiently in all cases. For non-degenerate situations, however, we present a simple and efficient way to perform this test.

- If an object is in contact with a surface and a sliding motion is not possible, e.g. because of high sticking friction, then it will perform a rolling motion. We present a method to simulate this behaviour for arbitrary objects in contact with arbitrary surfaces. We derive a system of ordinary differential equations that describes the motion of the object and prove that this system is non-singular. Thus, the motion is uniquely determined by the initial velocities and the external forces such as gravity. With this approach we were able to simulate an object called *the oloid*[1] (see figure 1.2) rolling down an inclined plane in real-time.

[1] The oloid is the convex hull of two circles lying in perpendicular planes such that each of them contains the center of the other.

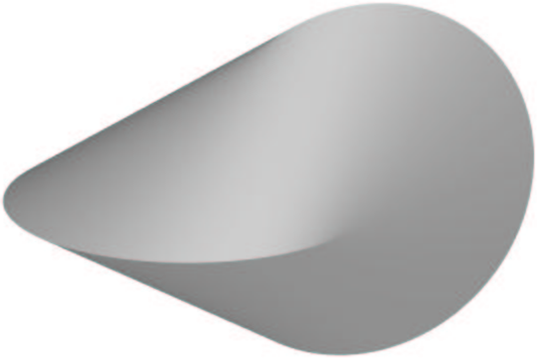


Figure 1.2: The oloid

1.5 Outline

This thesis is structured as follows.

- In chapter 2 we introduce the basic concepts that are used throughout the thesis. We begin with giving a short overview over the notation used in the following. Then, we give a rather general definition of the notion of rigid objects and introduce the special object classes that are used in the following. We define the curve and surface types that the boundaries of these objects consist of and present some mathematical background including a new result concerning the existence of tangential intersection points between quadrics and between conics. Finally, we describe how physical properties such as mass or moments of inertia can be computed for a rigid object. We present a method to perform these computations on the curved objects rather than on approximations in order to achieve the highest possible precision.

- Chapter 3 is the main part of the thesis. It starts with a section about heuristics for fast feature culling. We describe two such heuristics, namely the bounding volume hierarchies and the space partitioning. In the next section we present a static collision test. We first describe a generic algorithm that works for all rigid objects corresponding to our general definition. Then, we specialize the generic algorithm for our special objects classes. Finally, we present a dynamic collision test. Again, we first describe a generic algorithm that works for general rigid objects. We specialize the generic algorithm for two of our special object classes.

- In chapter 4 we describe two approaches to simulate the dynamic behaviour of rigid objects subject to mutual contacts. These are the impulse based and the constraint based simulation method. In the context of the latter method, we present a new approach for the simulation of rolling motions.

- We close the thesis with chapter 5 where we give a conclusion and say a few words about possible further research.

2 Basics

In this chapter we present some basic definitions and concepts that are used throughout the thesis. We start by introducing some conventions of notation that we will abide by in the following. These conventions mainly concern the typesets used for vectors, matrices etc. Next, we introduce the concept of rigid objects. We first give a definition and then describe how a data structure can be organized to represent rigid objects. We decided to use the so called boundary representation which represents a rigid object by describing its boundary. We will also define the classes of rigid objects that we will deal with in the following chapters. The curve and surface types that the boundaries of the objects in these classes consist of will then be described in the next section. After that, we will present some mathematical preliminaries. That section contains – amongst other things – a new necessary condition for two general quadrics or conics to have a real tangential intersection point. This result is a generalization of a condition given in [WWK01]. The chapter will be closed by a section about the physical properties that we assign to each object. These properties will be important in chapter 4 where we talk about the dynamic behaviour of rigid objects.

2.1 Notation

In this section we briefly want to introduce some notations that will be used in the following.

- **Matrices** will usually be denoted by bold-face upper-case letters. Examples include $\mathbf{M}, \mathbf{R}, \mathbf{I}$. We write $\mathbf{E}$ for the $(n \times n)$-identity matrix and $\mathbf{0}$ for the zero-matrix. The transposed of $\mathbf{A}$ will be denoted as $\mathbf{A}^T$.

- **Vectors** and vector-valued functions will be written as bold-face sans-serif letters, as for example $\mathsf{a}, \mathsf{f}, \mathsf{F}, \boldsymbol{\omega}$. A vector $\mathsf{a} \in \mathbb{R}^k$ is usually viewed as a column vector

  $$\mathsf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}.$$

  Row vectors are written as transposed column vectors, such as a^T. We write $\mathsf{0}$ for the zero vector. The scalar product of two vectors a and b is written as matrix product $\mathbf{a}^T\mathbf{b}$. For a vector $\mathbf{a} = [a_1, a_2, a_3]^T \in \mathbb{R}^3$ we write

$$\mathbf{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

for the skew-symmetric matrix associated with $\mathbf{a}$. With this definition we have the equality $\mathbf{a} \times \mathbf{b} = \mathbf{a}^\times \cdot \mathbf{b}$ for vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$.

- **Sets** are denoted by upper-case script letters, such as $\mathcal{A}, \mathcal{O}, \mathcal{K}$.
- **Quaternions** will be written as bold-face lower-case gothic letters, such as $\mathfrak{p}$ and $\mathfrak{q}$. The components of a quaternion $\mathfrak{q}$ are denoted as $q_0, \ldots, q_4$. We sometimes write

$$\mathfrak{q} = \begin{bmatrix} q_0 \\ \mathbf{q} \end{bmatrix}$$

with $\mathbf{q} = [q_1, q_2, q_3]^T$. The adjoint quaternion of $\mathfrak{q}$ is

$$\mathfrak{q}^* = \begin{bmatrix} q_0 \\ -\mathbf{q} \end{bmatrix}.$$

The length of a quaternion $\mathfrak{q}$ is $|\mathfrak{q}| = \sqrt{\mathfrak{q}\mathfrak{q}^*}$. For a vector $\mathbf{a}$ and a quaternion $\mathfrak{q}$ we sometimes write $\mathbf{a}\mathfrak{q}$ for the quaternion product

$$\begin{bmatrix} 0 \\ \mathbf{a} \end{bmatrix} \cdot \mathfrak{q}.$$

2.2 Rigid Objects

In this section we will define what we mean in the following by *rigid objects*. We start with a very general definition of this term and continue with a subsection about the representation of these objects. At the end of the section we will define the object classes that will be considered in the remainder of this thesis.

2.2.1 General Definition

Throughout this thesis, rigid objects will be non-deformable three-dimensional objects. To make this more precise we start with a very general definition of the term *object*, that is similar to the definition of *multishell manifold solid* in [Hof89]. Before we state this definition, recall that the boundary of an n-manifold with boundary is always an $(n-1)$-manifold without boundary (cf. [GH81]). We say that a compact 2-manifold in $\mathbb{R}^3$ is of *bounded variation* if every line and every plane intersects it in finitely many connected components.

Definition 2.1. *A point-set $\mathcal{O} \subset \mathbb{R}^3$ is called object, if it is a compact, connected 3-manifold with boundary $\partial\mathcal{O}$. We also require $\partial\mathcal{O}$ to be of bounded variation.*

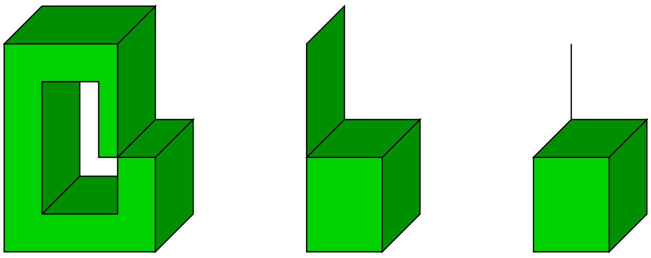


Figure 2.1: Examples for point-sets that are not 3-manifolds with boundary.

The main difference between this definition and the one given in [Hof89] is the requirement for compactness. The reason for this demand is that for dynamics simulations only bounded objects are of interest, whereas in solid modeling it might be convenient to allow unbounded objects, as well. We define objects as 3-manifolds with boundary to avoid point-sets like those shown in figure 2.1. We require $\partial\mathcal{O}$ to be of bounded variation in order to avoid "ruffled" or fractal boundaries. The reason for this is that we want to be able to decide whether a given point lies inside or outside an object by counting the intersections between a ray starting in that point and the boundary of the object. If the boundary is of bounded variation we can be sure to find a ray that intersects it in finitely many points. The objects that we consider in the following chapters will always have boundaries consisting of finitely many subsets of algebraic surfaces. It is easy to show with Bezout's theorem that algebraic surfaces are always of bounded variation.

A moving object is called *rigid* if its shape does not change during the motion. We want to state more formally what this means. A moving object is a function $\mathcal{O}(t)$ such that for each $t \geq 0$ the set $\mathcal{O}(t)$ is an object according to definition 2.1.

Definition 2.2. *A moving object $\mathcal{O}(t)$ is called rigid if there is an object $\tilde{\mathcal{O}}$, a continuous rotation matrix $\mathbf{R}(t)$ and a continuous vector $\mathbf{c}(t)$ such that*

$$\mathcal{O}(t) = \{\, \mathbf{R}(t)\tilde{\mathbf{p}} + \mathbf{c}(t) \mid \tilde{\mathbf{p}} \in \tilde{\mathcal{O}} \,\}.$$

This means that a rigid object is only allowed to change its position and orientation during the motion.

2.2.2 Boundary Representation

Now that we have defined what rigid objects are, we want to describe how to represent their geometry. One of the most common representations for solids is the boundary representation. As the name suggests, an object is represented only by its boundary. As already mentioned before, we only consider objects whose boundaries consist of finitely many subsets of algebraic surfaces. An algebraic surface $\mathcal{S}$ is defined as

$$\mathcal{S} = \{\, \mathbf{x} \in \mathbb{R}^3 \mid f(\mathbf{x}) = 0 \,\},$$

where f is a polynomial in $\mathbb{R}[x_1, x_2, x_3]$. The equation $f(\mathbf{x}) = 0$ is called the *implicit form* of $\mathcal{S}$. We call the region containing all points $\mathbf{x}$ with $f(\mathbf{x}) < 0$ the *interior region* with respect to $\mathcal{S}$, and the region consisting of those points $\mathbf{x}$ with $f(\mathbf{x}) > 0$ the *exterior region* with respect to $\mathcal{S}$. In this way the surface $\mathcal{S}$ is oriented. Note that the orientation of $\mathcal{S}$ can be flipped by multiplying the polynomial f by -1. We call a point $\mathbf{x} \in \mathcal{S}$ a *singular point* of $\mathcal{S}$ if $\nabla f(\mathbf{x}) = 0$. The *normal* of a non-singular point $\mathbf{x} \in \mathcal{S}$ is the vector

$$\mathbf{n}_\mathbf{x} = \frac{\nabla f(\mathbf{x})}{|\nabla f(\mathbf{x})|}.$$

The normal $\mathbf{n}_\mathbf{x}$ of $\mathbf{x}$ always points from the interior to the exterior region w.r.t. $\mathcal{S}$. This can be seen as follows. Define the function $g(t) = f(\mathbf{x} + t\mathbf{n}_\mathbf{x})$. Then the derivation of g w.r.t. t is $g'(t) = \mathbf{n}_\mathbf{x}^T \nabla f(\mathbf{x} + t\mathbf{n}_\mathbf{x})$. Obviously, $g'(0) = \mathbf{n}_\mathbf{x}^2 \cdot |\nabla f(\mathbf{x})| > 0$. So, if we start in the point $\mathbf{x}$ and then follow a straight line in the direction of $\mathbf{n}_\mathbf{x}$, then the value of f initially increases, i.e. we enter the exterior region w.r.t. $\mathcal{S}$.

Definition 2.3. *A subset $\mathcal{F} \subset \mathcal{S}$ is called a face (embedded in $\mathcal{S}$), if it is a compact, connected 2-manifold with boundary $\partial\mathcal{F}$, and if $\mathcal{F}$ contains no singular points of $\mathcal{S}$.*

The orientation of $\mathcal{F}$ is given by the orientation of $\mathcal{S}$. In the following, the boundary of a face will always consist of finitely many subsets of intersection curves between algebraic surfaces. Note that $\partial\mathcal{F}$ might be empty (choose $\mathcal{S}$ as a sphere and $\mathcal{F} = \mathcal{S}$). We will represent the boundary of an object $\mathcal{O}$ as the union of finitely many faces. A face $\mathcal{F} \subset \partial\mathcal{O}$ is called a face of $\mathcal{O}$.

Let $\mathcal{C}$ be an intersection curve between two algebraic surfaces and let $\mathbf{c}(t)$ be a parameterization of $\mathcal{C}$. The tangent vectors $\dot{\mathbf{c}}(t)$ define the orientation of $\mathcal{C}$. This orientation can be flipped by replacing the parameterization by $\tilde{\mathbf{c}}(t) := \mathbf{c}(-t)$. We call a point $\mathbf{x} = \mathbf{c}(t_0)$ a *singular point* of $\mathcal{C}$ if the tangent $\dot{\mathbf{c}}(t_0)$ does not exist.

Definition 2.4. *A subset $\mathcal{E} \subset \mathrm{C}$ is called an edge (embedded on $\mathcal{C}$), if it is a compact, connected 1-manifold with boundary $\partial\mathcal{E}$, and if $\mathcal{E}$ contains no singular points of $\mathcal{C}$.*

The orientation of $\mathcal{E}$ is given by the orientation of $\mathcal{C}$. Note that $\partial\mathcal{E}$ either is empty or consists of exactly two points, the *endpoints* of $\mathcal{E}$. Endpoints of edges are also called the *vertices* of $\mathcal{E}$. We call the first vertex of $\mathcal{E}$ with respect to the orientation of $\mathcal{E}$ the *start-vertex* and the second one the *end-vertex*. We will represent the boundary of a face $\mathcal{F}$ as the union of finitely many edges. An edge $\mathcal{E} \subset \partial\mathcal{F}$ is called an edge of $\mathcal{F}$ and the vertices of $\mathcal{E}$ are vertices of $\mathcal{F}$. If $\mathcal{F}$ is a face of an object $\mathcal{O}$ then $\mathcal{E}$ is also called an edge of $\mathcal{O}$ and the vertices of $\mathcal{E}$ are vertices of $\mathcal{O}$. If $\mathcal{E}$ is an edge of a face $\mathcal{F}$, then $\mathcal{F}$ is said to be *adjacent* to $\mathcal{E}$ and $\mathcal{E}$ is said to be *incident* to $\mathcal{F}$. If $\mathcal{E}$ is incident to two faces $\mathcal{F}_1$ and $\mathcal{F}_2$, then $\mathcal{F}_2$ is adjacent to $\mathcal{F}_1$.

Definition 2.5. *Let $\mathcal{E}$ be an edge of a face $\mathcal{F}$. We say that the orientation of $\mathcal{E}$ is induced by the orientation of $\mathcal{F}$ if the following holds for each point $\mathbf{p} \in \mathcal{E}$. If one looks at $\mathbf{p}$ in the opposite direction of the normal $\mathbf{n}$ of $\mathcal{F}$ in $\mathbf{p}$, then the interior of $\mathcal{F}$ lies to the left of the oriented edge $\mathcal{E}$.*

Definition 2.6. *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two adjacent faces. We say that they are coherently oriented if for each of their common edges $\mathcal{E}$ the orientations of $\mathcal{E}$ induced by $\mathcal{F}_1$ and $\mathcal{F}_2$ are opposite.*

Coherent orientation of $\mathcal{F}_1$ and $\mathcal{F}_2$ implies that along common edges both faces agree on how to divide the space into an interior and an exterior region.

We call the connected components of the boundary of an object $\mathcal{O}$ the *shells* of $\mathcal{O}$. The connected components of the boundary of a face $\mathcal{F}$ are called the *loops* of $\mathcal{F}$. Now we are able to give a first description of the boundary representation of an object $\mathcal{O}$.

1. An object is represented by a singly linked list of its shells.

2. A shell is represented by a singly linked list of its faces.

3. A face is represented by the surface that it is embedded in and by a singly linked list of its loops.

4. A loop is represented by a singly linked list of its edges.

5. An edge is represented by the curve that it is embedded on and by its start- and end-vertex.

6. A vertex is represented by a point.

We call the components shell, face, loop, edge and vertex the *topologic components* of the representation, whereas the components surface, curve and point are called its *geometric components*. We will describe later how to represent the geometric components. The above representation is not sufficient, yet. It still allows to represent structures that are not objects according to our definition. It is possible to represent objects like those in figure 2.1, point sets that do not contain any volume, surfaces that are not orientable, etc. Moreover, it would be convenient to have the possibility to easily enumerate all edges or vertices of a face in the order of the orientation that is induced by the orientation of the face.

As a first improvement, we add a test to the representation that checks whether the two vertices of an edge lie on the same connected component of the curve that contains the edge. Next, we make sure that the start-vertex is a predecessor of the end-vertex with respect to the orientation of the curve. Now, we must check that the edge does not contain self intersections as sketched in figure 2.2.

As another improvement of the representation described above we demand that each vertex of a face is a vertex of exactly two edges of that face (of course it may also be a vertex of an edge of another face). This implies that loops always represent compact 1-manifolds without boundary. This avoids the situation shown in the rightmost picture of figure 2.1.

As two adjacent faces induce opposite orientations on common edges it is convenient to introduce one more topological component to the representation, namely the *coedge*. We change the representation described above as follows.

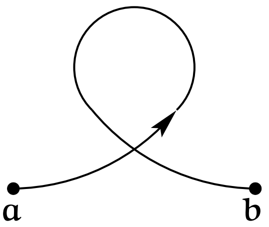


Figure 2.2: An oriented curve with start-vertex $\mathbf{a}$ and end-vertex $\mathbf{b}$ with self-intersection.

- A loop is represented by a cyclic doubly linked list of coedges.
- A coedge is represented by a pointer to an edge $\mathcal{E}$ plus a direction flag. The direction flag indicates whether the coedge is equally or opposite oriented as $\mathcal{E}$.

We use the terms start-vertex and end-vertex in the context of coedges with respect to the direction flag. We demand that the loops are oriented, i.e. the end-vertex of a coedge in a loop must be the start-vertex of the next coedge of that loop. In this way we can easily enumerate all edges and all vertices of a face in the order of their orientation.

Another improvement of our representation is a test that ensures that all loops of a face are located on the same connected component of the surface that contains the face. Regarding definition 2.5, an oriented loop on a surface $\mathcal{S}$ defines an interior region and an exterior region on $\mathcal{S}$. Let $\mathcal{L}_1, \ldots, \mathcal{L}_k$ be all loops of the face $\mathcal{F}$ and let the interior region on $\mathcal{S}$ defined by $\mathcal{L}_i$ be denoted by $\mathcal{I}_i$. Then $\mathcal{F}$ is defined by

$$\bigcap_{i=1}^{k} (\mathcal{L}_i \cup \mathcal{I}_i).$$

It holds that $\mathcal{F}$ is connected and $\partial\mathcal{F}$ is the union of all its loops if and only if $\mathcal{L}_i \subset \mathcal{I}_j$ for all $i \neq j$. This is a similar statement as that of lemma 2.7 below and its proof is similar to the proof of that lemma, as well. So, another improvement of our representation is a test that checks this property for all pairs of loops of a face.

In order to ensure that a face $\mathcal{F}$ is bounded we need one more test. If the connected component $\tilde{\mathcal{S}}$ of the surface that contains $\mathcal{F}$ is bounded, then nothing has to be checked. Otherwise, let the loops of $\mathcal{F}$ be enclosed by a cube that is given by planes $\mathcal{P}_1, \ldots, \mathcal{P}_6$. Then each connected component of the intersection curve between one of the $\mathcal{P}_i$ and $\tilde{\mathcal{S}}$ lies completely in the interior or completely in the exterior of $\mathcal{F}$. We compute one point on each such connected component. If one of these points lies in the interior of $\mathcal{F}$, then $\mathcal{F}$ is unbounded, otherwise it is bounded.

Next, we require each edge to be adjacent to exactly two faces. More precisely, this means that each edge belongs to exactly two coedges, each coedge to exactly one loop and each loop to exactly one face. In this way we ensure that shells always represent compact 2-manifolds without boundary. This avoids the situation shown in the leftmost and in the middle picture of figure 2.1.

We require the faces of a shell to be pairwise coherently oriented. This ensures that each shell defines an interior and an exterior region in $\mathbb{R}^3$. Let $\mathcal{S}_1, \dots, \mathcal{S}_k$ be all shells and let the interior region defined by $\mathcal{S}_i$ be denoted by $\mathcal{I}_i$. Then, the object $\mathcal{O}$ is defined by

$$\bigcap_{i=1}^{k} (\mathcal{S}_i \cup \mathcal{I}_i).$$

Lemma 2.7. *$\mathcal{O}$ is connected and $\partial\mathcal{O} = \bigcup_{i=1}^{k} \mathcal{S}_i \quad \Leftrightarrow \quad \mathcal{S}_i \subset \mathcal{I}_j$ for all $i \neq j$.*

Proof. Let $\mathcal{O}$ be connected and $i \neq j$. Let furthermore $\mathbf{p}$ be a point on $\mathcal{S}_i$ and $\mathbf{q}$ be a point on $\mathcal{S}_j$. Every neighbourhood of $\mathbf{q}$ contains a point $\mathbf{r} \in \mathcal{O} \cap \mathcal{I}_j$. As $\mathcal{O}$ is connected, there is a path from $\mathbf{r}$ to $\mathbf{p}$ within $\mathcal{O}$. Therefore, $\mathbf{p} \in \mathcal{I}_j$.

Conversely, let $\mathcal{S}_i \subset \mathcal{I}_j$ for all $i \neq j$. We first prove that $\partial\mathcal{O}$ is the union of all shells. First, we notice that $\mathcal{O}$ can be expressed as

$$\bigcup_{i=1}^{k} \mathcal{S}_i \ \cup \ \bigcap_{i=1}^{k} \mathcal{I}_i.$$

Let $\mathbf{p} \in \partial\mathcal{O}$. If $\mathbf{p}$ is in none of the $\mathcal{S}_i$, then it must be in $\bigcap_{i=1}^{k} \mathcal{I}_i$, which is an open subset of $\mathcal{O}$. But then $\mathbf{p} \notin \partial\mathcal{O}$ and we have a contradiction. If on the other hand $\mathbf{p} \in \mathcal{S}_i$, then every neighbourhood of $\mathbf{p}$ contains a point that is not contained in $\mathcal{S}_i \cup \mathcal{I}_i \supset \mathcal{O}$. Therefore $\mathbf{p} \in \partial\mathcal{O}$. To prove that $\mathcal{O}$ is connected, we pick two points $\mathbf{p}, \mathbf{q} \in \mathcal{O}$. We construct a path $\mathcal{P}$ from $\mathbf{p}$ to $\mathbf{q}$ in $\mathcal{O}$ as follows. We follow the straight line from $\mathbf{p}$ to $\mathbf{q}$ until we reach the first point $\mathbf{r}_1$ where this line leaves a set $\mathcal{S}_i \cup \mathcal{I}_i$. Let $\mathbf{r}_2$ be the last point where the line re-enters this set. We follow a path from $\mathbf{r}_1$ to $\mathbf{r}_2$ on $\mathcal{S}_i$. Such a path exists because $\mathcal{S}_i$ is connected. In this way we proceed until we finally reach $\mathbf{q}$. □

According to this lemma we add a test that checks this property for all pairs of shells.

What is missing is a test whether $\mathcal{O}$ is bounded or not.

Lemma 2.8. *Let $\mathcal{O}$ be a closed 3-manifold with boundary and let $\partial\mathcal{O}$ be bounded and of bounded variation. Let $\mathcal{B}$ be a ball that encloses $\partial\mathcal{O}$ and let $\mathbf{p} \notin \mathcal{B}$ be a point.*

$$\mathcal{O} \textit{ is bounded } \Leftrightarrow \mathbf{p} \notin \mathcal{O}.$$

Proof. We prove "$\mathcal{O}$ bounded $\Rightarrow \mathbf{p} \notin \mathcal{O}$" and "$\mathcal{O}$ unbounded $\Rightarrow \mathbf{p} \in \mathcal{O}$" at the same time. Let $\mathcal{O}$ be bounded (unbounded) and suppose that $\mathbf{p} \in \mathcal{O}$ ($\mathbf{p} \notin \mathcal{O}$). Then there is a point $\mathbf{q} \notin \mathcal{B}$ with $\mathbf{q} \notin \mathcal{O}$ ($\mathbf{q} \in \mathcal{O}$) and a path $\mathcal{P}$ from $\mathbf{p}$ to $\mathbf{q}$ with $\mathcal{P} \cap \mathcal{B} = \emptyset$, and thus $\mathcal{P} \cap \partial\mathcal{O} = \emptyset$. So, $\mathcal{P}$ connects a point inside $\mathcal{O}$ with a point outside $\mathcal{O}$ without crossing the boundary. This is a contradiction. □

Based on this lemma, we add a test to our representation to check whether $\mathcal{O}$ is bounded. This test looks as follows. Pick any point $\mathbf{p} \in \mathbb{R}^3$ and consider a ray from $\mathbf{p}$ to infinity that intersects the boundary of $\mathcal{O}$. Compute the points of intersection between this ray and $\partial\mathcal{O}$. Let $\mathbf{q}$ be the last such point (with respect to the direction of the ray). Depending on the sign of the dotproduct between the direction of the ray and the normal of $\partial\mathcal{O}$ in $\mathbf{q}$ we can decide whether the ray enters or leaves $\mathcal{O}$ at that point. Lemma 2.8 tells us that $\mathcal{O}$ is bounded if and only if the ray leaves $\mathcal{O}$ at $\mathbf{q}$.

Finally, we notice that sometimes it is necessary to answer questions like *"Do these two vertices belong to the same loop/shell/face?"* This means we should be able to "navigate" through the topologic structure of an object efficiently. To this end, we add backward pointers to the representation.

- A vertex points to one of the edges that has it as start- or end-vertex.
- An edge has a pointer to one of its two coedges.
- A coedge has a pointer to its *adjacent* coedge, i.e. the other coedge that belongs to the same edge. Moreover, it has a pointer to the loop it belongs to.
- A loop points to its face.
- Finally, a face has a pointer to its shell.

We close this section with a brief description of how to represent the geometric components curve and surface. Curves are represented by a parameterization. In section 2.3 we will present parameterizations of conics and of general quadric intersection curves. The representation of a conic also contains the plane in which the conic lies along with the implicit form of the conic in that plane. In the case of general quadric intersection curves the representation also contains the two quadrics that intersect in that curve. If an edge $\mathcal{E}$ is represented by a curve with parameterization $\mathbf{c}(t)$ and by vertices $\mathbf{a}$ and $\mathbf{b}$, then we always assume that we know a parameter interval $[u, v]$ such that $\mathbf{a} = \mathbf{c}(u), \mathbf{b} = \mathbf{c}(v)$ and $\mathcal{E} = \{\, \mathbf{c}(t) \,|\, t \in [u, v] \,\}$.

An algebraic surface is represented by its implicit form. In section 2.3 we will present implicit forms of the algebraic surfaces used in this thesis, namely the quadrics and the torus.

2.2.3 Definition of Special Object Classes

In this subsection we will define the classes of objects that will be considered in the following chapters. We will distinguish these classes by the surface and curve types that their boundaries will consist of. These surfaces and curves will be described in more detail in 2.3

Quadratic Complexes (QC) The boundaries of quadratic complexes (QC) consist of patches of quadrics as faces and segments of conics as edges. Note, that we consider planes as special cases of quadrics and straight lines as special cases of conics. This is the natural extension of polyhedra, since the algebraic degree of the boundary is increased by one.

Natural Quadratic Complexes plus Torus (NQC+T) An object is a NQC+T if its faces are embedded on natural quadrics and the torus and its edges are straight line segments and circle segments. Natural quadrics are planes, spheres, circular cones and circular cylinders.

Quadratic Complexes plus QIC (QC+QIC) This class is the boolean closure of the class of QC. The boundary of a QC+QIC consists of faces that are embedded on quadrics and of edges that are embedded on quadric intersection curves (QIC). Conics are a special case of QIC.

2.3 Curves and Surfaces

In this section we will describe the curves and surfaces that the boundaries of the objects in this thesis consist of. The surfaces that we will introduce are the quadrics and the torus. The curves that we will deal with are the conics and quadric intersection curves.

2.3.1 Quadrics

A quadric is a surface whose implicit form is a quadratic form, i.e. an equation of the form

$$\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} + 2\mathbf{x}^{\mathsf{T}}\mathbf{a} + a_0 = 0, \tag{2.1}$$

where $\mathbf{A} \in \mathbb{R}^{3\times 3}$ is symmetric and $\mathbf{a} \in \mathbb{R}^3$. Often, we write this equation in homogeneous coordinates as

$$\begin{aligned} \mathbf{x}^{\mathsf{T}}\mathbf{A}_{\mathsf{H}}\mathbf{x} &= 0, \text{ where} \\ \mathbf{A}_{\mathsf{H}} &= \begin{bmatrix} \mathbf{A} & \mathbf{a} \\ \mathbf{a}^{\mathsf{T}} & a_0 \end{bmatrix}, \\ \mathbf{A} &\in \mathbb{R}^{3\times 3} \text{ symmetric}, \\ \mathbf{a} &\in \mathbb{R}^3. \end{aligned} \tag{2.2}$$

We will always use the index H to indicate homogeneous matrices. If we have a homogeneous matrix $\mathbf{M}_{\mathsf{H}}$, then $\mathbf{M}$ always denotes the upper-left (3×3)-submatrix.

The coordinate system can be rotated and translated in such a way that $\mathbf{A}$ is a diagonal matrix and $\mathbf{A}\mathbf{a} = \mathbf{0}$. This is done by first applying a principle axis transformation to

Normal form $(a > 0, b > 0, c > 0)$	Name of the quadric
$ax_1^2 + bx_2^2 + cx_3^2 = 1$	Ellipsoid
$ax_1^2 + bx_2^2 - cx_3^2 = 1$	One-sheet hyperboloid
$ax_1^2 + bx_2^2 - cx_3^2 = 0$	Double cone
$-ax_1^2 - bx_2^2 + cx_3^2 = 1$	Two-sheet hyperboloid
$ax_1^2 + bx_2^2 - 2cx_3 = 0$	Elliptic paraboloid
$ax_1^2 - bx_2^2 - 2cx_3 = 0$	Hyperbolic paraboloid
$ax_1^2 + bx_2^2 = 1$	Elliptic cylinder
$ax_1^2 - bx_2^2 = 1$	Hyperbolic cylinder
$ax_1^2 - bx_2^2 = 0$	Two intersecting planes
$bx_2^2 - 2x_1 = 0$	Parabolic cylinder
$ax_1^2 = 1$	Two parallel planes
$x_1^2 = 0$	Double plane

Table 2.1: Normal forms of quadrics

$\mathbf{A}$, which yields the rotation matrix. In this way, $\mathbf{A}$ is replaced by a diagonal matrix $\tilde{\mathbf{A}} = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3)$. Then we translate the coordinate system by replacing $\mathbf{x}$ by $\mathbf{x} + \mathbf{c}$, where $\mathbf{c}$ is defined as

$$c_i = \begin{cases} -\frac{a_i}{\lambda_i} & \text{if } \lambda_i \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

This does not change $\tilde{\mathbf{A}}$ but the vector $\mathbf{a}$ is replaced by $\tilde{\mathbf{a}} = \tilde{\mathbf{A}}\mathbf{c} + \mathbf{a}$. It is easy to verify that $\tilde{\mathbf{A}}\tilde{\mathbf{a}} = \mathbf{0}$. In the case of *central surfaces*, i.e. $\det \mathbf{A} \neq 0$, we obtain $\tilde{\mathbf{a}} = \mathbf{0}$. If $a_0 \neq 0$ we multiply the quadratic form by $-1/a_0$. In this way we can assume that a_0 is either zero or minus one. By further rotations around the coordinate axes, if necessary, we obtain the *normal forms* shown in table 2.1. During our computations we will always keep a transformation matrix $\mathbf{T}_H \in \mathbb{R}^{4\times4}$ such that $\mathbf{T}_H^T \mathbf{A}_H \mathbf{T}_H$ is in normal form. The implicit forms of the intersecting planes, the parallel planes and the double plane are only given for the sake of completeness. We will always use the linear implicit form

$$\mathbf{n}^T\mathbf{x} - n_0 = 0$$

in order to define a plane, where $\mathbf{n}$ is the normal vector. We usually assume that $|\mathbf{n}| = 1$.

Parameterization $\mathbf{x}(u,v)$ $(a > 0, b > 0, c > 0)$	Name of the quadric
$\left[\frac{1}{\sqrt{a}}\cos u\cos v,\ \frac{1}{\sqrt{b}}\cos u\sin v,\ \frac{1}{\sqrt{c}}\sin u\right]^T$	Ellipsoid
$\left[\frac{1}{\sqrt{a}}(\cos u + v\sin u),\ \frac{1}{\sqrt{b}}(\sin u - v\cos u),\ \frac{1}{\sqrt{c}}v\right]^T$	One-sheet hyperboloid
$\left[\frac{1}{\sqrt{a}}v\cos u,\ \frac{1}{\sqrt{b}}v\sin u,\ \frac{1}{\sqrt{c}}v\right]^T$	Double cone
$\left[\frac{1}{\sqrt{a}}\sinh u\cos v,\ \frac{1}{\sqrt{b}}\sinh u\sin v,\ \pm\frac{1}{\sqrt{c}}\cosh u\right]^T$	Two-sheet-hyperboloid
$\left[\frac{1}{\sqrt{a}}v\cos u,\ \frac{1}{\sqrt{b}}v\sin u,\ \frac{1}{2c}v^2\right]^T$	Elliptic paraboloid
$\left[\frac{1}{\sqrt{a}}(u+v),\ \frac{1}{\sqrt{b}}(u-v),\ \frac{2}{c}uv\right]^T$	Hyperbolic paraboloid
$\left[\frac{1}{\sqrt{a}}\cos u,\ \frac{1}{\sqrt{b}}\sin u, v\right]^T$	Elliptic cylinder
$\left[\pm\frac{1}{\sqrt{a}}\cosh u,\ \frac{1}{\sqrt{b}}\sinh u, v\right]^T$	Hyperbolic cylinder
$\left[\frac{1}{2}u^2, \frac{1}{\sqrt{b}}u, v\right]^T$	Parabolic cylinder

Table 2.2: Parameterizations of non-planar quadrics

In the following, a quadric $\mathcal{Q}$ will often be called a *natural quadric* if it is a plane (as a special case), a sphere, a circular cone or a circular cylinder.

We will also need parameterizations of quadrics. Table 2.2 shows parametric forms for the non-planar quadrics, where the coefficients a, b and c correspond to those in table 2.1. Except for the two-sheet hyperboloid and the hyperbolic cylinder, these parameterizations are one-to-one and onto if one restricts the parameter range as follows.

ellipsoid	$u \in [-\frac{\pi}{2}, \frac{\pi}{2}), v \in [-\pi, \pi)$
1-sheet hyperboloid, cone, elliptic paraboloid, elliptic cylinder	$u \in [-\pi, \pi), v \in \mathbb{R}$
2-sheet hyperboloid	$u \in \mathbb{R}, v \in [-\pi, \pi)$
hyperbolic paraboloid, hyperbolic cylinder, parabolic cylinder	$u, v \in \mathbb{R}$

If we choose only the plus sign or only the minus sign in the parameterization of the two-sheet hyperboloid and the hyperbolic cylinder, then with these restrictions, the parameterization is one-to-one and onto for one connected component of the surface.

2.3.2 Torus

The implicit form of a torus is a polynomial of degree four. Let r and R be the minor and major radii of the torus, respectively, and let the unit vector $\mathbf{n}$ be the normal of its main plane. Furthermore, let $\mathbf{c}$ be the center of the main circle. Then the torus is defined by the implicit form

$$\left((\mathbf{x}-\mathbf{c})^2 + R^2 - r^2\right)^2 - 4R^2\left((\mathbf{x}-\mathbf{c}) \times \mathbf{n}\right)^2 = 0. \tag{2.3}$$

We do not consider tori with self-intersections. Therefore, we always assume $r < R$. For any point $\mathbf{p}$ on a torus $\mathcal{T}$ there are four circles passing through $\mathbf{p}$. These are the cross-sectional circle $CSC(\mathcal{T}, \mathbf{p})$, the profile circle $PFC(\mathcal{T}, \mathbf{p})$ and the two Villarceau circles. Cross-sectional circles, profile circles and Villarceau circles are the only conics that can be embedded on a torus. Let $\mathbf{p}_0 = \mathbf{p} - \mathbf{n}^T(\mathbf{p}-\mathbf{c}) \cdot \mathbf{n}$ be the projection of $\mathbf{p}$ onto the main plane of $\mathcal{T}$. Then the cross-sectional circle through $\mathbf{p}$ is defined by the normal $(\mathbf{p}-\mathbf{c}) \times \mathbf{n}$ of the plane containing it, its center $R(\mathbf{p}_0 - \mathbf{c})/\sqrt{(\mathbf{p}_0-\mathbf{c})^2}$ and its radius r. The normal of the plane containing the profile circle through $\mathbf{p}$ is $\mathbf{n}$, its center is $\mathbf{c} + \mathbf{n}^T(\mathbf{p}-\mathbf{c}) \cdot \mathbf{n}$ and the radius is given by $\sqrt{(\mathbf{p}_0-\mathbf{c})^2}$. Villarceau circles on $\mathcal{T}$ are obtained by intersecting $\mathcal{T}$ with a plane $\mathcal{P}$ through $\mathbf{c}$ with unit normal vector $\mathbf{n}_\mathcal{P}$, where $\mathbf{n}_\mathcal{P}$ is chosen in such a way that $\mathbf{n}^T\mathbf{n}_\mathcal{P} = \sqrt{R^2 - r^2}/R$. Each such intersection consists of two Villarceau circles. We do not describe how to find the two Villarceau circles through a given point $\mathbf{p}$ since we will not use these circles in this thesis, although they may occur as edges of faces on tori.

As in the case of quadrics, we give a parameterization of the torus. Therefore, we assume that the main plane is the (x_1, x_2)-plane and the center is the origin. It is easy to verify that the torus can be parameterized by

$$\mathbf{x}(u,v) = \begin{bmatrix} (R + r\cos u)\cos v \\ (R + r\cos u)\sin v \\ r\sin u \end{bmatrix}. \tag{2.4}$$

If we restrict the parameter range by $u, v \in [-\pi, \pi)$, then this mapping is one-to-one and onto.

2.3.3 Conics

A conic is an intersection curve between a quadric and a plane. Let a quadric in general position and orientation be defined by equation (2.1). We rewrite this equation in the form

$$\begin{bmatrix} \tilde{x} \\ x_3 \end{bmatrix}^T \begin{bmatrix} \tilde{\mathbf{A}} & \tilde{b} \\ \tilde{b}^T & b_0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ x_3 \end{bmatrix} + 2 \begin{bmatrix} \tilde{x} \\ x_3 \end{bmatrix}^T \begin{bmatrix} \tilde{a} \\ c \end{bmatrix} + a_0,$$

where $\tilde{\mathbf{A}} \in \mathbb{R}^{2\times 2}$ and all vectors with tildes are two-dimensional. If we intersect this quadric with the plane defined by $x_3 = 0$, then we obtain the following quadratic form in the (x_1, x_2)-plane (we omit the tildes):

$$\mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{x}^T\mathbf{a} + a_0 = 0, \tag{2.5}$$

Normal form $(a > 0, b > 0)$	Name of the conic
$\frac{x_1^2}{a^2} + \frac{x_2^2}{b^2} = 1$	Ellipse
$\frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} = 1$	Hyperbola
$ax_1^2 - bx_2^2 = 0$	Two intersecting lines
$ax_1^2 - x_2 = 0$	Parabola
$x_1^2 = 0$	Double line
$ax_1^2 = 1$	Two parallel lines

Table 2.3: Normal forms of conics

where $\mathbf{x}, \mathbf{a} \in \mathbb{R}^2$ and $\mathbf{A} \in \mathbb{R}^{2\times 2}$ symmetric. As in the case of quadrics, we can rotate and translate the coordinate system and multiply the quadratic form with a scalar such that the matrix $\mathbf{A}$ is diagonal, $\mathbf{A}\mathbf{a} = \mathbf{0}$ and $a_0 = -1$. In this way we obtain the normal forms shown in table 2.3. We mentioned the intersecting, parallel and double lines only for the sake of completeness. We will always consider lines that can be implicitly defined by a linear equation of the form $\mathbf{n}^T\mathbf{x} - n_0 = 0$, where $\mathbf{n}$ is perpendicular to the line in the (x_1, x_2)-plane. Obviously, we can parameterize the ellipse, the hyperbola and the parabola in the (x_1, x_2)-plane as follows.

$$\begin{array}{rcrcl} \text{ellipse} & : & \mathbf{x}(\varphi) & = & [a\cos(\varphi+\Delta), b\sin(\varphi+\Delta), 0]^T \\ \text{hyperbola} & : & \mathbf{x}(\lambda) & = & [\pm a\cosh\lambda, b\sinh\lambda, 0]^T \\ \text{parabola} & : & \mathbf{x}(\lambda) & = & [\lambda, a\lambda^2, 0]^T. \end{array} \tag{2.6}$$

The reason why we introduced the pase shift Δ in the parameterization of the ellipse will become clear below, when we show how to find a rational parameterization. The straight line can be parameterized as $\mathbf{x}(\lambda) = [\lambda, 0, 0]^T$. From a parameterization $\mathbf{x}(\lambda) = [x_1(\lambda), x_2(\lambda), 0]^T$ of a curve in the (x_1, x_2)-plane we can obtain a parameterization $\mathbf{x}(\lambda) = \mathbf{c} + x_1(\lambda)\mathbf{u} + x_2(\lambda)\mathbf{v}$ in 3-space by choosing vectors $\mathbf{c}, \mathbf{u}$ and $\mathbf{v}$ with $|\mathbf{u}| = |\mathbf{v}| = 1$ and $\mathbf{u}^T\mathbf{v} = 0$. In order to represent a conic, we store the vectors $\mathbf{c}, \mathbf{u}$ and $\mathbf{v}$ as well as the functions x_1 and x_2. We also store the implicit form (2.5) of the conic in the plane defined by $(\mathbf{u} \times \mathbf{v})^T(\mathbf{x} - \mathbf{c}) = 0$.

By substituting $\varphi = 2\arctan t$ we can parameterize the cos and sin functions rationally. We have

$$\cos\varphi = \frac{1-t^2}{1+t^2}, \quad \text{and} \quad \sin\varphi = \frac{2t}{1+t^2}. \tag{2.7}$$

Using this and the identities

$$\begin{array}{rcl} \cos(\varphi+\Delta) & = & \cos\varphi\cos\Delta - \sin\varphi\sin\Delta \quad \text{and} \\ \sin(\varphi+\Delta) & = & \sin\varphi\cos\Delta + \cos\varphi\sin\Delta \end{array}$$

and by substituting $\gamma = \cos\Delta$ and $\delta = \sin\Delta$, the parameterization of the ellipse in (2.6) becomes

$$\mathbf{x}(t) = \left[a\frac{\gamma(1-t^2)-2\delta t}{1+t^2},\ b\frac{\delta(1-t^2)+2\gamma t}{1+t^2},\ 0\right]^T. \tag{2.8}$$

Hence, we can assume that ellipses are always given by rational parameterizations. Now we can see why we introduced the phase shift Δ. If an ellipse $\mathcal{C}$ is parameterized rationally by (2.8), then there is always a point $\mathbf{p} \in \mathcal{C}$ for which there is no value for t such that $\mathbf{p} = \mathbf{x}(t)$. This point has the coordinates $\mathbf{p} = \lim_{t\to\pm\infty} \mathbf{x}(t) = [-a\gamma, -b\delta, 0]^T$ and corresponds to the angle $\varphi = \pi$ in (2.6). If an edge $\mathcal{E}$ that is embedded on $\mathcal{C}$ is not the whole ellipse, then we can choose Δ in such a way that $\mathbf{p} \notin \mathcal{E}$.

By substituting $\lambda = 2 \operatorname{arctanh} t$ we can also parameterize the cosh and sinh functions rationally. We obtain

$$\cosh\lambda = \frac{1+t^2}{1-t^2}, \quad \text{and} \quad \sinh\lambda = \frac{2t}{1-t^2} \tag{2.9}$$

for $-1 < t < 1$. Hence, we can assume that hyperbolas are also given by rational parameterizations.

2.3.4 Quadric Intersection Curves (QIC)

Now we want to derive a parameterization for the intersection curve between two quadrics $\mathcal{A}$ and $\mathcal{B}$ defined by the quadratic forms $\mathbf{x}^T\mathbf{A}_H\mathbf{x} = 0$ and $\mathbf{x}^T\mathbf{B}_H\mathbf{x} = 0$. We will use the notation introduced in (2.2).

Definition 2.9. *We call a quadric $\mathcal{A}$ defined by the homogeneous matrix $\mathbf{A}_H$ an L-quadric, if one of the eigenvalues of $\mathbf{A}$ is zero and the product of the remaining two eigenvalues is ≤ 0.*

Remark: We use the term L-quadric because the following theorem is due to J. Levin (see [Lev76]). From table 2.1 we see that the L-quadrics are the following quadrics: double, parallel and intersecting planes, hyperbolic and parabolic cylinder, hyperbolic paraboloid.

A nice property of L-quadrics is that they are *ruled*, i.e. they have a parameterization $\mathbf{x}(t, \lambda)$ that is linear in the parameter λ. Hence the intersection curve between an L-quadric and an arbitrary quadric can be parameterized as follows. We insert the parameterization of the L-quadric into the quadratic form of the other quadric. This leads to a polynomial $f(t, \lambda)$ that is quadratic in λ. We can solve $f(t, \lambda) = 0$ for λ and insert the resulting expression into the parameterization of the L-quadric. The resulting function $\mathbf{x}(t)$ is a parameterization for the intersection curve.

Theorem 2.10. *The intersection curve between two arbitrary quadrics lies on an L-quadric.*

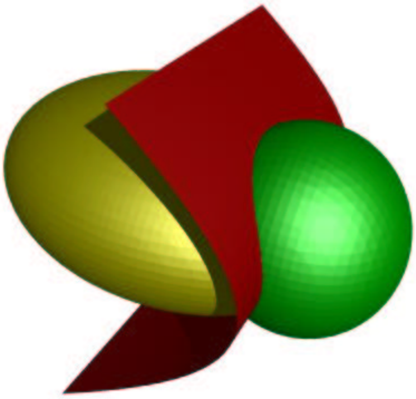


Figure 2.3: L-quadric that intersects an ellipsoid and a sphere in the same curve

So, if we are interested in the intersection curve between $\mathcal{A}$ and $\mathcal{B}$ this theorem ensures that there is an L-quadric $\mathcal{C}$ that contains this curve. If we knew $\mathcal{C}$ we could replace $\mathcal{A}$ or $\mathcal{B}$ by $\mathcal{C}$ and then we could use the above stated approach to determine a parameterization of the intersection curve. Figure 2.3 shows a sphere intersecting an ellipsoid, and a hyperbolic cylinder that intersects both of them in the same curve. In the following, we will prove theorem 2.10 and then describe how to effectively determine a parameterization of the intersection curve between $\mathcal{A}$ and $\mathcal{B}$. Alternatively, we could use the results presented in [DLLP03]. The approach presented there is in some sense similar to the one described here, but they make some improvements that reduce the number of nested radicals in the coefficients of the parameterization to near optimality. This is particularly important if one wants to use exact arithmetic.

Before we prove theorem 2.10 we need to make some observations and state some lemmas.

First, we give another characterization of L-quadrics. Therefore, we make the following observation. The characteristic polynomial of a matrix $\mathbf{M} \in \mathbb{R}^{3\times 3}$ is given by

$$\chi_{\mathbf{M}}(\mu) = -\mu^3 + \mu^2 \cdot \operatorname{tr}(\mathbf{M}) - \mu \cdot \operatorname{tr}(\operatorname{adj}\mathbf{M}) + \det \mathbf{M}. \tag{2.10}$$

Lemma 2.11. *A quadric $\mathcal{C}$ defined by the homogeneous matrix $\mathbf{C}_{\mathrm{H}}$ is an L-quadric iff* $\det \mathrm{C} = 0$ *and* $\operatorname{tr}(\operatorname{adj}\mathbf{C}) \leq 0$.

Proof. Let μ_1, μ_2 and μ_3 be the eigenvalues of $\mathbf{C}$. Without loss of generality we assume that $\mu_1\mu_2 \leq 0$ and $\mu_3 = 0$. By equation (2.10) this means that $\det \mathbf{C} = 0$ and

$$\begin{aligned} \chi_{\mathbf{C}}(\mu) &= -\mu(\mu^2 - \mu \cdot \operatorname{tr}(\mathbf{C}) + \operatorname{tr}(\operatorname{adj}\mathbf{C})) \\ &= -\mu\big(\mu^2 - (\mu_1 + \mu_2)\mu + \mu_1\mu_2\big) \end{aligned}$$

and thus $\operatorname{tr}(\operatorname{adj}\mathbf{C}) = \mu_1\mu_2$. The other direction works analogously. □

We make one more observation. Let $\mathbf{C}_\mathsf{H}$ define a quadric and let $\mathbf{T}_\mathsf{H} \in \mathbb{R}^{4\times 4}$ be a regular matrix. Then, by Sylvester's Law of Inertia,

$$\mathbf{C}_\mathsf{H} \text{ defines an L-quadric} \quad \Leftrightarrow \quad \mathbf{C}'_\mathsf{H} = \mathbf{T}_\mathsf{H}^\mathsf{T}\mathbf{C}_\mathsf{H}\mathbf{T}_\mathsf{H} \text{ defines an L-quadric.}$$

In particular this means that if $\det \mathsf{C} = 0$, then $\operatorname{sign}(\operatorname{tr}(\operatorname{adj}\mathbf{C}')) = \operatorname{sign}(\operatorname{tr}(\operatorname{adj}\mathbf{C}))$.

Definition 2.12. *Let two quadrics $\mathcal{A}$ and $\mathcal{B}$ be defined by the homogeneous matrices $\mathbf{A}_\mathsf{H}$ and $\mathbf{B}_\mathsf{H}$. We call the set of quadrics $\mathcal{Q}_{\mathbf{A},\mathbf{B}}$ defined by the matrices*

$$\mathbf{Q}_{\mathbf{A},\mathbf{B},\mathsf{H}}(\lambda) = \mathbf{A}_\mathsf{H} + \lambda\mathbf{B}_\mathsf{H}, \quad \lambda \in \mathbb{R}$$

the pencil generated by $\mathcal{A}$ and $\mathcal{B}$, or briefly the pencil of $\mathcal{A}$ and $\mathcal{B}$.

Clearly, all quadrics in the pencil of $\mathcal{A}$ and $\mathcal{B}$ intersect in the same curve. The following lemma shows what the determinant of $\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda)$ looks like.

Lemma 2.13. *Let $\mathbf{A},\mathbf{B} \in \mathbb{R}^{3\times 3}$ be arbitrary matrices. Then*

$$\det(\mathbf{A} + \lambda\mathbf{B}) = \lambda^3 \cdot \det\mathbf{B} + \lambda^2 \cdot \operatorname{tr}(\mathbf{A}\cdot\operatorname{adj}\mathbf{B}) + \lambda\cdot\operatorname{tr}(\mathbf{B}\cdot\operatorname{adj}\mathbf{A}) + \det\mathbf{A}.$$

Proof.

$$\begin{aligned}
\det(\mathbf{A}+\lambda\mathbf{B}) &= \det\left((-\mathbf{A}\mathbf{B}^{-1} - \lambda\mathbf{E})\cdot(-\mathbf{B})\right)\\
&= -\det(-\mathbf{A}\mathbf{B}^{-1} - \lambda\mathbf{E})\cdot\det\mathbf{B}\\
&= -\chi_{_{-\mathbf{A}\mathbf{B}^{-1}}}(\lambda)\cdot\det\mathbf{B}\\
&= \left(\lambda^3 + \lambda^2\cdot\operatorname{tr}\left(\mathbf{A}\mathbf{B}^{-1}\right) + \lambda\cdot\operatorname{tr}\left(\operatorname{adj}(\mathbf{A}\mathbf{B}^{-1})\right) + \det(\mathbf{A}\mathbf{B}^{-1})\right)\cdot\det\mathbf{B}\\
&= \lambda^3\cdot\det\mathbf{B} + \lambda^2\cdot\operatorname{tr}(\mathbf{A}\cdot\operatorname{adj}\mathbf{B}) + \lambda\cdot\operatorname{tr}(\mathbf{B}\cdot\operatorname{adj}\mathbf{A}) + \det\mathbf{A}.
\end{aligned}$$

This computation requires that $\mathbf{B}$ is regular. To overcome this, we assume that

$$\mathbf{A} = (A_{i,j})_{i,j=1}^3, \quad \mathbf{B} = (B_{i,j})_{i,j=1}^3 \quad \in \mathbb{R}(A_{1,1},\dots,A_{3,3},B_{1,1},\dots,B_{3,3})^{3\times 3}$$

are symbolic matrices. Over this field, $\mathbf{B}$ is a regular matrix. At the end of the computation we replace the variables by real values. As this is a homomorphism into $\mathbb{R}$, and as both sides of the final equation can be computed using only addition and multiplication, this equation still holds after replacing the variables. □

Proof of theorem 2.10. If $\mathcal{A}$ or $\mathcal{B}$ already is an L-quadric, then there is nothing to show. So, we assume that neither $\mathcal{A}$ nor $\mathcal{B}$ is an L-quadric. We show that $\mathcal{Q}_{\mathbf{A},\mathbf{B}}$ contains an L-quadric. First we consider the case that $\det(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda))$ is constantly zero. Let $\mathbf{T}_1$ be a rotation matrix such that $\mathbf{T}_1^\mathsf{T}\mathbf{B}\mathbf{T}_1 = \operatorname{diag}(\mu_1,\mu_2,0)$, and set

$$\mathbf{T}_2 = \operatorname{diag}\left(\frac{1}{\sqrt{|\mu_1|}}, \frac{1}{\sqrt{|\mu_2|}}, 1\right).$$

We define $\mathbf{T} = \mathbf{T}_1\mathbf{T}_2$ and set $\tilde{\mathbf{B}} = \sigma\cdot\mathbf{T}^\mathsf{T}\mathbf{B}\mathbf{T}$ and $\tilde{\mathbf{A}} = \sigma\cdot\mathbf{T}^\mathsf{T}\mathbf{A}\mathbf{T}$ with $\sigma = \operatorname{sign}(\mu_1)$. Then, $\tilde{\mathbf{B}} = \operatorname{diag}(1,1,0)$ and $f(\lambda) := \operatorname{tr}(\operatorname{adj}(\mathbf{Q}_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}}(\lambda)))$ has the same sign as $\operatorname{tr}(\operatorname{adj}(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda)))$. Writing $f(\lambda)$ in components of $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ we get

$$f(\lambda) = \left(\tilde{A}_{33}(\tilde{A}_{22}+\lambda) - \tilde{A}_{23}^2\right) + \left(\tilde{A}_{33}(\tilde{A}_{11}+\lambda) - \tilde{A}_{13}^2\right) + \left((\tilde{A}_{11}+\lambda)(\tilde{A}_{22}+\lambda) - \tilde{A}_{12}^2\right),$$

where the addends in parentheses are the diagonal elements of $\operatorname{adj}(\mathbf{Q}_{\tilde{\mathbf{A}},\tilde{\mathbf{B}}}(\lambda))$. From this we compute the discriminant of $f(\lambda)$ as

$$\frac{1}{4}(\tilde{A}_{11} - \tilde{A}_{22})^2 + \tilde{A}_{12}^2 + \tilde{A}_{13}^2 + \tilde{A}_{23}^2 + \tilde{A}_{33}^2 \geq 0,$$

so that we can find a root λ_0 of f. By lemma 2.11 we know that $\mathbf{Q}_{\mathbf{A},\mathbf{B},H}(\lambda_0)$ is an L-quadric.

Now let us assume that $\det(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda))$ is not constantly zero. Then we can find a value $\tilde{\lambda} \neq 0$ such that $\det(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\tilde{\lambda})) \neq 0$. Let $\tilde{\mathbf{B}}_H = \mathbf{A}_H + \tilde{\lambda}\mathbf{B}_H$. If we now can find a value for λ such that $\mathbf{A}_H + \lambda\tilde{\mathbf{B}}_H$ defines an L-quadric, then $\lambda \neq -1$ because this would imply that $\mathbf{B}_H$ already defines an L-quadric. But in this case $\mathbf{A}_H + \lambda\tilde{\mathbf{B}}_H$ defines the same quadric as

$$\mathbf{Q}_{\mathbf{A},\mathbf{B},H}\left(\frac{\lambda\tilde{\lambda}}{1+\lambda}\right)$$

and we are done. Since $\det(\tilde{\mathbf{B}}) \neq 0$, the polynomial $\det(\mathbf{Q}_{\mathbf{A},\tilde{\mathbf{B}}}(\lambda_0))$ has degree three and thus has a real root λ_0. Let μ_1, μ_2 and 0 be the eigenvalues of $\mathbf{Q}_{\mathbf{A},\tilde{\mathbf{B}}}(\lambda_0)$. If $\mu_1\mu_2 \leq 0$, then we are done. Otherwise let $\mathbf{P}_{1,H} = \mathbf{Q}_{\mathbf{A},\tilde{\mathbf{B}},H}(\lambda_0)$ and let $\mathbf{T}_1$ be a regular matrix such that

$$\mathbf{P}_2 = \mathbf{T}_1^\mathsf{T}\mathbf{P}_1\mathbf{T}_1 = \operatorname{diag}(\mu_1, \mu_2, 0).$$

We set $\mathbf{T}_{1,H} = \operatorname{diag}(\mathbf{T}_1, 1)$ and $\mathbf{P}_{2,H} = \mathbf{T}_{1,H}^\mathsf{T}\mathbf{P}_{1,H}\mathbf{T}_{1,H}$. By setting $\sigma = \operatorname{sign}(\mu_1)$ and

$$\mathbf{P}_{3,H} = \sigma\cdot\mathbf{T}_{2,H}^\mathsf{T}\mathbf{P}_{2,H}\mathbf{T}_{2,H}, \text{ where } \mathbf{T}_{2,H} = \operatorname{diag}\left(\frac{1}{\sqrt{|\mu_1|}}, \frac{1}{\sqrt{|\mu_2|}}, 1, 1\right),$$

we achieve $\mathbf{P}_3 = \operatorname{diag}(1,1,0)$. Let $\mathbf{Q}_H = \mathbf{T}_{2,H}^\mathsf{T}\mathbf{T}_{1,H}^\mathsf{T}\tilde{\mathbf{B}}\mathbf{T}_{1,H}\mathbf{T}_{2,H}$. If $Q_{13} = 0$ we set $\mathbf{T}_{3,H} = \mathbf{E}$. Otherwise we define the transformation

$$\mathbf{T}_{3,H} = \operatorname{diag}\left(\frac{1}{\sqrt{Q_{13}^2 + Q_{23}^2}}\begin{bmatrix} Q_{23} & Q_{13} \\ -Q_{13} & Q_{23}\end{bmatrix}, 1, 1\right).$$

We set $\mathbf{P}_{4,H} = \mathbf{T}_{3,H}^\mathsf{T}\mathbf{P}_{3,H}\mathbf{T}_{3,H}$ and $\mathbf{R}_H = \mathbf{T}_{3,H}^\mathsf{T}\mathbf{Q}_H\mathbf{T}_{3,H}$. We see that $\mathbf{P}_4 = \mathbf{P}_3$ and verify that

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & 0 \\ R_{12} & R_{22} & R_{23} \\ 0 & R_{23} & R_{33}\end{bmatrix}.$$

We now distinguish the cases $R_{33} = 0$ and $R_{33} \neq 0$.

1. $R_{33} = 0$: We set $\mathbf{P}_{5,H} = \mathbf{R}_H - R_{11}\mathbf{P}_{4,H}$. Then we have

$$\mathbf{P}_5 = \begin{bmatrix} 0 & a & 0 \\ a & b & c \\ 0 & c & 0 \end{bmatrix}$$

and claim that $\mathbf{P}_{5,H}$ defines an L-quadric. But as $\det(\mathbf{P}_5) = 0$, this follows immediately from lemma 2.11 if one observes that $\mathrm{tr}\,(\mathrm{adj}\,\mathbf{P}_5) = -c^2 - a^2 \leq 0$. If we set $\mathbf{T}_H = \mathbf{T}_{1,H}\mathbf{T}_{2,H}\mathbf{T}_{3,H}$, we obtain

$$\begin{aligned} \mathbf{P}_{5,H} &= -\sigma R_{11}\mathbf{T}_H^T\mathbf{Q}_{\mathbf{A},\tilde{\mathbf{B}},H}(\lambda_1)\mathbf{T}_H, \text{ where} \\ \lambda_1 &= \lambda_0 - \frac{1}{\sigma R_{11}}. \end{aligned}$$

We know that $R_{11} \neq 0$ because otherwise the matrix $\mathbf{R}_H$ would already define an L-quadric, and thus $\tilde{\mathbf{B}}_H$ would define an L-quadric, as well.

2. $R_{33} \neq 0$: In this case we set $\mathbf{S}_H = \frac{1}{R_{33}}\mathbf{R}_H$. We determine a value $\alpha \in \mathbb{R}$ such that $\det(\mathbf{S} - \alpha\mathbf{P}_4) = 0$. This value is the solution of a quadratic equation:

$$\begin{aligned} \beta &= S_{22} - S_{23}^2, \\ \gamma &= \sqrt{(S_{11} - \beta)^2 + 4S_{12}^2}, \\ \alpha &= \frac{1}{2}(S_{11} + \beta \pm \gamma). \end{aligned} \tag{2.11}$$

Since the discriminant $(S_{11} - \beta)^2 + 4S_{12}^2 \geq 0$, such an α can always be found. We know that $\alpha \neq 0$ because otherwise the determinant of $\tilde{\mathbf{B}}$ would be zero, which is a contradiction to our choice of $\tilde{\mathbf{B}}$. Now we define $\mathbf{P}_{5,H} = \mathbf{S}_H - \alpha\mathbf{P}_{4,H}$. If we can show that this matrix defines an L-quadric, we have

$$\begin{aligned} \mathbf{P}_{5,H} &= -\alpha\sigma\mathbf{T}_H^T\mathbf{Q}_{\mathbf{A},\mathbf{B},H}(\lambda_1)\mathbf{T}_H, \text{ where} \\ \lambda_1 &= \lambda_0 - \frac{1}{\alpha\sigma R_{33}}, \end{aligned}$$

and we are done. In order to show that $\mathbf{P}_{5,H}$ defines an L-quadric, we use lemma 2.11.

$$\begin{aligned} \mathrm{tr}\,(\mathrm{adj}\,(\mathbf{P}_5)) &= S_{11} + S_{22} - S_{12}^2 - S_{23}^2 - 2\alpha + (S_{11} - \alpha)(S_{22} - \alpha) \\ &= S_{11} + S_{22} - S_{23}^2 + S_{23}^2 S_{11} - 2\alpha - S_{23}^2\alpha. \end{aligned}$$

The last equality holds because $\det(\mathbf{P}_5) = (S_{11}-\alpha)(S_{22}-\alpha) - S_{23}^2(S_{11}-\alpha) - S_{12}^2 = 0$. Using equation (2.11) we obtain

$$\mathrm{tr}\,(\mathrm{adj}\,(\mathbf{P}_5)) = \frac{1}{2}S_{23}^2(S_{23}^2 + S_{11} - S_{22}) \mp (1 + \frac{1}{2}S_{23}^2)\gamma.$$

By further transformations we show

$$\left((1+\frac{1}{2}S_{23}^2)\gamma\right)^2 - \left(\frac{1}{2}S_{23}^2(S_{23}^2+S_{11}-S_{22})\right)^2 = \gamma^2(1+S_{23}^2)+S_{12}^2S_{23}^4 \geq 0.$$

This ensures that tr (adj ($\mathbf{P}_5$)) ≤ 0 holds, provided that we choose the plus-sign in equation (2.11). □

In fact we have proven the following stronger version of theorem 2.10. If $f(\lambda) = \det(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda))$ is not constantly zero, then there is a real root λ_0 of f such that $\mathbf{Q}_{\mathbf{A},\mathbf{B},H}(\lambda_0)$ defines an L-quadric. If f is constantly zero, then for each real root λ_0 of $g(\lambda) = \mathrm{tr}\,(\mathrm{adj}\,(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda)))$ the matrix $\mathbf{Q}_{\mathbf{A},\mathbf{B},H}(\lambda_0)$ defines an L-quadric. This gives us a method to effectively compute an L-quadric containing the intersection curve between $\mathcal{A}$ and $\mathcal{B}$. First compute the at most cubic polynomial f. If at least one of its coefficients is non-zero, then there is at least one real root. For all these roots λ_0 check whether $\mathrm{tr}\,(\mathrm{adj}\,(\mathbf{Q}_{\mathbf{A},\mathbf{B}}(\lambda_0))) \leq 0$. If so, then we have found an L-quadric. If all coefficients of f are zero, then compute the quadratic polynomial g. There is a real root λ_0 of g, and this root leads us to an L-quadric.

In the following, we will describe in detail how to determine a parameterization of the intersection curve between an arbitrary quadric $\mathcal{A}$ and an L-quadric $\mathcal{B}$. First, we check whether $\mathbf{B} = \mathbf{0}$. In this case $\mathcal{B}$ has the equation $\mathbf{b}^T\mathbf{x} + b_0 = 0$. If $\mathbf{b} = \mathbf{0}$, then there are two possibilities. If $b_0 = 0$, then the intersection between $\mathcal{A}$ and $\mathcal{B}$ is the whole quadric $\mathcal{A}$. If $b_0 \neq 0$, then $\mathcal{A}$ and $\mathcal{B}$ do not intersect at all. If $\mathbf{b} \neq \mathbf{0}$, then $\mathcal{B}$ is a plane and the intersection curve is a conic. In this case we set $\mathbf{n} = \mathbf{b}/|\mathbf{b}|$ and choose vectors $\mathbf{u}$ and $\mathbf{v}$ with $|\mathbf{u}| = |\mathbf{v}| = 1$ and $\mathbf{u}^T\mathbf{v} = 0$. By setting $\mathbf{T}_{1,H} = \mathrm{diag}([\mathbf{u},\mathbf{v},\mathbf{n}],1)$ we obtain

$$\mathbf{C}_H = \mathbf{T}_{1,H}^T\mathbf{B}_H\mathbf{T}_{1,H} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3 \\ 0 & 0 & c_3 & c_0 \end{bmatrix},$$

which defines a plane parallel to the (x,y)-plane. With the translation matrix

$$\mathbf{T}_{2,H} = \begin{bmatrix} \mathbf{E} & \mathbf{v} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 0 \\ 0 \\ -\frac{c_0}{2c_3} \end{bmatrix},$$

$\mathbf{T}_{2,H}^T\mathbf{C}_H\mathbf{T}_{2,H}$ is equal to the (x,y)-plane and can be parameterized by $\mathbf{x}(\lambda,\mu) = [\lambda,\mu,0]^T$. We define the transformation $\mathbf{T}_H = \mathbf{T}_{1,H}\mathbf{T}_{2,H}$ and transform the quadric $\mathcal{A}$ by setting $\mathbf{Q}_H = \mathbf{T}_H^T\mathbf{A}_H\mathbf{T}_H$. If we have a parameterization $\mathbf{x}(\mu)$ of the intersection curve between the quadric defined by $\mathbf{Q}_H$ and the (x,y)-plane, then $\mathbf{c}(\mu) = \mathbf{T}_1(\mathbf{x}(\mu)+\mathbf{v})$ is a parameterization of the intersection curve between $\mathcal{A}$ and $\mathcal{B}$. We obtain $\mathbf{x}(\mu)$ as follows. First, we insert the parameterization of the (x,y)-plane into the quadratic form defining the other quadric and obtain

$$[\lambda,\mu]\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{bmatrix}\begin{bmatrix} \lambda \\ \mu \end{bmatrix} + 2[\lambda,\mu]\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} + q_0 = 0.$$

We have already seen in the paragraph about conics how to parameterize a conic given in this form.

Now we consider the situation when $\mathbf{B} \neq \mathbf{0}$. In that case we determine a transformation matrix $\mathbf{T}_\mathsf{H}$ such that $\mathbf{T}_\mathsf{H}^\mathsf{T}\mathbf{B}_\mathsf{H}\mathbf{T}_\mathsf{H}$ is of a form that allows us to find a simple parameterization $\mathbf{x}(t,\lambda)$ in homogeneous coordinates of the L-quadric defined by that matrix. This parameterization will have the form

$$\mathbf{x}(t,\lambda) = \mathbf{p}(t) + \lambda\mathbf{r}(t),$$

where $\mathbf{p}$ and $\mathbf{r}$ are polynomial vectors satisfying the condition

$$\deg(\mathbf{p}) + \deg(\mathbf{r}) \leq 2. \tag{2.12}$$

Then we transform the quadric $\mathcal{A}$ with the transformation $\mathbf{T}_\mathsf{H}$ by defining $\mathbf{Q}_\mathsf{H} = \mathbf{T}_\mathsf{H}^\mathsf{T}\mathbf{A}_\mathsf{H}\mathbf{T}_\mathsf{H}$. Now we determine a parameterization $\mathbf{c}(t)$ in homogeneous coordinates of the intersection curve between the transformed quadrics. The intersection curve of the original quadrics $\mathcal{A}$ and $\mathcal{B}$ can then be parameterized by $\mathbf{T}_\mathsf{H}\mathbf{c}(t)$. In order to find the parameterization $\mathbf{c}(t)$, we insert $\mathbf{x}(t,\lambda)$ into the quadratic form $\mathbf{Q}_\mathsf{H}$ and obtain a polynomial

$$\begin{aligned} f(t,\lambda) = \alpha(t)\lambda^2 + \beta(t)\lambda + \gamma(t) &= 0, \quad \text{where} \\ \alpha(t) &= \mathbf{r}(t)^\mathsf{T}\mathbf{Q}_\mathsf{H}\mathbf{r}(t), \\ \beta(t) &= 2\mathbf{p}(t)^\mathsf{T}\mathbf{Q}_\mathsf{H}\mathbf{r}(t) \quad \text{and} \\ \gamma(t) &= \mathbf{p}(t)^\mathsf{T}\mathbf{Q}_\mathsf{H}\mathbf{p}(t). \end{aligned}$$

If the polynomials α and β are not both constantly zero, then we can solve this for λ and obtain a function $\lambda(t)$. Then we have $\mathbf{c}(t) = \mathbf{x}(t,\lambda(t))$. If α is not constantly zero, then this function has the form

$$\lambda(t) = \frac{-\beta(t) \pm \sqrt{D(t)}}{2\alpha(t)}, \tag{2.13}$$

where $D(t) = \beta(t)^2 - 4\alpha(t)\gamma(t)$ is the discriminant of f with respect to λ. Because of condition (2.12), we have $\deg(D) \leq 4$. In the case $\alpha \equiv 0$ and $\beta \not\equiv 0$, we have

$$\lambda(t) = -\frac{\gamma(t)}{\beta(t)}. \tag{2.14}$$

If $\alpha \equiv \beta \equiv 0$, then f does not depend on λ. In this case there are three possibilities.

1. $f \equiv 0$: Then the quadric $\mathcal{A}$ and $\mathcal{B}$ are identical.

2. $f(t) \neq 0$ for all t: Then $\mathcal{A}$ and $\mathcal{B}$ do not intersect.

3. Otherwise, for each of root t_i of $f(t) = 0$ we have a straight line $\mathbf{c}(\lambda) = \mathbf{x}(t_i, \lambda)$. The intersection curve is the union of these lines.

Now we describe how to determine the transformation matrix $\mathbf{T}_H$ and what the parameterizations $\mathbf{x}(t, \lambda)$ of the quadric defined by $\mathbf{T}_H^T \mathbf{B}_H \mathbf{T}_H$ look like. During our further computations we will always keep the matrix $\mathbf{T}_H$ as well as its inverse up to date.

First, we perform a principle axis transformation of $\mathbf{B}$. We determine a matrix $\mathbf{U}$ such that $\mathbf{U}^T\mathbf{B}\mathbf{U} = \operatorname{diag}(\mu_1, \mu_2, 0)$ and $|\mu_1| \geq |\mu_2|$. If $\mu_1 < 0$ then we multiply $\mathbf{B}_H$ by -1. This does not change the L-quadric and we can assume in the following that $\mu_1 \geq 0$. We set $\mathbf{U}_H = \operatorname{diag}(\mathbf{U}, 1)$ and $\mathbf{P}_{1,H} = \mathbf{U}_H^T \mathbf{B}_H \mathbf{U}_H$. We distinguish the two cases $\mu_1 > 0, \mu_2 = 0$ and $\mu_1 > 0, \mu_2 < 0$.

Case 1: $\mu_1 > 0, \mu_2 = 0$
Then $\mathbf{P}_{1,H}$ defines the empty set, a double plane, two parallel planes or a parabolic cylinder. $\mathbf{P}_{1,H}$ has the form

$$\begin{bmatrix} \mu & 0 & 0 & p_1 \\ 0 & 0 & 0 & p_2 \\ 0 & 0 & 0 & p_3 \\ p_1 & p_2 & p_3 & p_0 \end{bmatrix}.$$

We distinguish two cases.

1. $p_2 = p_3 = 0$: In this case $\mathbf{P}_{1,H}$ defines the empty set, a double plane or two parallel planes. We define a translation by

$$\mathbf{T}_{1,H} = \begin{bmatrix} \mathbf{E} & \mathbf{v} \\ 0^T & 1 \end{bmatrix}, \quad \mathbf{v} = \left[-\frac{p_1}{\mu}, 0, 0\right]^T$$

which yields $\mathbf{P}_{2,H} = \mathbf{T}_{1,H}^T \mathbf{P}_{1,H} \mathbf{T}_{1,H} = \operatorname{diag}(\mu, 0, 0, q_0)$. We make one more transformation by setting $\mathbf{T}_{2,H} = \operatorname{diag}(\frac{1}{\sqrt{\mu}}, 1, 1, s)$ and $\mathbf{P}_{3,H} = \mathbf{T}_{2,H}^T \mathbf{P}_{2,H} \mathbf{T}_{2,H}$ with

$$s = \begin{cases} \frac{1}{\sqrt{|q_0|}} & \text{if } q_0 \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

Then we have $\mathbf{P}_{3,H} = \operatorname{diag}(1, 0, 0, t)$ with $t \in \{\pm 1, 0\}$. The transformation matrix is given by $\mathbf{T}_H = \mathbf{U}_H \mathbf{T}_{1,H} \mathbf{T}_{2,H}$. If $t = 1$, then $\mathbf{P}_{3,H}$ defines the empty set and the two quadrics do not intersect. In the case $t = 0$ we have a double plane that can be parameterized with $\mathbf{x}(\lambda, \mu) = [0, \lambda, \mu, 1]^T$. If $t = -1$, then we have two parallel planes $\mathbf{x}(\lambda, \mu) = [\pm 1, \lambda, \mu, 1]^T$.

2. $p_2 \neq 0$ or $p_3 \neq 0$: W.l.o.g. we assume that $p_2 \neq 0$ and define the transformation matrix

$$\mathbf{T}_{1,H} \begin{bmatrix} T_1 & \mathbf{v} \\ 0 & 1 \end{bmatrix}, \quad \mathbf{T}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & R_1 & R_2 \\ 0 & -R_2 & R_1 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ 0 \end{bmatrix} \quad \text{with}$$

$$R_1 = \frac{p_3}{\sqrt{p_2^2+p_3^2}}, \qquad R_2 = \frac{p_2}{\sqrt{p_2^2+p_3^2}},$$
$$\nu_1 = -\frac{p_1}{\mu}, \qquad \nu_2 = \frac{p_1^2 - p_0\mu}{2p_2\mu}.$$

With this definition the matrix $\mathbf{P}_{2,H} = \mathbf{T}_{1,H}^T \mathbf{P}_{1,H} \mathbf{T}_{1,H}$ has the form

$$\mathbf{P}_{2,H} = \operatorname{diag}\left(1, 0, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right).$$

We finally scale with the matrix

$$\begin{aligned} \mathbf{T}_{2,H} &= \operatorname{diag}\left(1, 1, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right) \quad \text{and obtain} \\ \mathbf{T}_{3,H} &= \mathbf{T}_{2,H}^T \mathbf{P}_{2,H} \mathbf{T}_{2,H} = \operatorname{diag}\left(1, 1, \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}\right). \end{aligned}$$

This defines a parabolic cylinder with the parameterization $\boldsymbol{x}(t, \lambda) = [t, \lambda, -t^2, 1]^T$.

$$\mathbf{T}_H = \mathbf{U}_H \mathbf{T}_{1,H} \mathbf{T}_{2,H} \mathbf{T}_{3,H}.$$

Case 2: $\mu_1 > 0, \mu_2 < 0$
In this case

$$\mathbf{P}_{1,H} = \begin{bmatrix} \mu_1 & 0 & 0 & p_1 \\ 0 & \mu_2 & 0 & p_2 \\ 0 & 0 & 0 & p_3 \\ p_1 & p_2 & p_3 & p_0 \end{bmatrix}$$

defines a hyperbolic paraboloid, a hyperbolic cylinder or two intersecting planes. We distinguish two cases.

1. $p_3 \neq 0$: First we define a translation

$$\mathbf{T}_{1,H} = \begin{bmatrix} \mathbf{E} & \boldsymbol{v} \\ 0^T & 1 \end{bmatrix}, \quad \boldsymbol{v} = [\nu_1, \nu_2, \nu_3]^T \quad \text{with}$$

$$\nu_1 = -\frac{p_1}{\mu_1}, \quad \nu_2 = -\frac{p_2}{\mu_2}, \quad \nu_3 = -\frac{p_1\nu_1 + p_2\nu_2 + p_0}{2p_3}.$$

We set $\mathbf{P}_{2,H} = -\frac{1}{p_3}\mathbf{T}_{1,H}^T \mathbf{P}_{1,H} \mathbf{T}_{1,H}$ and obtain

$$\mathbf{P}_{2,H} = \operatorname{diag}\left(\begin{bmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}\right)$$

with $\gamma_1 < 0$ and $\gamma_2 > 0$. Scaling with the matrix

$$\begin{aligned} \mathbf{T}_{2,H} &= \operatorname{diag}\left(\frac{1}{\sqrt{|\gamma_1|}}, \frac{1}{\sqrt{|\gamma_2|}}, 1, 1\right) \quad \text{leads to} \\ \mathbf{P}_{3,H} &= \mathbf{T}_{2,H}^T \mathbf{P}_{2,H} \mathbf{T}_{2,H} = \operatorname{diag}\left(\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}\right). \end{aligned}$$

We finally define the rotation

$$\begin{aligned}\mathbf{T}_{3,\mathrm{H}} &= \operatorname{diag}\left(\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, 1, 1\right) \quad \text{and obtain} \\ \mathbf{P}_{4,\mathrm{H}} &= \mathbf{T}_{3,\mathrm{H}}^{\mathsf{T}}\mathbf{P}_{3,\mathrm{H}}\mathbf{T}_{3,\mathrm{H}} = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}\right).\end{aligned}$$

This defines a hyperbolic paraboloid with the parameterization $x(t,\lambda) = [t,\lambda,t\lambda,1]^{\mathsf{T}}$. The transformation matrix is given by $\mathbf{T}_{\mathrm{H}} = \mathbf{U}_{\mathrm{H}}\mathbf{T}_{1,\mathrm{H}}\mathbf{T}_{2,\mathrm{H}}\mathbf{T}_{3,\mathrm{H}}$.

2. $p_3 = 0$: We define the translation

$$\mathbf{T}_{1,\mathrm{H}} = \begin{bmatrix} \mathbf{E} & \boldsymbol{\nu} \\ 0^{\mathsf{T}} & 1 \end{bmatrix} \quad \text{with} \quad \boldsymbol{\nu} = \left[-\frac{p_1}{\mu_1}, -\frac{p_2}{\mu_2}, 0\right]$$

and obtain $\mathbf{P}_{2,\mathrm{H}} = \mathbf{T}_{1,\mathrm{H}}^{\mathsf{T}}\mathbf{P}_{1,\mathrm{H}}\mathbf{T}_{1,\mathrm{H}} = \operatorname{diag}(\mu_1,\mu_2,0,d_0)$. We distinguish two subcases.

a) $d_0 \neq 0$: If $d_0 < 0$, then we set $\mathbf{T}_{2,\mathrm{H}} = \mathbf{E}$. Otherwise define

$$\mathbf{T}_{2,\mathrm{H}} = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, 1, 1\right).$$

We set $\mathbf{P}_{3,\mathrm{H}} = -\frac{1}{d_0}\mathbf{T}_{2,\mathrm{H}}^{\mathsf{T}}\mathbf{P}_{2,\mathrm{H}}\mathbf{T}_{2,\mathrm{H}} = \operatorname{diag}(\rho_1,\rho_2,0,-1)$ with $\rho_1 > 0$ and $\rho_2 < 0$. Finally, we scale with the matrix

$$\begin{aligned}\mathbf{T}_{3,\mathrm{H}} &= \operatorname{diag}\left(\frac{1}{\sqrt{\rho_1}}, \frac{1}{\sqrt{-\rho_2}}, 1, 1\right) \quad \text{and obtain} \\ \mathbf{P}_{4,\mathrm{H}} &= \mathbf{T}_{3,\mathrm{H}}^{\mathsf{T}}\mathbf{P}_{3,\mathrm{H}}\mathbf{T}_{3,\mathrm{H}} = \operatorname{diag}(1,-1,0,-1),\end{aligned}$$

which defines a hyperbolic cylinder with the parameterization

$$x(t,\lambda) = [\pm(1+t^2), 2t, \lambda, 1-t^2]^{\mathsf{T}}.$$

The transformation matrix is $\mathbf{T}_{\mathrm{H}} = \mathbf{U}_{\mathrm{H}}\mathbf{T}_{1,\mathrm{H}}\mathbf{T}_{2,\mathrm{H}}\mathbf{T}_{3,\mathrm{H}}$.

b) $d_0 = 0$: Then scaling and rotating with

$$\begin{aligned}\mathbf{T}_{2,\mathrm{H}} &= \operatorname{diag}\left(\frac{1}{\sqrt{\mu_1}}, \frac{1}{\sqrt{-\mu_2}}, 1, 1\right) \cdot \operatorname{diag}\left(\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, 1, 1\right) \quad \text{leads to} \\ \mathbf{P}_{3,\mathrm{H}} &= \mathbf{T}_{2,\mathrm{H}}^{\mathsf{T}}\mathbf{P}_{2,\mathrm{H}}\mathbf{T}_{2,\mathrm{H}} = \operatorname{diag}\left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, 0, 0\right).\end{aligned}$$

This defines two intersecting planes with the parameterizations

$$\begin{aligned} x_1(t,\lambda) &= [t,0,\lambda,1]^{\mathsf{T}} \quad \text{and} \\ x_2(t,\lambda) &= [0,t,\lambda,1]^{\mathsf{T}}, \end{aligned}$$

and the transformation matrix is $\mathbf{T}_{\mathrm{H}} = \mathbf{U}_{\mathrm{H}}\mathbf{T}_{1,\mathrm{H}}\mathbf{T}_{2,\mathrm{H}}$.

2.4 Mathematical Preliminaries

In this section we will give some mathematical background that will be used in the following chapters. We start by giving a brief introduction to interval arithmetic. This will be used later in this section when we describe methods for finding roots of polynomials, as well as in 3.1.1 where it will be applied in the dynamic collision detection for bounding volumes. Next, we will introduce the notion of resultants, which we will use for the elimination of variables in systems of polynomial equations. Then, we will derive necessary conditions for the existence of tangential intersection points between two quadrics or two conics. After that, we will describe two concepts for representing rotations, namely the rotation matrices and the quaternions. We close this section by presenting some methods for finding the roots of polynomials.

2.4.1 Interval Arithmetic

In this section we give a very brief introduction to interval arithmetic. A good introduction to this field can be found in [Sny92].

The general idea is to calculate with intervals of real numbers instead of the real numbers themselves. To make this more precise, let $\mathbb{IR} = \{[a,b] \subset \mathbb{R} \mid a \leq b\}$ denote the set of non-empty intervals. Then, for any function $f : \mathbb{R} \to \mathbb{R}$ an *inclusion function* $\tilde{f} : \mathbb{IR} \to \mathbb{IR}$ is associated, such that for an interval $I \in \mathbb{IR}$ it holds that

$$x \in I \Rightarrow f(x) \in \tilde{f}(I).$$

This definition is generalized componentwise to functions $f : \mathbb{R}^n \to \mathbb{R}^k$. Usually, it is clear from the context whether the argument is a real number or an interval. Thus, we use the same symbol for both the real function and the inclusion function. Ideally, the inclusion function bounds exactly the function range for every interval. But often, these are too difficult to compute. We now present inclusion functions for the field operations, the absolute value function, the square root, the sine and cosine functions and the power functions with natural exponents. Let $a, b, c, d \in \mathbb{R}$ with $a \leq b$ and $c \leq d$. Then we define

$$\begin{aligned}
[a,b] + [c,d] &= [a+c, b+d], \\
[a,b] - [c,d] &= [a-d, b-c], \\
[a,b] \cdot [c,d] &= [\min\{ac, ad, bc, bd\}, \max\{ac, ad, bc, bd\}], \\
1/[a,b] &= [1/b, 1/a] \quad \text{for } a > 0 \text{ or } b < 0, \\
[a,b]/[c,d] &= [a,b] \cdot (1/[c,d]) \quad \text{for } c > 0 \text{ or } d < 0, \\
|[a,b]| &= \begin{cases} [0, \max\{|a|, |b|\}] & \text{if } 0 \in [a,b] \\ [\min\{|a|, |b|\}, \max\{|a|, |b|\}] & \text{otherwise.} \end{cases} \\
\sqrt{[a,b]} &= [\sqrt{a}, \sqrt{b}] \quad \text{for } a \geq 0.
\end{aligned}$$

For the sine function on the interval $[a, b]$ with $a \leq b$ we first define the values $\hat{a} = 2a/\pi$ and $\hat{b} = 2b/\pi$. Then we can define

$$\begin{aligned}\sin[a,b] &= [c,d] \quad \text{with} \\ c &= \begin{cases} -1 & \text{if } \exists n \in \mathbb{Z} \text{ such that } 4n+3 \in [\hat{a}, \hat{b}], \\ \min\{\sin a, \sin b\} & \text{otherwise,} \end{cases} \\ d &= \begin{cases} 1 & \text{if } \exists n \in \mathbb{Z} \text{ such that } 4n+1 \in [\hat{a}, \hat{b}], \\ \max\{\sin a, \sin b\} & \text{otherwise.} \end{cases}\end{aligned}$$

The cosine can be defined as

$$\cos[a,b] = \sin\left([a,b] + \left[\frac{\pi}{2}, \frac{\pi}{2}\right]\right).$$

For the power functions x^n with natural exponent we observe that for odd values of n they are increasing for all x and for even values of n they are decreasing for $x \leq 0$ and increasing for $x > 0$. Hence, we define

$$[a,b]^n = \begin{cases} [a^n, b^n] & \text{if } 0 < a \text{ or } n \text{ odd}, \\ [0, \max\{|a|, |b|\}^n] & \text{if } 0 \in [a,b] \text{ and } n \text{ even}, \\ [b^n, a^n] & \text{if } b < 0 \text{ and } n \text{ even}. \end{cases}$$

The following simple example shows that it makes sense to define the inclusion functions for the power functions in this way rather than expressing them as successive multiplications. We want to compute the square of the interval $[-1, 2]$. If we use the inclusion function for the multiplication, we obtain $[-1, 2] \cdot [-1, 2] = [-2, 4]$. Using the above definition for the power function, we get $[-1, 2]^2 = [0, 4]$, which is obviously a much tighter range.

2.4.2 Resultants

Resultants are a strong tool to eliminate variables from a system of polynomial equations. They will be extensively used in the following chapters. We will first define them for the case of a single variable and then extend the notion to multivariate polynomials. More details about this subject can be found in [CLO97].

In the univariate case, the resultant between two polynomials $f, g \in \mathbb{C}[x]$ is an integer polynomial in the coefficients of f and g. Since these coefficients are complex numbers, the resultant is a complex number, as well. It will be zero if and only if f and g have a common root. Hence, resultants can be used to decide whether two polynomials have a common factor without computing their greatest common divisor. The basic observation is the following lemma.

Lemma 2.14. *Let k be a field and let $f, g \in k[x]$ be polynomials of positive degrees l and m, respectively. Then f and g have a common factor of positive degree in $k[x]$ if and only if there are polynomials $A, B \in k[x]$ such that*

1. *A and B are not both zero.*
2. *$\deg(A) \leq m - 1$, $\deg(B) \leq l - 1$.*
3. *$Af + Bg = 0$.*

Proof. Let us first assume that f and g have a common factor h of positive degree. Thus, we can write $f = hf_1$ and $g = hg_1$, where $\deg(f_1) \leq l - 1$ and $\deg(g_1) \leq m - 1$. Obviously, $g_1 f = f_1 g$ and hence, the polynomials $A = g_1$ and $B = -f_1$ have the desired properties.

Conversely, let us assume that polynomials A and B with above three properties exist. By the first property, we may assume without loss of generality that $B \neq 0$. Suppose that the claim does not hold, i.e. $\gcd(f, g) = 1$. Then there are polynomials $\tilde{A}$ and $\tilde{B}$ such that $\tilde{A}f + \tilde{B}g = 1$. If we multiply this by B and use the fact that $Bg = -Af$ we obtain $(\tilde{A}B - A\tilde{B})f = B$. Since B is non-zero, the degree of B must be at least l, which is a contradiction to the second property. □

In the sequel, we choose $k = \mathbb{C}$. Then two polynomials $f, g \in \mathbb{C}[x]$ have a common factor if and only if they have a common root. Hence, lemma 2.14 gives a necessary condition for two univariate polynomials to have a common complex root. We use linear algebra to decide the existence of the polynomials A and B. Therefore we write

$$\begin{aligned} f(x) &= \sum_{i=0}^{l} a_i x^i, & g(x) &= \sum_{i=0}^{m} b_i x^i, \\ A(x) &= \sum_{i=0}^{m-1} c_i x^i, & B(x) &= \sum_{i=0}^{l-1} d_i x^i. \end{aligned} \tag{2.15}$$

Moreover, we write

$$A(x)f(x) + B(x)g(x) = \sum_{i=0}^{l+m-1} e_i x^i.$$

We define the vectors $\boldsymbol{e} = [e_{l+m-1}, \ldots, e_0]^T$ and $\mathbf{c} = [c_{m-1}, \ldots, c_0, d_{l-1}, \ldots, d_0]^T$. These two vectors are linearly related by $\boldsymbol{e} = \mathbf{S}\mathbf{c}$, where the $(l + m) \times (l + m)$-matrix $\mathbf{S}$ has the form

$$\mathbf{S} = \begin{bmatrix} a_l & & & & b_m & & & \\ a_{l-1} & a_l & & & b_{m-1} & b_m & & \\ a_{l-2} & a_{l-1} & \ddots & & b_{m-2} & b_{m-1} & \ddots & \\ \vdots & & \ddots & a_l & \vdots & & \ddots & b_m \\ & \vdots & & a_{l-1} & & \vdots & & b_{m-1} \\ a_0 & & & & b_0 & & & \\ & a_0 & & \vdots & & b_0 & & \vdots \\ & & \ddots & & & & \ddots & \\ & & & a_0 & & & & b_0 \end{bmatrix}.$$

The empty spaces are filled by zeros. By lemma 2.14, the polynomials f and g have a common factor if and only if there is a non-zero vector $\mathbf{c}$ such that $\mathbf{S}\mathbf{c} = 0$. This is equivalent to $\det \mathbf{S} = 0$. We make the following definition.

Definition 2.15. *Let the non-constant polynomials* $f, g \in \mathbb{C}[x]$ *be written in the form (2.15) and let the matrix* $\mathbf{S}$ *be defined as above. We call the matrix* $\mathbf{Syl}_x(f, g) = \mathbf{S}^T$ *the* Sylvester matrix *of* f *and* g *with respect to* x*. The* resultant *of* f *and* g *with respect to* x *is the determinant of the Sylvester matrix. Thus,*

$$\operatorname{res}_x(f, g) = \det(\mathbf{Syl}_x(f, g)).$$

The above argumentation and the definition of the determinant of a matrix prove the following lemma.

Lemma 2.16. *Let* $f, g \in \mathbb{C}[x]$ *have positive degrees. The resultant* $\operatorname{res}_x(f, g)$ *is an integer polynomial in the coefficients of* f *and* g*. Moreover,* f *and* g *have a common complex root if and only if* $\operatorname{res}_x(f, g) = 0$*.*

With the resultants we have a tool to decide whether two polynomials have a common root without computing their GCD. We want to extend this notion to polynomials of multiple variables.

Let now $f, g \in \mathbb{C}[x_1, \dots, x_n]$ be multivariate polynomials with positive degrees in x_1. We write f and g in the form (2.15) with x_1 instead of x, where now the coefficients a_i and b_j are polynomials in $\mathbb{C}[x_2, \dots, x_n]$. We define the Sylvester matrix and the resultant of f and g with respect to x_1 in the same way as in the univariate case. By lemma 2.16, the resultant is now a polynomial in $\mathbb{C}[x_2, \dots, x_n]$. The following lemma shows that in the resultant of two polynomials f and g with respect to x_1 the variable x_1 is eliminated in the sense that the projections of the common roots of f and g into the $(x_2, \dots, x_n)$-subspace are also roots of the resultant.

Lemma 2.17. *Let* $f, g \in \mathbb{C}[x_1, \dots, x_n]$ *be polynomials with positive degrees in* x_1 *and let* $(c_1, \dots, c_n) \in \mathbb{C}^n$*. If* $f(c_1, \dots, c_n) = g(c_1, \dots, c_n) = 0$*, then* $\operatorname{res}_{x_1}(f, g)(c_2, \dots, c_n) = 0$*.*

Proof. To make the proof easier we introduce some notation. We write $\mathbf{c} = (c_2, \dots, c_n)$ and f and g as in (2.15). Moreover we define $\tilde{f}(x_1) = f(x_1, \mathbf{c})$ and $\tilde{g}(x_1) = g(x_1, \mathbf{c})$. We first notice that $\operatorname{res}_{x_1}(f, g)(\mathbf{c}) = \det\big(\mathbf{Syl}_{x_1}(f, g)(\mathbf{c})\big)$, i.e. it does not matter whether we first take the determinant of $\mathbf{Syl}_{x_1}(f, g)$ and then evaluate this at $\mathbf{c}$ or first evaluate the matrix and then compute the determinant. If $a_l(\mathbf{c}) = b_m(\mathbf{c}) = 0$, then the first row of $\mathbf{Syl}_{x_1}(f, g)(\mathbf{c})$ is zero. Hence, the resultant is zero, as well. In the case that $a_l(\mathbf{c})$ and $b_m(\mathbf{c})$ are both non-zero, it is obviously true that $\mathbf{Syl}_{x_1}(f, g)(\mathbf{c}) = \mathbf{Syl}_{x_1}(\tilde{f}, \tilde{g})$. Since $\tilde{f}(c_1) = \tilde{g}(c_1) = 0$, the claim follows by lemma 2.16.

So, let us assume that $a_l(\mathbf{c}) = 0$ and $b_m(\mathbf{c}) \neq 0$. If $a_{l-1}(\mathbf{c}) \neq 0$, then expanding the determinant of $\mathbf{Syl}_{x_1}(f, g)(\mathbf{c})$ by cofactors of the first column yields

$$\operatorname{res}_{x_1}(f, g)(\mathbf{c}) = \pm b_m(\mathbf{c}) \cdot \det\big(\mathbf{Syl}_{x_1}(\tilde{f}, \tilde{g})\big) = \pm b_m(\mathbf{c}) \cdot \operatorname{res}_{x_1}(\tilde{f}, \tilde{g}),$$

and the claim follows again by lemma (2.16). If a_{l-1} is also zero and $a_{l-2} \neq 0$, then the same argument yields

$$\mathrm{res}_{x_1}(f,g)(\mathbf{c}) = \pm b_m(\mathbf{c})^2 \cdot \mathrm{res}_{x_1}(\tilde{f},\tilde{g}) = 0.$$

In this way, we obtain $\mathrm{res}_{x_1}(f,g)(\mathbf{c}) = \pm b_m(\mathbf{c})^k \cdot \mathrm{res}_{x_1}(\tilde{f},\tilde{g})$ for some k if there is an $i \geq 1$ such that $a_i(\mathbf{c}) \neq 0$. Otherwise, from $\tilde{f}(c_1) = 0$ it follows that $a_0(\mathbf{c})$ must be zero, as well. Thus, the first row of $\mathbf{Syl}_{x_1}(f,g)(\mathbf{c})$ is zero, which completes the proof. □

The next lemma shows that each root of the resultant of f and g either causes both coefficients a_l and b_m to vanish or can be extended to a common root of f and g.

Lemma 2.18. *Let* $f, g \in \mathbb{C}[x_1,\ldots,x_n]$ *be polynomials with positive degrees in* x_1 *given in the form (2.15), and let* $(c_2,\ldots c_n) \in \mathbb{C}^{n-1}$. *If* $\mathrm{res}_{x_1}(f,g)(c_2,\ldots c_n) = 0$*, then either*

1. $a_l(c_2,\ldots,c_n) = b_m(c_2,\ldots c_n) = 0$ *or*

2. *there is a* $c_1 \in \mathbb{C}$ *such that* $f(c_1,\ldots c_n) = g(c_1,\ldots c_n) = 0$.

Proof. We use the notation introduced in lemma 2.17. Let $b_m(\mathbf{c}) \neq 0$. We must show that there is a $c_1 \in \mathbb{C}$ such that $\tilde{f}(c_1) = \tilde{g}(c_1) = 0$. As in the proof of lemma 2.17, one shows that if there is an $i \geq 1$ with $a_i(\mathbf{c}) \neq 0$ then there is some k such that

$$\mathrm{res}_{x_1}(f,g)(\mathbf{c}) = \pm b_m(\mathbf{c})^k \cdot \det\left(\mathbf{Syl}_{x_1}(\tilde{f},\tilde{g})\right)$$

and the claim follows by lemma 2.16. In the case that $a_1(\mathbf{c}) = \ldots = a_l(\mathbf{c}) = 0$ we find by successive expansion by cofactors of the first row that $\mathrm{res}_{x_1}(f,g)(\mathbf{c}) = \pm a_0(\mathbf{c})^m \cdot b_m(\mathbf{c})^l$. Since $b_m(\mathbf{c}) \neq 0$ by assumption, we conclude that $a_0(\mathbf{c}) = 0$ and thus, $\tilde{f} \equiv 0$. But since $\tilde{g}$ is not constant, the claim follows. □

The following lemma gives an upper bound on the degree of the resultant of two polynomials.

Lemma 2.19. *Let* $f, g \in \mathbb{C}[x_1,\ldots x_n]$ *be polynomials. Let* $l > 0$ *be the degree of* f *in* x_1 *and let* $m > 0$ *be the degree of* g *in* x_1. *Let* d *denote the total degree of* f *and* e *the total degree of* g. *Then, the total degree of* $\mathrm{res}_{x_1}(f,g)$ *is at most* $el + dm - lm$.

Proof. As usual, we write f and g in the form (2.15). First, we notice that the total degree of a_i is at most $d-i$ and the total degree of b_i is at most $e-i$. Let $(s_{i,j}) = \mathbf{Syl}_{x_1}(f,g)$ be the Sylvester matrix. We observe that

$$s_{i,j} = \begin{cases} a_{l+i-j} & \text{if } i \leq m \\ b_{i-j} & \text{if } i > m \end{cases}$$

and thus, for the total degree it holds

$$\deg(s_{i,j}) \leq \begin{cases} d - l + j - i & \text{if } i \leq m \\ e + j - i & \text{if } i > m. \end{cases}$$

The resultant is the determinant of the matrix $(s_{i,j})$ and can therefore be written as the sum of products of the form

$$P_\sigma = \pm \prod_{i=1}^{l+m} c_{\sigma(i),i} = \pm \prod_{\sigma(i) \leq m} c_{\sigma(i),i} \cdot \prod_{\sigma(i) > m} c_{\sigma(i),i},$$

where σ is a permutation of the set $\{1, \dots, l+m\}$. In order to bound the degree we can assume that σ is chosen in such a way that none of the $c_{\sigma(i),i}$ is zero. Then, for the total degree of P_σ it holds

$$\begin{aligned} \deg(P_\sigma) &= \sum_{\sigma(i) \leq m} \deg(c_{\sigma(i),i}) + \sum_{\sigma(i) > m} \deg(c_{\sigma(i),i}) \\ &\leq \sum_{\sigma(i) \leq m} d - l + i - \sigma(i) + \sum_{\sigma(i) > m} e + i - \sigma(i) \\ &= m(d - l) + le + \sum_{i=1}^{l+m} i - \sum_{i=1}^{l+m} \sigma(i) \\ &= el + dm - lm. \end{aligned}$$

□

In the important special case that the total degrees of f and g are equal to the individual degrees in x_1, this bound simplifies to lm.

Often, one obtains a better bound on the degree of the resultant by considering the degree matrix of the Sylvester matrix, which we define as follows.

Definition 2.20. *Let $(s_{i,j}) = \mathbf{Syl}_{x_1}(f, g)$ be the Sylvester matrix of two polynomials f and g with respect to x_1. We call the matrix $(\deg(s_{i,j}))$ the degree matrix of $(s_{i,j})$. We define the degree of the zero-polynomial as $-\infty$.*

Let $\mathbf{D} = (d_{i,j})$ be the degree matrix of the Sylvester matrix of f and g. Let $\mathbf{D}_{ij}$ denote the submatrix of $\mathbf{D}$ that is obtained by deleting the ith row and the jth column. Then we define the value $\delta_{\mathbf{D}}$ recursively as follows. For a (1×1)-matrix (d) we set $\delta_{(d)} = d$. Otherwise, we choose any index $i = 1, \dots, l+m$ and define

$$\delta_{\mathbf{D}} = \max\{d_{i,j} + \delta_{\mathbf{D}_{ij}} \mid 1 \leq j \leq l+m\}.$$

This definition is similar to the expansion of the determinant by cofactors of the ith row. The difference is that multiplication is replaced by summation and summation is replaced by taking the maximum. Hence, two things are obvious. Firstly, from this definition it is clear that $\deg(\mathrm{res}_{x_1}(f,g)) \le \delta_{\mathbf{D}}$. Secondly, $\delta_{\mathbf{D}}$ is well defined, i.e. it does neither depend on the choice of the index i nor on the fact that we iterate along a row of $\mathbf{D}$ instead of a column.

It is often the case that the resultant of two polynomials is a product of two polynomials of smaller degrees. With the following lemma we identify such a situation in which the resultant can be factorized. However, it does not provide a method to find the non-trivial one of the two factors.

Lemma 2.21. *Let* $f_1, f_2 \in \mathbb{C}[x_1, \ldots, x_n, y]$ *be polynomials that can be written in the form*

$$f_i(\mathbf{x}, y) = (x_1 - \xi)^k \cdot r_i(\mathbf{x}, y) + \prod_{j=1}^{l} (y - \zeta_j) \cdot s_i(\mathbf{x}, y) \tag{2.16}$$

for $i = 1, 2$ *with* $k, l \in \mathbb{N}, \xi, \zeta_j \in \mathbb{C}, r_i, s_i \in \mathbb{C}[x_1, \ldots, x_n, y]$ *and* $\mathbf{x} = (x_1, \ldots, x_n)$. *Then there exists a polynomial* $h \in \mathbb{R}[x_1, \ldots, x_n]$ *such that*

$$\mathrm{res}_y(f_1, f_2)(\mathbf{x}) = (x_1 - \xi)^{kl} \cdot h(\mathbf{x}).$$

Proof. First, we assume that the ζ_j are pairwise different and prove the claim by induction on k. Let $k = 1$. We shear the coordinate system in such a way that the x_1-coordinates of the roots (ξ, ζ_j) of f_1 and f_2 become pairwise different. Formally this means that we define the polynomials $f_{i,\varepsilon}(\mathbf{x}, y) = f_i(x_1 - \varepsilon y, x_2, \ldots, x_n, y)$ for $\varepsilon \in \mathbb{R}$. If we set $\xi_{j,\varepsilon} = \xi + \varepsilon\zeta_j$, then because the ζ_j are pairwise different, for $\varepsilon \neq 0$ we have $\xi_{i,\varepsilon} \neq \xi_{j,\varepsilon}$ for $i \neq j$ and moreover $f_{i,\varepsilon}(\xi_{j,\varepsilon}, x_2, \ldots, x_n, \zeta_j) = 0$ for all $j = 1, \ldots, l$. Thus, it also holds that $\mathrm{res}_y(f_{1,\varepsilon}, f_{2,\varepsilon})(\xi_{j,\varepsilon}, x_2, \ldots, x_n) = 0$ for all j. So, for $\varepsilon \neq 0$ there exists a polynomial h_ε such that $\mathrm{res}_y(f_{1,\varepsilon}, f_{2,\varepsilon})(\mathbf{x}) = (x_1 - \xi_{1,\varepsilon}) \cdots (x_1 - \xi_{l,\varepsilon}) \cdot h_\varepsilon(\mathbf{x})$. Thus, we have

$$\begin{aligned} \mathrm{res}_y(f_1, f_2)(\mathbf{x}) &= \lim_{\varepsilon \longrightarrow 0} \mathrm{res}_y(f_{1,\varepsilon}, f_{2,\varepsilon})(\mathbf{x}) \\ &= (x_1 - \xi)^l \cdot h(\mathbf{x}) \end{aligned}$$

with $h = h_0$. Now assume that the claim is true for $k \in \mathbb{N}$. In order to prove the claim for $k + 1$ we define

$$\tilde{f}_i(\mathbf{x}, y) = (x - \xi)^k (x - \tilde{\xi}) \cdot r_i(\mathbf{x}, y) + \prod_{j=1}^{l} (y - \zeta_j) \cdot s_i(\mathbf{x}, y)$$

for $i = 1, 2$. Then, there are polynomials h_1 and h_2 such that

$$\begin{aligned} \mathrm{res}_y(\tilde{f}_1, \tilde{f}_2)(\mathbf{x}) &= (x_1 - \xi)^{kl} \cdot h_1(\mathbf{x}) \text{ by induction hypothesis} \\ &= (x_1 - \tilde{\xi})^l \cdot h_2(\mathbf{x}) \text{ by the case } k = 1. \end{aligned}$$

If $\tilde{\xi} \neq \xi$, then $\gcd\left((x_1-\tilde{\xi})^l, (x_1-\xi)^{kl}\right) = 1$, and thus there is a polynomial h such that $h_1(\mathbf{x}) = (x_1 - \tilde{\xi})^l \cdot h(\mathbf{x})$. Therefore, we have

$$\begin{aligned} \mathrm{res}_y(f_1, f_2)(\mathbf{x}) &= \lim_{\tilde{\xi} \longrightarrow \xi} \mathrm{res}_y(\tilde{f}_1, \tilde{f}_2)(\mathbf{x}) \\ &= (x_1 - \xi)^{(k+1)l} \cdot h(\mathbf{x}). \end{aligned}$$

Now, we consider the case that the ζ_j are not pairwise different. Let $r_j, \varphi_j \in \mathbb{R}$ be chosen in such a way that $\zeta_j = r_j e^{I\varphi_j}$. We use the capital letter $I = \sqrt{-1}$ to avoid confusion with the index i. W.l.o.g. we assume that $r_1 \leq \ldots \leq r_l$. For $\varepsilon > 0$ we define $\zeta_{j,\varepsilon} = r_{j,\varepsilon} e^{I\varphi_j}$ with $r_{j,\varepsilon} = r_j + j\varepsilon$. In this way we get

$$r_{j,\varepsilon} \leq r_{j+1} + j\varepsilon < r_{j+1} + (j+1)\varepsilon = r_{j+1,\varepsilon},$$

and hence the $\zeta_{j,\varepsilon}$ are pairwise different. We define the polynomials $f_{i,\varepsilon}$ by replacing the ζ_j in equation (2.16) by $\zeta_{j,\varepsilon}$. Then, by the above argumentation there exists a polynomial h_ε for each $\varepsilon > 0$ such that $\mathrm{res}_y(f_{1,\varepsilon}, f_{2,\varepsilon})(\mathbf{x}) = (x_1 - \xi)^{kl} \cdot h_\varepsilon(\mathbf{x})$. Finally, we have

$$\begin{aligned} \mathrm{res}_y(f_1, f_2)(\mathbf{x}) &= \lim_{\varepsilon \longrightarrow 0} \mathrm{res}_y(f_{1,\varepsilon}, f_{2,\varepsilon})(\mathbf{x}) \\ &= (x_1 - \xi)^{kl} \cdot h(\mathbf{x}) \end{aligned}$$

with $h = h_0$. □

2.4.3 Tangential Intersections Between Quadrics or Conics

The following two theorems provide a necessary condition for two quadrics to have a tangential intersection point. At the end of the section we formulate a similar theorem that provides a necessary condition for two conics to intersect tangentially.

We first define the notion of the characteristic polynomial of two quadrics.

Definition 2.22. *Let two quadrics $\mathcal{A}$ and $\mathcal{B}$ be given by the (4×4)—matrices $\mathbf{A}_\mathrm{H}$ and $\mathbf{B}_\mathrm{H}$, respectively. We define the characteristic polynomial of $\mathcal{A}$ and $\mathcal{B}$ as the determinant of the pencil of $\mathcal{A}$ and $\mathcal{B}$, i.e. $\chi_{\mathbf{A},\mathbf{B}}(\lambda) = \det \mathbf{Q}_{\mathbf{A},\mathbf{B},\mathrm{H}}(\lambda)$.*

The first theorem is proven in [WWK01]. It gives necessary and sufficient conditions for two ellipsoids to be separate or touching.

Theorem 2.23. *Let two ellipsoids be given by the matrices $\mathbf{A}_\mathrm{H}$ and $\mathbf{B}_\mathrm{H}$, such that for their centers $\mathbf{a}_\mathrm{H}$ and $\mathbf{b}_\mathrm{H}$ it holds that $\mathbf{a}_\mathrm{H}^T \mathbf{A}_\mathrm{H} \mathbf{a}_\mathrm{H} < 0$ and $\mathbf{b}_\mathrm{H}^T \mathbf{B}_\mathrm{H} \mathbf{b}_\mathrm{H} < 0$. The ellipsoids do not intersect (and neither one contains the other) if and only if their characteristic polynomial has two positive roots. Moreover,*

1. *they are separate if and only if the two positive roots are distinct, and*

2. *they externally touch each other if and only if the two positive roots are a double root.*

In [FNO89] it is stated that any two quadrics have a degenerate intersection in projective space over the complex numbers if and only if their characteristic polynomial has a double (real or complex) root. We want to prove a necessary condition for two quadrics to have a real tangential intersection. As we do not consider faces containing singular points, we can exclude the case of such points being involved in the intersection.

Theorem 2.24. *Let two quadrics be given by the matrices* $\mathbf{A}_H$ *and* $\mathbf{B}_H$ *such that* $\chi_{\mathbf{A},\mathbf{B}} \not\equiv 0$. *If there is a real point* $\mathbf{p}_H$ *and* $\lambda_0 \in \mathbb{R}$ *such that* $\mathbf{p}_H^T \mathbf{A}_H \mathbf{p}_H = \mathbf{p}_H^T \mathbf{B}_H \mathbf{p}_H = 0$ *and* $\mathbf{A}_H \mathbf{p}_H = -\lambda_0 \mathbf{B}_H \mathbf{p}_H \neq \mathbf{0}$, *then* λ_0 *is a multiple root of* $\chi_{\mathbf{A},\mathbf{B}}$.

The exclusion of the singular points is achieved by demanding that the gradients in $\mathbf{p}_H$ are non-zero. In order to prove this theorem we recall a well known result from linear algebra, namely the Jordan normal form for real square matrices.

Definition 2.25. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two square matrices of the same dimension. We say that* $\mathbf{A}$ *is similar to* $\mathbf{B}$ *over the real numbers if there is a regular real matrix* $\mathbf{S}$ *such that* $\mathbf{B} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S}$. *We say that* $\mathbf{A}$ *is congruent to* $\mathbf{B}$ *over the real numbers if there is a regular real matrix* $\mathbf{S}$ *such that* $\mathbf{B} = \mathbf{S}^T\mathbf{A}\mathbf{S}$.

Definition 2.26. *A matrix* $\mathbf{M} \in \mathbb{R}^{k\times k}$ *for* $k \geq 1$ *is called a Jordan block of type A if it has the form*

$$\begin{aligned}
\mathbf{J}(\lambda, k) &:= \begin{bmatrix} \lambda & 1 & & & 0 \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & 1 \\ 0 & & & & \lambda \end{bmatrix} \quad \textit{for } k \geq 2 \textit{ and} \\
\mathbf{J}(\lambda, 1) &:= [\lambda]
\end{aligned}$$

for $\lambda \in \mathbb{R}$. *It is called a Jordan block of type B if it has the form*

$$\begin{aligned}
\mathbf{J}(a, b, k) &:= \begin{bmatrix} \mathbf{J}(a,b,2) & \mathbf{E}_2 & & & \mathbf{0} \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \mathbf{E}_2 \\ \mathbf{0} & & & & \mathbf{J}(a,b,2) \end{bmatrix} \quad \textit{for } k \geq 4 \textit{ and} \\
\mathbf{J}(a, b, 2) &:= \begin{bmatrix} a & -b \\ b & a \end{bmatrix}
\end{aligned}$$

for $a, b \in \mathbb{R}$ *where* $\mathbf{E}_2$ *is the* $(2 \times 2)-$*identity matrix.*

A proof for the following well known theorem from linear algebra can e.g. be found in [Kow79].

Theorem 2.27 (Jordan normal form). *Every real square matrix* $\mathbf{A}$ *is similar over the real numbers to a matrix* $\mathbf{J} = \operatorname{diag}(\mathbf{J}_1, \ldots, \mathbf{J}_s)$*, where each* $\mathbf{J}_i$ *is either of the form*

1. $\mathbf{J}(\lambda, k)$ *for a real eigenvalue* λ *of* $\mathbf{A}$ *or*

2. $\mathbf{J}(a, b, k)$ *for a pair of conjugate complex eigenvalues* $a \pm ib$ *of* $\mathbf{A}$*.*

$\mathbf{J}$ *is called the real Jordan normal form of* $\mathbf{A}$*. It is uniquely determined by* $\mathbf{A}$ *except for the order of the Jordan blocks.*

Now, we state a theorem proven in [Uhl76] which finally enables us to prove theorem 2.24.

Theorem 2.28. *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two real symmetric matrices with* $\det \mathbf{B} \neq 0$*. Let* $\mathbf{B}^{-1}\mathbf{A}$ *have the real Jordan normal form* $\operatorname{diag}(\mathbf{J}_1, \ldots, \mathbf{J}_r, \mathbf{J}_{r+1}, \mathbf{J}_s)$*, where* $\mathbf{J}_1, \ldots \mathbf{J}_r$ *are Jordan blocks of type A and* $\mathbf{J}_{r+1}, \ldots, \mathbf{J}_s$ *are Jordan blocks of type B. Then* $\mathbf{A}$ *and* $\mathbf{B}$ *are simultaneously congruent over the real numbers to*

$$\operatorname{diag}(\varepsilon_1 \mathbf{D}_1 \mathbf{J}_1, \ldots, \varepsilon_r \mathbf{D}_r \mathbf{J}_r, \mathbf{D}_{r+1} \mathbf{J}_{r+1}, \ldots, \mathbf{D}_s \mathbf{J}_s),$$

and

$$\operatorname{diag}(\varepsilon_1 \mathbf{D}_1, \ldots, \varepsilon_r \mathbf{D}_r, \mathbf{D}_{r+1}, \ldots, \mathbf{D}_s)$$

respectively, where $\varepsilon_i = \pm 1$ *and* $\mathbf{D}_i$ *denotes the matrix*

$$\begin{bmatrix} 0 & & 1 \\ & \cdot^{\cdot^{\cdot}} & \\ 1 & & 0 \end{bmatrix}$$

of the same size as $\mathbf{J}_i$ *for* $i = 1 \ldots s$*.*

Proof of theorem 2.24. It is clear that λ_0 is a root of $\chi_{\mathbf{A},\mathbf{B}}$ because there is a point $\mathbf{p}_H$ in homogeneous coordinates (in particular, at least one coordinate is non-zero) with $(\mathbf{A}_H + \lambda_0 \mathbf{B}_H)\mathbf{p}_H = 0$. We distinguish three cases.

1. $\mathbf{B}_H$ is regular,

2. $\mathbf{A}_H$ is regular and $\mathbf{B}_H$ is singular and

3. both $\mathbf{A}_H$ and $\mathbf{B}_H$ are singular.

First, we suppose we have already proven the claim for the first case and use this to prove it for the second and third case.

Let $\mathbf{A}_\mathrm{H}$ be regular and $\mathbf{B}_\mathrm{H}$ singular. Because of $\lambda_0 \neq 0$ we can define $\mu_0 = 1/\lambda_0$ and have $\mathbf{B}_\mathrm{H}\mathbf{p}_\mathrm{H} = -\mu_0\mathbf{A}_\mathrm{H}\mathbf{p}_\mathrm{H}$. Thus, by the first case we know that μ_0 is a multiple root of $\chi_{\mathbf{B},\mathbf{A}}(\mu) = \det(\mathbf{B}_\mathbf{H} + \mu\mathbf{A}_\mathrm{H})$. For all $\mu \neq 0$ we have the identity $\chi_{\mathbf{B},\mathbf{A}}(\mu) = \mu^4\chi_{\mathbf{A},\mathbf{B}}(1/\mu)$. As μ_0 is a multiple root of this polynomial it is a root of its derivative which is given by $4\mu^3\chi_{\mathbf{A},\mathbf{B}}(1/\mu) - \mu^2\chi'_{\mathbf{A},\mathbf{B}}(1/\mu)$. From $\mu_0 \neq 0$ and $\chi_{\mathbf{A},\mathbf{B}}(1/\mu_0) = \chi_{\mathbf{A},\mathbf{B}}(\lambda_0) = 0$ it follows that $\chi'_{\mathbf{A},\mathbf{B}}(\lambda_0) = 0$. This proves the claim for the second case.

Now, let both $\mathbf{A}_\mathrm{H}$ and $\mathbf{B}_\mathrm{H}$ be singular. Since $\chi_{\mathbf{A},\mathbf{B}} \not\equiv 0$ there is an $\alpha \in \mathbb{R}$ such that $\chi_{\mathbf{A},\mathbf{B}}(\alpha) = \det(\mathbf{A}_\mathrm{H} + \alpha\mathbf{B}_\mathrm{H}) \neq 0$. Clearly, $\alpha \neq \lambda_0$. Let $\mathbf{A}'_\mathrm{H} = \mathbf{A}_\mathrm{H} + \alpha\mathbf{B}_\mathrm{H}$. We have $\chi_{\mathbf{A}',\mathbf{B}} \not\equiv 0$ and $\mathbf{p}_\mathrm{H}^\mathsf{T}\mathbf{A}'_\mathrm{H}\mathbf{p}_\mathrm{H} = 0$. Moreover, $\mathbf{A}'_\mathrm{H}\mathbf{p}_\mathrm{H} = \mathbf{A}_\mathrm{H}\mathbf{p}_\mathrm{H} + \alpha\mathbf{B}_\mathrm{H}\mathbf{p}_\mathrm{H} = (\alpha - \lambda_0)\mathbf{B}_\mathrm{H}\mathbf{p}_\mathrm{H} \neq \mathbf{0}$. Hence, by case two we know that $\lambda_0 - \alpha$ is a multiple root of $\chi_{\mathbf{A}',\mathbf{B}}$. By the identity $\chi_{\mathbf{A}',\mathbf{B}}(\mu) = \chi_{\mathbf{A},\mathbf{B}}(\alpha + \mu)$ it follows that λ_0 is a multiple root of $\chi_{\mathbf{A},\mathbf{B}}$.

It remains to prove the claim for the first case. Let therefore $\mathbf{B}_\mathrm{H}$ be regular. By theorem 2.28, $\mathbf{A}_\mathrm{H}$ and $\mathbf{B}_\mathrm{H}$ are simultaneously congruent over the real numbers to block matrices $\mathbf{C}_\mathrm{H}$ and $\mathbf{D}_\mathrm{H}$, respectively. Let $\mathbf{S}$ be the transformation matrix such that $\mathbf{C}_\mathrm{H} = \mathbf{S}^\mathsf{T}\mathbf{A}_\mathrm{H}\mathbf{S}$ and $\mathbf{D}_\mathrm{H} = \mathbf{S}^\mathsf{T}\mathbf{B}_\mathrm{H}\mathbf{S}$. Then, $\chi_{\mathbf{C},\mathbf{D}} = (\det\mathbf{S})^2\chi_{\mathbf{A},\mathbf{B}} \not\equiv 0$. We define $\mathbf{q}_\mathrm{H} = \mathbf{S}^{-1}\mathbf{p}_\mathrm{H}$. We have $\mathbf{q}_\mathrm{H}^\mathsf{T}\mathbf{C}_\mathrm{H}\mathbf{q}_\mathrm{H} = \mathbf{q}_\mathrm{H}^\mathsf{T}\mathbf{D}_\mathrm{H}\mathbf{q}_\mathrm{H} = 0$ and $\mathbf{C}_\mathrm{H}\mathbf{q}_\mathrm{H} = -\lambda_0\mathbf{D}_\mathrm{H}\mathbf{q}_\mathrm{H} \neq \mathbf{0}$. If we show that λ_0 is a multiple root of $\chi_{\mathbf{C},\mathbf{D}}$ we are done. We do this by contradiction. So, let λ_0 be a single root of $\chi_{\mathbf{C},\mathbf{D}}$. Then, there are two cases.

1. There are three more single real roots λ_1, λ_2 and λ_3. In this case we have $\mathbf{C}_\mathrm{H} = \operatorname{diag}(-\varepsilon_0\lambda_0, \dots, -\varepsilon_3\lambda_3)$ and $\mathbf{D}_\mathrm{H} = \operatorname{diag}(\varepsilon_0, \dots, \varepsilon_3)$. Then, from $(\mathbf{C}_\mathrm{H} + \lambda_0\mathbf{D}_\mathrm{H})\mathbf{q}_\mathrm{H} = \mathbf{0}$ it follows that the last three coordinates of $\mathbf{q}_\mathrm{H}$ are zero. But this together with $\mathbf{q}_\mathrm{H}^\mathsf{T}\mathbf{D}_\mathrm{H}\mathbf{q}_\mathrm{H} = 0$ implies that also the first coordinate of $\mathbf{q}_\mathrm{H}$ is zero, which is a contradiction.

2. There is one more single real root λ_1 and a pair of conjugate complex roots $a \pm ib$. In that case we have

$$\begin{aligned} \mathbf{C}_\mathrm{H} &= \operatorname{diag}\left(-\varepsilon_0\lambda_0, -\varepsilon_1\lambda_1, \begin{bmatrix} -b & -a \\ -a & b \end{bmatrix}\right) \quad \text{and} \\ \mathbf{D}_\mathrm{H} &= \operatorname{diag}\left(\varepsilon_0, \varepsilon_1, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right). \end{aligned}$$

The condition

$$(\mathbf{C}_\mathrm{H} + \lambda_0\mathbf{D}_\mathrm{H})\mathbf{q}_\mathrm{H} = \operatorname{diag}\Big(0, \varepsilon_1(\lambda_0 - \lambda_1), \underbrace{\begin{bmatrix} -b & \lambda_0 - a \\ \lambda_0 - a & b \end{bmatrix}}_{=:\mathbf{M}}\Big)\mathbf{q}_\mathrm{H} = \mathbf{0}$$

implies that the last three coordinates of $\mathbf{q}_\mathrm{H}$ are zero. This is because $\lambda_0 \neq \lambda_1$ and $\det\mathbf{M} = -b^2 - (\lambda_0 - a)^2 < 0$. As in the first case, this implies together with $\mathbf{q}_H^T \mathbf{D}_H \mathbf{q}_H = 0$ that the first coordinate of $\mathbf{q}_H$ is zero, as well. This is again a contradiction. □

Since the normals in the points of a quadric define an interior and an exterior region, we say that two quadrics touch externally if their normals in the touching point are directed towards each other.

Corollary 2.29. *Let $\mathbf{A}_H$ and $\mathbf{B}_H$ define two quadrics such that $\chi_{\mathbf{A},\mathbf{B}} \not\equiv 0$. If these quadrics touch externally in a real point which is not a singular point of either of them, then their characteristic polynomial has a positive multiple root.*

Proof. Theorem 2.24 says that λ_0 is a multiple root of the characteristic polynomial, where $-\lambda_0$ is the factor with which the normal of the second quadric has to be multiplied to obtain the normal of the first quadric. If these normals are opposite directed, then λ_0 must be positive. □

We state another theorem which gives a necessary condition for two conics in the same plane to have a tangential intersection. This theorem is similar to theorem 2.24, which also holds for its proof, which we omit for this reason.

Theorem 2.30. *Let two conics be given by the (3×3)-matrices $\mathbf{A}_H$ and $\mathbf{B}_H$ such that $\chi_{\mathbf{A},\mathbf{B}} \not\equiv 0$. If there is a real point $\mathbf{p}_H$ and $\lambda_0 \in \mathbb{R}$ such that $\mathbf{p}_H^T \mathbf{A}_H \mathbf{p}_H = \mathbf{p}_H^T \mathbf{B}_H \mathbf{p}_H = 0$ and $\mathbf{A}_H \mathbf{p}_H = -\lambda_0 \mathbf{B}_H \mathbf{p}_H \neq \mathbf{0}$, then λ_0 is a multiple root of $\chi_{\mathbf{A},\mathbf{B}}$.*

The definition of $\chi_{\mathbf{A},\mathbf{B}}$ for (3×3)-matrices is analogous to the definition in the case of (4×4)-matrices.

2.4.4 Rotation Matrices and Quaternions

Rotation matrices are a very common way to describe rotations by a certain angle about a certain axis. Alternatively, one can use the so-called quaternions. Since we will use quaternions in chapter 4 and in section 2.5, we will define them here and show how they are related to rotation matrices.

Rotation matrices can be defined as follows.

Definition 2.31. *We call a matrix $\mathbf{R} \in \mathbb{R}^{3\times 3}$ a rotation matrix, if*

1. $\mathbf{R}\mathbf{R}^T = \mathbf{E}$ *and*

2. $\det \mathbf{R} = 1$.

This means that the rows as well as the columns are pairwise orthogonal unit vectors that form a right-handed system. A point $\mathbf{p} \in \mathbb{R}^3$ is rotated with the rotation matrix $\mathbf{R}$ by multiplication from the left, i.e. the rotated point has coordinates $\mathbf{R}\mathbf{p}$. It is well known from algebra that transformations with a rotation matrix preserve lengths, orientations and angles. Let λ be a real eigenvalue of the rotation matrix $\mathbf{R}$ and let

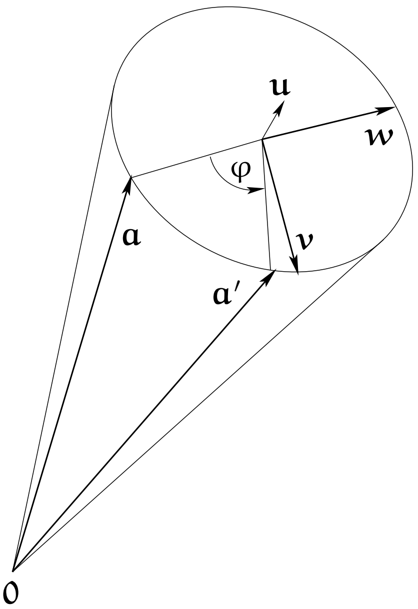


Figure 2.4: Rotation of a point $\mathbf{a}$ about an axis $\mathbf{u}$ by the angle φ.

$\mathbf{u}$ be a unit eigenvector with respect to λ. By the definition of $\mathbf{R}$ it holds that $1 = \mathbf{u}^2 = (\mathbf{R}\mathbf{u})^2 = \lambda^2$ and hence, $|\lambda| = 1$. The characteristic polynomial of $\mathbf{R}$ has the form $\chi_{\mathbf{R}}(\lambda) = -\lambda^3 + \ldots + 1$. Obviously, $\chi_{\mathbf{R}}(0) = 1$ and $\lim_{\lambda\to\infty} \chi_{\mathbf{R}}(\lambda) = -\infty$. Therefore, $\lambda = 1$ must be an eigenvalue of $\mathbf{R}$. The unit eigenvector $\mathbf{u}$ with respect to the eigenvalue 1 is the axis of rotation. Because of the preservation of angles, every vector which is perpendicular to $\mathbf{u}$ is rotated by $\mathbf{R}$ about $\mathbf{u}$ by the same angle φ. One way to determine φ is the following. Choose an arbitrary unit vector $\mathbf{v}$ perpendicular to $\mathbf{u}$. Then, $\cos\varphi = \mathbf{v}^\mathsf{T}\mathbf{R}\mathbf{v}$ and $\sin\varphi = \mathbf{u}^\mathsf{T}(\mathbf{v} \times (\mathbf{R}\mathbf{v}))$.

Clearly, the matrix that describes a rotation about $\mathbf{u}$ by the angle φ is uniquely determined by $\mathbf{u}$ and φ. Therefore, we denote this matrix by $\mathbf{R}_{\mathbf{u},\varphi}$. We will now show how $\mathbf{R}_{\mathbf{u},\varphi}$ can be computed if $\mathbf{u}$ and φ are given. Let therefore $\mathbf{a}$ be an arbitrary point. With $\mathbf{a}'$ we denote the point that is obtained by rotating $\mathbf{a}$ about $\mathbf{u}$ by the angle φ. As sketched in figure 2.4, $\mathbf{a}$ and $\mathbf{a}'$ lie on a circular cone whose axis is given by the vector $\mathbf{u}$. The vectors $\mathbf{v}$ and $\mathbf{w}$ are defined as $\mathbf{v} = \mathbf{u} \times \mathbf{a}$ and $\mathbf{w} = \mathbf{u} \times \mathbf{v}$. We observe that $\mathbf{a}' = \mathbf{a} + (1 - \cos\varphi)\mathbf{w} + \sin\varphi\mathbf{v}$. We write this in the form

$$\begin{aligned} \mathbf{a}' &= \mathbf{a} + (1 - \cos\varphi)\big(\mathbf{u} \times (\mathbf{u} \times \mathbf{a})\big) + \sin\varphi(\mathbf{u} \times \mathbf{a}) \\ &= \big(\mathbf{E} + (1 - \cos\varphi)(\mathbf{u}\mathbf{u}^\mathsf{T} - \mathbf{E}) + \sin\varphi\mathbf{u}^\times\big)\mathbf{a}. \end{aligned}$$

Since $\mathbf{a}$ was chosen arbitrarily, we conclude that

$$\mathbf{R}_{\mathbf{u},\varphi} = \cos\varphi\mathbf{E} + (1 - \cos\varphi)\mathbf{u}\mathbf{u}^\mathsf{T} + \sin\varphi\mathbf{u}^\times.$$

This equation is known as *Rodrigues' formula.* Starting with this formula, we show how the axis $\mathbf{u}$ and the angle φ can be computed if $\mathbf{R}_{\mathbf{u},\varphi}$ is given. We immediately verify that

$$\cos\varphi = \frac{\operatorname{tr}(\mathbf{R}) - 1}{2}.$$

Furthermore, we observe that $\mathbf{R} - \mathbf{R}^\mathsf{T} = 2\sin\varphi\mathbf{u}^\times$, which we write as $\mathbf{r}^\times$. We make a case distinction whether $\mathbf{r}$ is the zero vector. If $\mathbf{r} \neq 0$, then we set $\mathbf{u} = \mathbf{r}/|\mathbf{r}|$ which implies $\sin\varphi = |\mathbf{r}|/2$. If $\mathbf{r} = 0$, then $\sin\varphi = 0$. If $\cos\varphi = 1$, then the angle of rotation is zero and we can choose $\mathbf{u}$ arbitrarily. If $\cos\varphi = -1$, then $\mathbf{u}\mathbf{u}^\mathsf{T} = \frac{1}{2}(\mathbf{R} + \mathbf{E})$. This determines $\mathbf{u}$ up to the sign. But since $\varphi = \pi$, the sign of $\mathbf{u}$ does not matter.

Now, we introduce the quaternions and show how they can be used to describe rotations.

Definition 2.32. *Let $p_0, q_0 \in \mathbb{R}$ and $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$ and set $\mathfrak{p} = [p_0, \mathbf{p}^\mathsf{T}]^\mathsf{T}$ and $\mathfrak{q} = [q_0, \mathbf{q}^\mathsf{T}]^\mathsf{T}$. We call the operation defined on $\mathbb{R}^4$ by*

$$\mathfrak{p} \cdot \mathfrak{q} = \begin{bmatrix} p_0 q_0 - \mathbf{p}^\mathsf{T}\mathbf{q} \\ p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q} \end{bmatrix}$$

the quaternion product. The set $\mathbb{R}^4$ together with the component-wise sum and the quaternion product is called the set of quaternions. The quaternion $\mathfrak{q}^* = [q_0, -\mathbf{q}^\mathsf{T}]^\mathsf{T}$ is called the conjugate of $\mathfrak{q}$. The length of a quaternion is its length as a vector in $\mathbb{R}^4$.

It is not hard to verify that the quaternions form a division ring with the one-element $[1, 0, 0, 0]^\mathsf{T}$. The multiplicative inverse of $\mathfrak{q}$ is $\mathfrak{q}^*/|\mathfrak{q}|$. We call a quaternion of length one a unit quaternion. For a vector $\mathbf{a} \in \mathbb{R}^3$ we write $\mathfrak{a} = [0, \mathbf{a}^\mathsf{T}]^\mathsf{T}$. The following lemma shows how quaternions can be used to describe rotations.

Lemma 2.33. *Let $\mathbf{a} \in \mathbb{R}^3$ be a point and let $\mathbf{u} \in \mathbb{R}^3$ be a unit vector. Further, let $\varphi \in \mathbb{R}$ be an angle. Define the unit quaternion $\mathfrak{q}$ by $q_0 = \cos\frac{\varphi}{2}$ and $\mathbf{q} = \sin\frac{\varphi}{2}\mathbf{u}$. Then,*

$$\mathfrak{q}\mathfrak{a}\mathfrak{q}^* = \begin{bmatrix} 0 \\ \mathbf{R}_{\mathbf{u},\varphi}\mathbf{a} \end{bmatrix}.$$

Proof. Let $\mathfrak{a}' = \mathfrak{q}\mathfrak{a}\mathfrak{q}^*$. By applying the definition of the quaternion product twice, we find $a_0' = 0$ and

$$\begin{aligned} \mathbf{a}' &= (q_0^2 - \mathbf{q}^2)\mathbf{a} + 2\mathbf{q}\mathbf{q}^\mathsf{T}\mathbf{a} + 2q_0(\mathbf{q} \times \mathbf{a}) \\ &= \underbrace{\left(\cos^2\frac{\varphi}{2} - \sin^2\frac{\varphi}{2}\right)}_{\cos\varphi}\mathbf{a} + \underbrace{2\sin^2\frac{\varphi}{2}}_{1-\cos\varphi}\mathbf{u}\mathbf{u}^\mathsf{T}\mathbf{a} + \underbrace{2\sin\frac{\varphi}{2}\cos\frac{\varphi}{2}}_{\sin\varphi}(\mathbf{u} \times \mathbf{a}) \\ &= \mathbf{R}_{\mathbf{u},\varphi}\mathbf{a} \end{aligned}$$

by Rodrigues' formula. □

In this way, every rotation can be expressed by a unit quaternion. Conversely, each unit quaternion defines a rotation, which we also state as a lemma.

Lemma 2.34. *Let* $\mathbf{q}$ *be a unit quaternion. Then there is an axis* $\mathbf{u}$ *and an angle* φ *such that for each point* $\mathbf{a}$ *it holds that*

$$\mathbf{q}\mathbf{a}\mathbf{q}^* = \begin{bmatrix} 0 \\ \mathbf{R}_{\mathbf{u},\varphi}\mathbf{a} \end{bmatrix}.$$

Proof. Set $\varphi = 2\arccos q_0$ and

$$\mathbf{u} = \begin{cases} \frac{\mathbf{q}}{|\mathbf{q}|} & \text{if } |\mathbf{q}| \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, $q_0 = \cos\frac{\varphi}{2}$ and $\mathbf{q} = \sin\frac{\varphi}{2}\mathbf{u}$. Hence, the claim follows by lemma 2.33. □

2.4.5 Solving Polynomial Equations

In chapter 3 we will reduce a number of problems arising in the static as well as the dynamic collision detection to polynomial equations. These equations will always be given in one of the following two forms.

1. In the power representation, the coefficients of the polynomial are given explicitly, i.e. the equation has the form

$$p(x) = \sum_{i=0}^{d} a_i x^i = 0. \tag{2.17}$$

2. In the matrix representation, the polynomial is given as the determinant of a matrix polynomial, i.e. there is a matrix $\mathbf{M}(x) \in \mathbb{R}[x]^{n\times n}$ such that the equation has the form

$$p(x) = \det \mathbf{M}(x) = 0.$$

   Equivalently, this can be expressed in the form

$$\det\left(\sum_{i=0}^{d} \mathbf{M}_i x^i\right) = 0, \tag{2.18}$$

   where $\mathbf{M}_i \in \mathbb{R}^{n\times n}$ for $i = 0, \dots, d$.

The second form often arises if the coefficients of the resultant of two bivariate polynomials are too complex to be computed symbolically. Then, the matrix polynomial $\mathbf{M}(x)$ is just the Sylvester matrix of the two polynomials with respect to one variable. In this section, we will describe methods to solve suchlike equations efficiently.

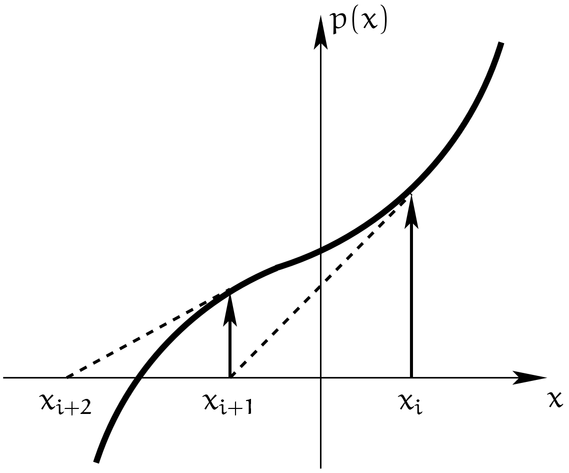


Figure 2.5: The Newton-Raphson method finds the next estimate of the root by intersecting the local tangent line with the x-axis.

Polynomials in power representation

Let the polynomial $p(x)$ be given as in (2.17). If the degree d is less than or equal to four, then the roots can be computed in closed form. For $d = 1$ and $d = 2$ this is obvious and for $d = 3$ and $d = 4$ we use Cardano's and Ferrari's formulae. Since it is a well-known result from algebra that there is no general formula for the roots of polynomials of degree greater than four, we are forced to use numerical methods. We describe local methods such as the Newton-Raphson method and Laguerre's method and how they can be used to find all roots of $p(x)$ using deflation (see also [PTVF94]). For the Newton-Raphson method we additionally describe how interval arithmetic can be used to achieve global convergence. Moreover, we describe the eigenvalue method which is a globally convergent approach to find all roots and can also be found in [PTVF94].

The Newton-Raphson Method This method works as follows. Given a guess x_i for a root of p, the tangent line in the point $[x_i, p(x_i)]^T$ is intersected with the x-axis. The next guess x_{i+1} is this intersection value. This is illustrated in figure 2.5. To be a bit more formal, we consider the Taylor series expansion in the neighbourhood of the point x_i, which we write as

$$p(x_i + \delta) = p(x_i) + \delta p'(x_i) + \delta^2 \frac{p''(x_i)}{2} + \ldots.$$

For very small values of δ, the terms involving δ^k for $k \geq 2$ are unimportant. Hence, $p(x_i + \delta) = 0$ implies $\delta = -p(x_i)/p'(x_i)$. Thus, the point x_{i+1} is defined as

$$x_{i+1} = x_i - \frac{p(x_i)}{p'(x_i)}. \tag{2.19}$$

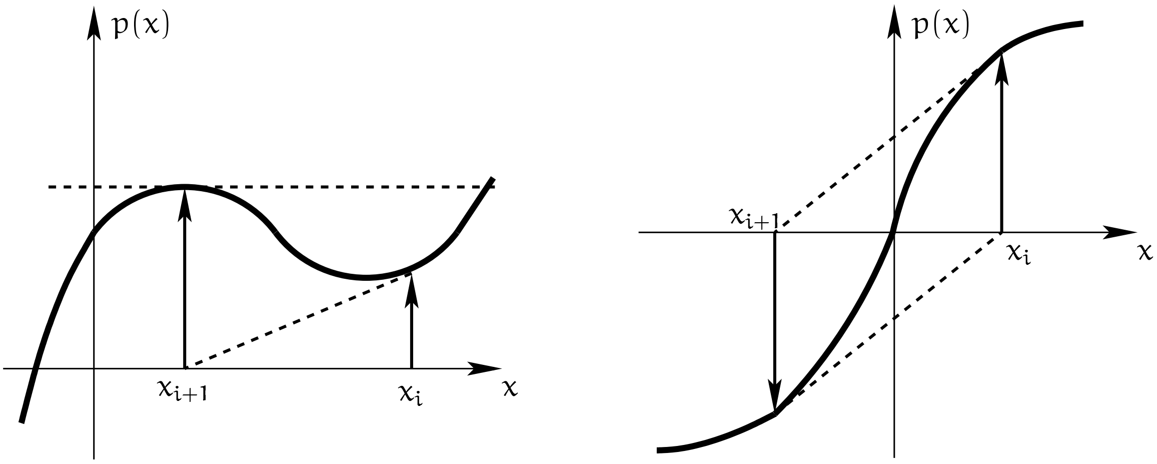


Figure 2.6: Two situations where the Newton-Raphson method runs into problems.

The method starts with an initial guess x_0 and improves this value in each iteration by taking the Newton step (2.19). This is repeated until the value $|x_i - x_{i+1}|$ falls below a given threshold. Then the value x_{i+1} is returned.

One nice fact about this method is that it converges quadratically, provided the initial guess is sufficiently close to a root. Quadratic convergence means that the distance ε_{i+1} between the estimate x_{i+1} and the desired root $\tilde{x}$ is a constant times ε_i^2. That the Newton-Raphson method has this property can be seen as follows. We define $\varepsilon_i = x_i - \tilde{x}$. If ε_i is very small, $p(x_i)$ and $p'(x_i)$ are approximately

$$\begin{aligned} p(\tilde{x} + \varepsilon_i) &\approx \varepsilon_i p'(\tilde{x}) + \varepsilon_i^2 \frac{p''(\tilde{x})}{2}, \\ p'(\tilde{x} + \varepsilon_i) &\approx p'(\tilde{x}) + \varepsilon_i p''(\tilde{x}). \end{aligned}$$

From (2.19) follows that

$$\varepsilon_{i+1} = \varepsilon_i - \frac{p(x_i)}{p'(x_i)}.$$

Inserting the above two expressions for $p(x_i)$ and $p'(x_i)$, we obtain

$$\varepsilon_{i+1} \approx \frac{1}{2}\varepsilon_i^2 \frac{p''(\tilde{x})}{p'(\tilde{x}) + \varepsilon_i p''(\tilde{x})} \approx \varepsilon_i^2 \frac{p''(\tilde{x})}{2p'(\tilde{x})}.$$

This means that the number of significant digits approximately doubles in each iteration. A fact which is not so nice is that one can run into problems if the initial guess is *not* sufficiently close to a root. Figure 2.6 shows two such situations. In the left-hand picture, the iteration reaches an extremal point. Since the tangent line in this point does not intersect the x-axis, the method will fail in this situation. In the right-hand picture, the method will run into an infinite loop, since $x_{i+2} = x_i$.

The Newton-Raphson method is very good if one knows good approximations of the roots in advance. Therefore, it can be used to "polish up" the results given by other methods.

Laguerre's Method This method behaves very well in practice, although not much is theoretically proved about its convergence for polynomials that have real as well as complex roots. For polynomials with only real roots it is guaranteed to converge to one of them from any starting point. But even in the presence of complex roots experiments suggest that it is extremely unusual that the method does not converge. If it converges to a single complex root it is known that its convergence is third order, which means that the number of significant digits approximately triples in each iteration. In the following we want to motivate how this method works. This motivation is taken from [PTVF94] and is not to be taken as a derivation of the approach.

If we denote the d roots of our polynomial p by $\xi_1, \dots \xi_d$, then we can write

$$\begin{aligned} p(x) &= a_d \prod_{i=1}^{d} (x - \xi_i) \text{ and hence} \\ \ln|p(x)| &= \ln|a_d| + \sum_{i=1}^{d} \ln|x - \xi_i|. \end{aligned}$$

If we derivate the second of these equations twice with respect to x we get

$$G(x) = \frac{d \ln|p(x)|}{dx} = \sum_{i=1}^{d} \frac{1}{x - \xi_i} = \frac{p'(x)}{p(x)} \quad \text{and} \tag{2.20}$$

$$H(x) = -\frac{d^2 \ln|p(x)|}{dx^2} = \sum_{i=1}^{d} \frac{1}{(x - \xi_i)^2} = \left(\frac{p'(x)}{p(x)}\right)^2 - \frac{p''(x)}{p(x)}. \tag{2.21}$$

Now, a set of very rigorous assumptions is made. We assume that the root ξ_1 that we are looking for is located at distance a from our estimate x_i, i.e. $x_i - \xi_1 = a$ and all other roots $\xi_2, \dots \xi_d$ are located at distance b from x_i, i.e. $x_i - \xi_j = b$ for $j = 2, \dots, d$. Then $G(x_i)$ and $H(x_i)$ are

$$\begin{aligned} G(x_i) &= \frac{1}{a} + \frac{d-1}{b} \quad \text{and} \\ H(x_i) &= \frac{1}{a^2} + \frac{d-1}{b^2}. \end{aligned}$$

If we solve the first equation for b, insert the result into the second equation and solve this for a, then we obtain after some transformations

$$a = \frac{d}{G(x_i) \pm \sqrt{(d-1)(dH(x_i) - G(x_i)^2)}}. \tag{2.22}$$

The method works as follows. For a guess x_i, compute $G(x_i)$ and $H(x_i)$ using the right-most expressions in (2.20) and (2.21). Then compute a by formula (2.22). For stability reasons, the sign in this formula should be chosen such that the absolute value of the denominator is maximized. The next estimate is computed as $x_{i+1} = x_i - a$. This is repeated until the distance between two successive estimates falls below a given threshold. Since the term inside the square root can be negative, this method requires complex arithmetic even when converging towards a real root.

Both the Newton-Raphson method and Laguerre's method find only one root of the polynomial p. But generally, we are interested in *all* roots. Therefore, we *deflate* the polynomial whenever we have computed a root ξ. This means we factor out the linear factor $x - \xi$ in the case that ξ is real and $(x - \xi)(x - \overline{\xi})$ if ξ is complex. After the deflation we compute the next root. We stop as soon as the result of the deflation is constant. If we use the Newton-Raphson method we must not choose the initial guesses on the real axis, since the method would never get off of that axis.

If we use this strategy with floating point arithmetic, we must be careful. Since each root is computed only finitely accurate, errors accumulate in the deflation processes. Thus, at the end of the computation, the roots must be "polished". This means we consider the values that we have computed only as good guesses for the roots and use them as starting values for the Newton-Raphson or Laguerre's method with the original, non-deflated polynomial.

Newton-Raphson with Interval Arithmetic Now we show how interval arithmetic can be used to enable the Newton-Raphson method to find all real roots of a polynomial. In fact, it can find all real roots of any continuously differentiable function $f(x)$ in a given interval if the derivative of f is known. This method can e.g. be found in [HHKR95].

The crucial idea is the following lemma.

Lemma 2.35. *Let $f : \mathbb{R} \to \mathbb{R}$ be continously differentiable and let $\mathcal{I} = [a, b] \subset \mathbb{R}$ be an interval. Moreover, assume that $0 \notin f'(\mathcal{I})$. Let $m = (a + b)/2$ be the midpoint of $\mathcal{I}$. Then, if there is a value $x^* \in \mathcal{I}$ such that $f(x^*) = 0$, then $x^* \in N(\mathcal{I})$, where*

$$N(\mathcal{I}) = m - \frac{f(m)}{f'(\mathcal{I})}. \tag{2.23}$$

Proof. If $x^* = m$, then $N(\mathcal{I}) = [m, m]$ and we are done. Hence, we assume that $x^* \neq m$. Then, by the mean value theorem, there is a value ξ between x^* and m such that

$$f(m) - f(x^*) = f'(\xi)(m - x^*).$$

Since x^* is a root of f this is can be transformed into

$$x^* = m - \frac{f(m)}{f'(\xi)} \in N(\mathcal{I}).$$

□

A simple corollary of this is the following.

Corollary 2.36. *If $N(\mathcal{I}) \cap \mathcal{I} = \emptyset$, then f has no root in $\mathcal{I}$.*

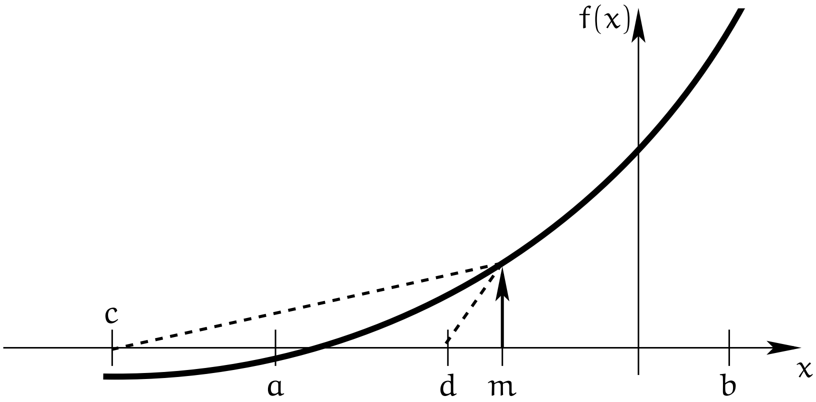


Figure 2.7: An iteration step in the Newton-Raphson method with interval arithmetic, if $0 \notin f'(\mathcal{I})$. $\mathcal{I} = [a, b]$ and $N(\mathcal{I}) = [c, d]$.

Given an interval $\mathcal{I}$ in which we want to find the roots, the method works as follows. If $0 \notin f'(\mathcal{I})$ we define $\tilde{\mathcal{I}} = N(\mathcal{I}) \cap \mathcal{I}$. If $\tilde{\mathcal{I}} \subsetneq \mathcal{I}$, then the next iteration step is applied to $\tilde{\mathcal{I}}$. This is illustrated in figure 2.7. If either $0 \in f'(\mathcal{I})$ or $\tilde{\mathcal{I}} = \mathcal{I}$, we perform a bisection step: Let m be the midpoint of $\mathcal{I} = [a, b]$. Then the next iteration step must be applied to the two intervals $[a, m]$ and $[m, b]$. If the width of an interval falls below a given threshold then its midpoint is returned as a root. If during the iteration an interval $\mathcal{J}$ occurs, such that $0 \notin f'(\mathcal{J})$, then by [HHKR95] the method is guaranteed to converge quadratically on this interval. But if there exist multiple roots it can degenerate to a bisection method.

At the beginning of this paragraph we stated that for a polynomial this method can be used to find all real roots. We have only seen so far that it is able to compute all real roots in a given interval $\mathcal{I}$. This is sufficient in some cases. In section 3.3 we are often only interested in the roots lying in the interval $[0, 1]$. If we want to determine all real roots, then we can use the following lemma, which is proven in [Mig92] (theorem 4.2).

Lemma 2.37 (Root bounds). *Let the polynomial* p *be given as in (2.17). The coefficients* a_i *may be complex numbers. Let* ξ *be any (real or complex) root of* p. *Then, the following inequalities hold.*

$$\begin{aligned}
|\xi| &\leq \max\left\{1, \frac{1}{|a_d|}\sum_{i=0}^{d-1}|a_i|\right\}, \\
|\xi| &\leq 1 + \frac{1}{|a_d|}\max_{0\leq i<d}\{\,|a_i|\,\}, \\
|\xi| &\leq \frac{1}{|a_d|}\left(|a_0| + \sum_{i=1}^{d}|a_{i-1} - a_i|\right).
\end{aligned}$$

Let r be one of the root bounds given by this lemma. Then we can be sure that all real roots of p lie in the interval $\mathcal{I} = [-r, r]$. Hence, starting the interval method on $\mathcal{I}$, it will find all real roots of the polynomial.

The Eigenvalue Method This method is able to find all roots of a polynomial simultaneously. The problem is reduced to finding the eigenvalues of a matrix. Let the polynomial p be given in the form (2.17). We define the *companion matrix* of p as

$$\mathbf{C}_p = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -\frac{a_0}{a_d} & -\frac{a_1}{a_d} & -\frac{a_2}{a_d} & \dots & -\frac{a_{d-1}}{a_d} \end{bmatrix}.$$

The key observation of the method is the following lemma.

Lemma 2.38. *The characteristic polynomial of* $\mathbf{C}_p$ *is*

$$|\mathbf{C}_p - x\mathbf{E}| = \frac{(-1)^d}{a_d} p(x).$$

Proof. We prove this by induction on the degree d of p. For $d = 1$ the claim is trivial. So let us assume that the lemma holds for polynomials of degree $d - 1$ and let p be a polynomial of degree d. We can write

$$p(x) = xq(x) + a_0 \quad \text{with } q(x) = \sum_{i=0}^{d-1} a_{i+1} x^i.$$

Since the degree of q is $d - 1$ we know by induction hypothesis that

$$|\mathbf{C}_q - x\mathbf{E}| = \frac{(-1)^{d-1}}{a_d} q(x).$$

The companion matrix of p has the form

$$\mathbf{C}_p = \left[\begin{array}{cccc} 0 & 1 & \dots & 0 \\ \vdots & & & \\ 0 & & \boxed{\mathbf{C}_q} & \\ -a_0/a_d & & & \end{array}\right].$$

We expand the determinant $|\mathbf{C}_p - x\mathbf{E}|$ by cofactors of the first column and obtain

$$\begin{aligned} |\mathbf{C}_p - x\mathbf{E}| &= -x|\mathbf{C}_q - x\mathbf{E}| + (-1)^d \frac{a_0}{a_d} \\ &= \frac{(-1)^d}{a_d}\big(xq(x) + a_0\big) = \frac{(-1)^d}{a_d} p(x). \end{aligned}$$

□

By this lemma, the roots of p are exactly the eigenvalues of $\mathbf{C}_p$. Therefore, we can apply a numerical method such as the QR algorithm to the matrix $\mathbf{C}_p$. The QR method can exploit the upper Hessenberg form of $\mathbf{C}_p$, which reduces its running time to $O(d^2)$ per iteration as opposed to $O(d^3)$ for a general matrix.

Polynomials in Matrix Representation

Now, let the polynomial $p(x)$ be given in the form (2.18). We describe two methods to find the roots of p. The first method that we present is a reduction to a (generalized) eigenvalue problem and the second one is an interpolation approach.

Reduction to a Generalized Eigenvalue Problem The idea is similar to the eigenvalue approach for polynomials in power representation and can be found in [MK96].

The basic observation is that if $p(x) = 0$ then there must be a non-zero vector $\mathbf{y}$ such that $\mathbf{M}(x)\mathbf{y} = \mathbf{0}$. We write this in the form

$$\sum_{i=0}^{d} \mathbf{M}_i x^i \cdot \mathbf{y} = 0 \quad \Leftrightarrow \quad -\sum_{i=0}^{d-1} \mathbf{M}_i x^i \cdot \mathbf{y} \;=\; \mathbf{M}_d x^d \cdot \mathbf{y} \;=\; x\mathbf{M}_d x^{d-1} \cdot \mathbf{y}.$$

If we define $\mathbf{u}_i = x^i \cdot \mathbf{y}$ for $i = 0, \dots, d-1$ we can rewrite this in the form

$$-\sum_{i=0}^{d-1} \mathbf{M}_i \mathbf{u}_i \;=\; x \cdot \mathbf{M}_d \mathbf{u}_{d-1}. \tag{2.24}$$

We define the *generalized companion matrix* of a matrix polynomial $\mathbf{M}(x)$ of degree d as

$$\mathbf{C}_\mathbf{M} = \begin{bmatrix} \mathbf{0} & \mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{E} \\ -\mathbf{M}_0 & -\mathbf{M}_1 & -\mathbf{M}_2 & \dots & -\mathbf{M}_{d-1} \end{bmatrix}.$$

If $\det \mathbf{M}_d \neq 0$ we set $\tilde{\mathbf{M}}_i = \mathbf{M}_d^{-1}\mathbf{M}_i$ for $i = 0, \dots, d-1$ and define $\tilde{\mathbf{M}}(x) = x^d\mathbf{E} + \sum_{i=0}^{d-1} \tilde{\mathbf{M}}_i x^i$. Then we have the following theorem

Theorem 2.39. *The solutions of* $p(x) = 0$ *are exactly the eigenvalues of the matrix* $\mathbf{C}_{\tilde{\mathbf{M}}}$.

Proof. If x is a root of p then, by the above argument we can construct a non-zero vector $\mathbf{u} = [\mathbf{u}_0^T, \dots, \mathbf{u}_{d-1}^T]^T$ such that equation (2.24) holds, which is equivalent to

$$-\sum_{i=0}^{d-1} \tilde{\mathbf{M}}_i \mathbf{u}_i = x \cdot \mathbf{u}_{d-1}. \tag{2.25}$$

The vector $\mathbf{u}$ is an eigenvector of $\mathbf{C}_{\tilde{\mathbf{M}}}$ with respect to the eigenvalue x. This can be seen as follows. The first $d-1$ components of $\mathbf{C}_{\tilde{\mathbf{M}}}\mathbf{u}$ are $[\mathbf{u}_1^T, \dots, \mathbf{u}_{d-1}^T]^T = x \cdot [\mathbf{u}_0, \dots, \mathbf{u}_{d-2}]$. This equality holds because of the construction of the vector $\mathbf{u}$. The last component of $\mathbf{C}_{\tilde{\mathbf{M}}}\mathbf{u}$ is equal to $x \cdot \mathbf{u}_{d-1}$ by equation (2.25).

Conversely, let x be an eigenvalue of $\mathbf{C}_{\tilde{\mathbf{M}}}$. Then there is a non-zero vector $\mathbf{u}$ such that $\mathbf{C}_{\tilde{\mathbf{M}}}\mathbf{u} = x \cdot \mathbf{u}$. We write this vector as $\mathbf{u} = [\mathbf{u}_0^T, \dots, \mathbf{u}_{d-1}^T]^T$. By the first $d-1$ of these equalities it holds for $i = 1, \dots, d-1$ that $\mathbf{u}_i = x \cdot \mathbf{u}_{i-1}$ and hence $\mathbf{u}_i = x^i \cdot \mathbf{u}_0$. Thus, the vector $\mathbf{u}_0$ cannot be zero and we set $\mathbf{y} = \mathbf{u}_0$. The last equality in $\mathbf{C}_{\tilde{\mathbf{M}}}\mathbf{u} = x \cdot \mathbf{u}$ is just equation (2.25) which, with this definition of $\mathbf{y}$ is equivalent to

$$\sum_{i=0}^{d} \mathbf{M}_i x^i \cdot \mathbf{y} = 0.$$

Since $\mathbf{y} \neq 0$ it holds that $\det \mathbf{M}(x) = 0$. □

If the determinant of $\mathbf{M}_d$ is zero but $\det \mathbf{M}_0 \neq 0$, then we can also reduce the problem to an eigenvalue problem by defining the polynomial $q(x) = \det \mathbf{N}(x)$ with $\mathbf{N}_i = \mathbf{M}_{d-i}$ for $i = 0, \dots, d$. Obviously, $q(x) = \det\left(x^d \cdot \mathbf{M}(x^{-1})\right)$. Hence, for $x \neq 0$ it holds that $q(x) = 0$ if and only if $p(x^{-1}) = 0$. We can find the roots of q by computing the eigenvalues of $\mathbf{C}_{\tilde{\mathbf{N}}}$ using e.g. the QR-algorithm.

If both $\det \mathbf{M}_d$ and $\det \mathbf{M}_0$ are zero, then we define the $(nd \times nd)$-matrix $\mathbf{D}_\mathbf{M} = \operatorname{diag}(\mathbf{E}, \dots, \mathbf{E}, \mathbf{M}_d)$. Then, the following theorem holds.

Theorem 2.40. *The solutions of* $p(x) = 0$ *are exactly the solutions of the generalized eigenvalue problem* $\mathbf{C}_\mathbf{M}\mathbf{u} = x \cdot \mathbf{D}_\mathbf{M}\mathbf{u}$.

The proof of this theorem works analogously to the proof of theorem 2.39.

We briefly describe an approach from [Man94] that uses inverse power iterations to find all real solutions of such a generalized eigenvalue problem. First, we notice that x is a solution of the problem stated in theorem 2.40 if and only if $f(x) = \det(\mathbf{C}_\mathbf{M} - x\mathbf{D}_\mathbf{M}) = 0$. The approach that we describe is able to find all real solutions in a given interval $[a, b]$. In order to find all real solutions the problem is decomposed into two parts. First, by substituting $x = \tilde{x}/(1-\tilde{x})$ we reduce the problem of finding all solutions $x \in [0, \infty)$ to finding all solutions $\tilde{x} \in [0, 1)$ of $g_1(\tilde{x}) = \det\left(\mathbf{C}_\mathbf{M} - \tilde{x}(\mathbf{C}_\mathbf{M} + \mathbf{D}_\mathbf{M})\right) = 0$. Then, by substituting $x = \tilde{x}/(1+\tilde{x})$ we reduce the problem of finding all solutions $x \in (-\infty, 0]$ to finding all solutions $\tilde{x} \in (-1, 0]$ of $g_2(\tilde{x}) = \det\left(\mathbf{C}_\mathbf{M} - \tilde{x}(\mathbf{D}_\mathbf{M} - \mathbf{C}_\mathbf{M})\right) = 0$.

Let $\mathbf{C}, \mathbf{D} \in \mathbb{R}^m$. In order to find all solutions of $g(x) = \det(\mathbf{C} - x\mathbf{D}) = 0$ in the interval $[a, b]$, the procedure produces a sequence of vectors $\mathbf{u}_k \in \mathbb{C}^m$ and a sequence of complex numbers x_k in the following way. Given an initial guess x^* and a random vector $\mathbf{u}_0 \in \mathbb{C}^m$ we set

$$\begin{aligned} \mathbf{z}_k &\leftarrow \text{solution of } (\mathbf{C} - x^*\mathbf{D})\mathbf{z}_k = \mathbf{D}\mathbf{u}_{k-1}, \qquad (2.26)\\ \mathbf{u}_k &= \frac{\mathbf{z}_k}{|\mathbf{z}_k|},\\ x_k &= \frac{\mathbf{u}_k^T\mathbf{C}\mathbf{u}_k}{\mathbf{u}_k^T\mathbf{D}\mathbf{u}_k} \end{aligned}$$

for $k = 1, 2, \ldots$. If $\mathbf{u}_k$ converges towards a vector $\mathbf{u}$, then $\mathbf{z}_k$ converges towards a vector $\mathbf{z} = \lambda\mathbf{u}$ for some value of λ. By (2.26) we have

$$\begin{aligned} \lambda(\mathbf{C} - x^*\mathbf{D})\mathbf{u} &= \mathbf{D}\mathbf{u} \\ \Leftrightarrow \quad \left(\mathbf{C} - \frac{\lambda x^* + 1}{\lambda}\mathbf{D}\right)\mathbf{u} &= 0. \end{aligned}$$

Hence, x_k converges towards to the solution $x = x^* + (1/\lambda)$ of $g(x) = 0$. The linear system (2.26) can be efficiently solved using e.g. LU-decomposition.

According to [Man94] this iteration converges towards the complex solution x_1 of $g(x) = 0$ that has smallest distance from the initial guess x^*. Hence, there is no further solution in the open disc centered at x^* with radius $R = |x^* - x_1|$. Thus, the following recursive procedure is proposed. Choose $x^* = (a + b)/2$. If x_1 lies in $[a, b]$, then add this value to the list of the solutions. Recursively apply the procedure to

- $[a, x^* - R]$, if $x^* - R \geq a$ and
- $[x^* + R, b]$, if $x^* + R \leq b$.

In this way, all real solutions in the interval $[a, b]$ can be computed.

Interpolation A different way to attack the problem of finding the roots of a polynomial in matrix representation is to determine its power representation using interpolation. Let l be an upper bound for the degree of $p(x)$. Further, let (x_i, y_i) for $i = 0, \ldots, l$ be sample points fulfilling $p(x_i) = y_i$. Then, the coefficients of the power representation of $p(x)$ are uniquely determined by these points. If l is exactly the degree of $p(x)$, then methods such as Lagrange's formula or Neville's algorithm can be used to find the coefficients (see [PTVF94]). However, higher accuracy can be expected if the polynomial is "oversampled", i.e. l is much greater than the degree of $p(x)$. A very common method to compute the coefficients is *fitting by linear least squares*, which we will describe in the following. This approach can also be found in [PTVF94].

We assume that we know the degree m of $p(x)$ and that $l > m$. Further we assume that we have computed the sample points (x_i, y_i). We define a penalty function f that depends on the coefficients a_i of $p(x)$ by

$$f(a_0, \ldots, a_m) = \sum_{j=0}^{l} \left(y_j - \sum_{i=0}^{m} a_i x_j^i \right)^2 .$$

Our goal is to minimize this function. We do so by setting the partial derivatives of f with respect to a_i equal to zero, which yields for $k = 0, \ldots, m$

$$\frac{\partial f(a_0, \ldots, a_m)}{\partial a_k} = 0 \quad \Leftrightarrow \quad \sum_{j=0}^{l} \left(y_j - \sum_{i=0}^{m} a_i x_j^i \right) x_j^k = 0.$$

Changing the order of summation we find that this is equivalent to

$$\sum_{i=0}^{m} \sum_{j=0}^{l} x_j^{k+i} a_i = \sum_{j=0}^{l} y_j x_j^k, \quad k = 0, \ldots, m. \tag{2.27}$$

We write $\mathbf{a} = [a_0, \ldots, a_m]^T$, $\mathbf{b} = [b_0, \ldots, b_m]^T$ and $\mathbf{A} = (\alpha_{ik})_{i,k=0}^m$ with

$$\begin{aligned} b_k &= \sum_{j=0}^{l} y_j x_j^k \quad \text{and} \\ \alpha_{ik} &= \sum_{j=0}^{l} x_j^{k+i}. \end{aligned}$$

With this, (2.27) can be rewritten as $\mathbf{A}\mathbf{a} = \mathbf{b}$. This is a system of linear equations in $a_0, \ldots, a_m$ and can e.g. be solved with LU-decomposition.

2.5 Physical Properties of Rigid Objects

In the following chapters we are dealing with moving rigid objects and the simulation of their dynamic behaviour. Thus, we must associate physical properties with each object such as a mass, a center of mass and moments of inertia. In this section we will define these properties for general rigid objects and give the equations that describe the motion of a rigid object in the presence of forces. Additionally, we will describe how the above mentioned physical characteristica can be computed for the kinds of objects that we consider in this thesis.

We always assume that the density function of any object $\mathcal{O}$ is a constant ρ. Hence, the mass of $\mathcal{O}$ is equal to $m = \rho V$, where V is the volume of $\mathcal{O}$. The volume is defined as

$$V = \iiint_{\mathcal{O}} d\mathbf{x}. \tag{2.28}$$

The center of mass of a finite number of particles with coordinates $\mathbf{x}_i$ and masses m_i is the weighted average $\sum m_i \mathbf{x}_i / \sum m_i$. For a continuous object $\mathcal{O}$ this generalizes to

$$\mathbf{c} = \frac{1}{m} \iiint_{\mathcal{O}} \mathbf{x} dm = \frac{\rho}{m} \iiint_{\mathcal{O}} \mathbf{x} d\mathbf{x}. \tag{2.29}$$

In order to describe rotational movements we need a relation between the angular momentum $\mathbf{L}$ and the angular velocity $\boldsymbol{\omega}$. The angular momentum $\mathbf{l}_i$ of a single particle with mass m_i located at $\mathbf{x}_i$ that rotates around a center $\mathbf{c}$ is $\mathbf{l}_i = \mathbf{r}_i \times \mathbf{p}_i$, where $\mathbf{r}_i = \mathbf{x}_i - \mathbf{c}$ is the vector pointing from the center of rotation to the particle and $\mathbf{p}_i = m_i \boldsymbol{\omega} \times \mathbf{r}_i$ is the linear momentum of the particle. The angular momentum of a finite set of such particles rotating around the same center with the same angular velocity is the sum $\sum l_i$. For a continuous object $\mathcal{O}$ rotating around its center of mass this generalizes to

$$\mathbf{L} = \iiint_{\mathcal{O}} \mathbf{r} \times (\boldsymbol{\omega} \times \mathbf{r}) dm = \rho \iiint_{\mathcal{O}} (\mathbf{r}^2 \mathbf{E} - \mathbf{r}\mathbf{r}^T) d\mathbf{r} \cdot \boldsymbol{\omega} = \mathbf{I}\boldsymbol{\omega}. \tag{2.30}$$

The matrix $\mathbf{I}$ is called the inertia matrix of $\mathcal{O}$. It is symmetric, since the integrand function is symmetric. Moreover, $\mathbf{I}$ is positive definite, since for any non-zero vector $\mathbf{a}$ it holds that

$$\mathbf{a}^T \mathbf{I} \mathbf{a} = \iiint_{\mathcal{O}} (\mathbf{r}^2 \mathbf{a}^2 - (\mathbf{r}^T \mathbf{a})^2) dm = \iiint_{\mathcal{O}} (\mathbf{r} \times \mathbf{a})^2 dm > 0.$$

The inequality is strict, since $\mathcal{O}$ is a 3-manifold and hence, there is at least one point $\mathbf{x} \in \mathcal{O}$ such that $(\mathbf{x} - \mathbf{c}) \times \mathbf{a} \neq 0$. Since $\mathbf{I}$ is symmetric and positive definite, it has three positive eigenvalues. These are called the principle moments of inertia of $\mathcal{O}$. If we rotate the object $\mathcal{O}$ with inertia matrix $\mathbf{I}$ around its center of mass with the rotation matrix $\mathbf{R}$, then it is easy to see from the definition of $\mathbf{I}$ that the inertia matrix of the rotated object is given by $\mathbf{R}\mathbf{I}\mathbf{R}^T$.

Before we describe how the integrals (2.28), (2.29) and (2.30) can be computed for an object of one of the classes defined earlier, we give the equations of motion. We assume that the coordinates of each point $\mathbf{p}$ associated with a moving object $\mathcal{O}$ at time t are given by

$$\mathbf{p}(t) = \mathbf{R}(t)(\tilde{\mathbf{p}} - \tilde{\mathbf{c}}) + \mathbf{c}(t), \tag{2.31}$$

where $\tilde{\mathbf{p}} = \mathbf{p}(0)$ and $\tilde{\mathbf{c}} = \mathbf{c}(0)$ is the center of mass. Then we can formulate the following theorem.

Theorem 2.41. *Let a moving object $\mathcal{O}$ be given by equation (2.31) and let $\mathbf{f}_1, \ldots, \mathbf{f}_k$ be the forces acting on $\mathcal{O}$. Let $\mathbf{r}_i$ for $i = 1, \ldots, k$ be the vector pointing from the center of mass $\mathbf{c}(t)$ to the point to which $\mathbf{f}_i$ is applied. Then the following equations of motion hold.*

$$\begin{aligned} \dot{\mathbf{c}} &= \mathbf{v}, \\ \dot{\mathbf{R}} &= \boldsymbol{\omega}^{\times} \mathbf{R}, \end{aligned} \tag{2.32}$$

and

$$\begin{aligned} m\dot{\mathbf{v}} &= \sum_{i=1}^{k} \mathbf{f}_i, \\ \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} &= \sum_{i=1}^{k} \mathbf{r}_i \times \mathbf{f}_i. \end{aligned} \tag{2.33}$$

The equations (2.33) are also known as the Newton-Euler dynamics equations.

Proof. The upper equation of (2.32) is just the definition of the linear velocity and the upper equation of (2.33) is Newton's second axiom.

In order to prove the lower equation of (2.32) we choose an arbitrary vector $\tilde{\mathbf{r}}$ and define $\mathbf{r}(t) = \mathbf{R}(t)\tilde{\mathbf{r}}$. This vector rotates with the same angular velocity as $\mathcal{O}$ and thus, it holds $\dot{\mathbf{r}}(t) = \boldsymbol{\omega} \times \mathbf{r}(t) = \boldsymbol{\omega} \times \mathbf{R}(t)\tilde{\mathbf{r}}$. On the other hand, derivating with respect to time yields $\dot{\mathbf{r}}(t) = \dot{\mathbf{R}}(t)\tilde{\mathbf{r}}$. The claim follows from the arbitrary choice of $\tilde{\mathbf{r}}$.

In order to prove the last equation, we consider the angular momentum $\mathbf{L}$. From (2.30) we know that $\mathbf{L} = \mathbf{I}\boldsymbol{\omega}$. Derivating this with respect to the time parameter we obtain $\mathbf{I}\dot{\boldsymbol{\omega}} + \dot{\mathbf{I}}\boldsymbol{\omega} = \dot{\mathbf{L}}$. The right-hand side of this is just the torque which is given by $\sum \mathbf{r}_i \times \mathbf{f}_i$. Thus, we have

$$\mathbf{I}\dot{\boldsymbol{\omega}} + \dot{\mathbf{I}}\boldsymbol{\omega} = \sum_{i=1}^{k} \mathbf{r}_i \times \mathbf{f}_i.$$

We complete the proof by computing $\dot{\mathbf{I}}$. Let $\tilde{\mathbf{I}} = \mathbf{I}(0)$. Then we have seen that $\mathbf{I}(t) = \mathbf{R}(t)\tilde{\mathbf{I}}\mathbf{R}(t)^T$. Derivating and using the lower equation of (2.32) yields

$$\dot{\mathbf{I}} = \boldsymbol{\omega}^{\times}\mathbf{R}\tilde{\mathbf{I}}\mathbf{R}^T + \mathbf{R}\tilde{\mathbf{I}}(\boldsymbol{\omega}^{\times}\mathbf{R})^T = \boldsymbol{\omega}^{\times}\mathbf{I} - \mathbf{I}\boldsymbol{\omega}^{\times}.$$

□

If we use a quaternion $\mathfrak{q}(t)$ instead of the rotation matrix $\mathbf{R}(t)$ to describe the rotation, then the lower equation of (2.32) is replaced according to the following lemma.

Lemma 2.42. *Let* $\mathfrak{q}(t)$ *be a unit quaternion defining a rotation with angular velocity* $\boldsymbol{\omega}$. *Then,*

$$\dot{\mathfrak{q}} = \frac{1}{2}\boldsymbol{\omega}\mathfrak{q}. \tag{2.34}$$

Proof. We choose an arbitrary vector $\tilde{\mathbf{r}}$ and define $\mathfrak{r}(t) = \mathfrak{q}(t)\tilde{\mathfrak{r}}\mathfrak{q}^*(t)$. Then, the vector $\mathbf{r}$ rotates with angular velocity $\boldsymbol{\omega}$ and it holds $\dot{\mathbf{r}} = \boldsymbol{\omega} \times \mathbf{r}$. Using quaternions, this can be written as $\dot{\mathfrak{r}} = \frac{1}{2}(\boldsymbol{\omega}\mathfrak{r} - \mathfrak{r}\boldsymbol{\omega}) = \frac{1}{2}(\boldsymbol{\omega}\mathfrak{q}\tilde{\mathfrak{r}}\mathfrak{q}^* - \mathfrak{q}\tilde{\mathfrak{r}}\mathfrak{q}^*\boldsymbol{\omega})$. On the other hand, we obtain $\dot{\mathfrak{r}} = \dot{\mathfrak{q}}\tilde{\mathfrak{r}}\mathfrak{q}^* + \mathfrak{q}\tilde{\mathfrak{r}}\dot{\mathfrak{q}}^*$. Because of $\boldsymbol{\omega}^* = -\boldsymbol{\omega}$ and $\mathfrak{r}^* = -\mathfrak{r}$ it follows that

$$(\dot{\mathfrak{q}} - \frac{1}{2}\boldsymbol{\omega}\mathfrak{q})\tilde{\mathfrak{r}}\mathfrak{q}^* = \mathfrak{q}\tilde{\mathfrak{r}}^*(\dot{\mathfrak{q}}^* - \frac{1}{2}\mathfrak{q}^*\boldsymbol{\omega}^*).$$

The quaternion on the right-hand side is the conjugate of the left-hand side, whose vector part must therefore be zero, i.e. this quaternion is a scalar α. Multiplying with $\mathfrak{q}^*$ from the left and with $\mathfrak{q}$ from the right and using $\mathfrak{q}\mathfrak{q}^* = 1$ we obtain $\mathfrak{q}^*(\dot{\mathfrak{q}} - \frac{1}{2}\boldsymbol{\omega}\mathfrak{q})\tilde{\mathfrak{r}} = \alpha$. We write this as $\mathfrak{p}\tilde{\mathfrak{r}} = \alpha$, which is equivalent to

$$\begin{bmatrix} -\mathbf{p}^T\tilde{\mathbf{r}} \\ (p_0\mathbf{E} + \mathbf{p}^{\times})\tilde{\mathbf{r}} \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}.$$

Since $\tilde{\mathbf{r}}$ was chosen arbitrarily, it follows that $p_0 \equiv 0$ and $\mathbf{p} \equiv \mathbf{0}$. But this implies that $\dot{\mathfrak{q}} - \frac{1}{2}\boldsymbol{\omega}\mathfrak{q} \equiv \mathfrak{o}$. □

Computation of the integrals (2.28), (2.29) and (2.30)

The following method to compute the volume integrals mentioned above can be found in [Imm01]. In the version that we present here we made some improvements that we will mention at the appropriate moment. The method is a generalization of [Mir96a], where these computations are performed for polyhedra.

The basic idea is to use Gauss' theorem (also known as the divergence theorem) to transfer the volume integrals into surface integrals, and then use Green's theorem to transform these into path integrals. These well known theorems can e.g. be found in [BSMM93].

Theorem 2.43 (Gauss' theorem). *Let $\mathcal{K}$ be a region in space with piecewise smooth boundary $\partial\mathcal{K}$, and let $\mathbf{V} = [P, Q, R]^T$ be a vector field which is C^1-continuous on $\mathcal{K}$. Then,*

$$\iiint_{\mathcal{K}} \operatorname{div}(\mathbf{V}) d\mathbf{x} = \iint_{\partial\mathcal{K}} P dx_2 dx_3 + Q dx_3 dx_1 + R dx_1 dx_2. \tag{2.35}$$

Our objects fulfill the preconditions of this theorem and the boundary of such an object $\mathcal{O}$ consists of finitely many smooth faces $\mathcal{F}_i$. Hence, the right-hand side of (2.35) becomes for $\mathcal{K} = \mathcal{O}$

$$\sum_i \iint_{\mathcal{F}_i} P dx_2 dx_3 + Q dx_3 dx_1 + R dx_1 dx_2.$$

Let the surface $\mathcal{S}_i$ containing $\mathcal{F}_i$ be parameterized by $\mathbf{r}_i(u, v)$ and let the parameter range of $\mathcal{F}$ be given by $\mathcal{U}_i \subset \mathbb{R}^2$. If we substitute $\mathbf{x}$ by $\mathbf{r}_i(u, v)$ then we have

$$dx_i dx_j = \left| \frac{\partial(r_i, r_j)}{\partial(u, v)} \right| du dv,$$

where $\partial(r_i, r_j)/\partial(u, v)$ is the Jacobian matrix. Using this, the ith addend in the above summation can be written as

$$\iint_{\mathcal{U}_i} \mathbf{V}(\mathbf{r}_i(u, v))^T \left(\frac{\partial \mathbf{r}_i}{\partial u}(u, v) \times \frac{\partial \mathbf{r}_i}{\partial v}(u, v) \right) du dv.$$

We denote the cross-product in this equation by $\mathbf{n}_i(u, v)$, since it is normal to $\mathcal{F}_i$ in the point $\mathbf{r}_i(u, v)$. We have to take care that the parameterization is chosen in such a way that these normals are outward-pointing. With this, (2.35) becomes

$$\iiint_{\mathcal{O}} \operatorname{div}(\mathbf{V}) d\mathbf{x} = \sum_i \iint_{\mathcal{U}_i} \mathbf{V}^T \mathbf{n}_i du dv. \tag{2.36}$$

In order to apply this to the integrals (2.28), (2.29) and (2.30) we need to find for each (component of the) integrand a vector field whose divergence is that function.

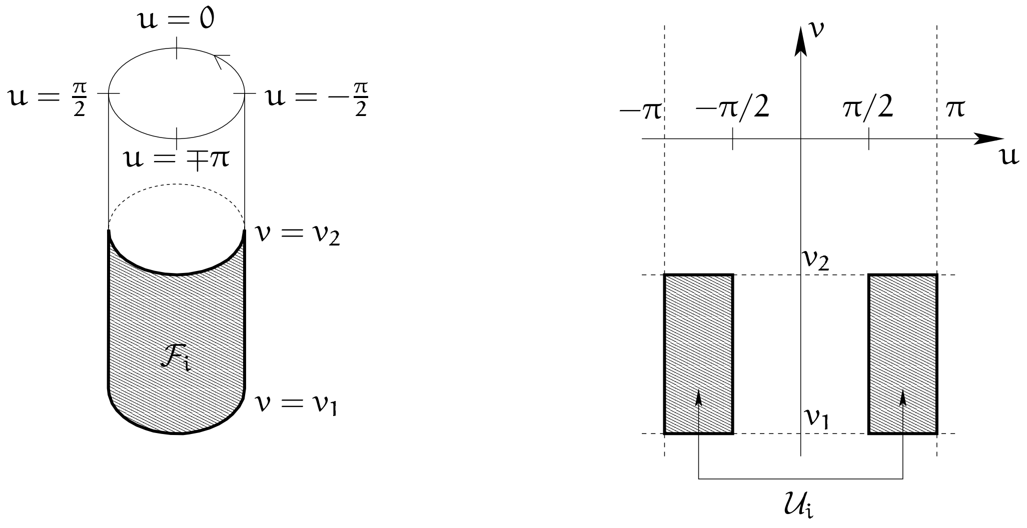


Figure 2.8: A face $\mathcal{F}_i$ embedded on a cylinder and its parameter range $\mathcal{U}_i$ with respect to the parameterization in table 2.2.

- The integrand in (2.28) is the constant 1. We can choose for instance the vector field $\mathbf{V}(\mathbf{x}) = \frac{1}{3}\mathbf{x}$ or $\mathbf{V}(\mathbf{x}) = [x_1, 0, 0]^T$.
- Each component of the integrand in (2.29) is one of the x_i for $i = 1, 2, 3$. One possibility to choose the vector fields is $\mathbf{V}(\mathbf{x}) = [V_1(\mathbf{x}), V_2(\mathbf{x}), V_3(\mathbf{x})]^T$ with $V_i(\mathbf{x}) = \frac{1}{2}x_i^2$ and $V_j \equiv 0$ for $j \neq i$.
- Each component of the integrand in (2.30) can either be written as $x_i^2 + x_j^2$ for $i \neq j$ or as $-x_i x_j$ for $i \neq j$. In the former case we can choose the vector field as $\mathbf{V}(\mathbf{x}) = [V_1(\mathbf{x}), V_2(\mathbf{x}), V_3(\mathbf{x})]^T$ with $V_k(\mathbf{x}) = \frac{1}{3}x_k^3$ for $k = i, j$ and $V_k \equiv 0$ for the remaining entry. In the latter case we may choose $V_i(\mathbf{x}) = -\frac{1}{2}x_i^2 x_j$ and $V_k \equiv 0$ for $k \neq i$.

Now, we show how these surface integrals can be transferred into path integrals using Green's theorem.

Theorem 2.44 (Green's theorem). *Let $\mathcal{B}$ be a region in the plane with piecewise smooth boundary $\partial\mathcal{B}$, and let $\mathbf{W} = [S, T]^T$ be a vector field which is C^1-continuous on $\mathcal{B}$. Then,*

$$\iint_{\mathcal{B}} \left(\frac{\partial T}{\partial u} - \frac{\partial S}{\partial v} \right) du dv = \int_{\partial\mathcal{B}} S du + T dv. \tag{2.37}$$

If we use the parameterizations in table 2.2 and equation (2.4) for the surfaces $\mathcal{S}_i$, it is obvious that the planar regions $\mathcal{U}_i$ in (2.36) fulfill the preconditions of this theorem. The boundary of $\mathcal{U}_i$ consists of finitely many smooth edges $\mathcal{E}_{ij}$. Note that the $\mathcal{E}_{ij}$ do not necessarily correspond to the edges of $\mathcal{F}_i$. This is illustrated in figure 2.8 that shows a face $\mathcal{F}_i$ embedded on a cylinder. The parameterization of this cylinder is taken from table 2.2 where the parameter ranges are $u \in [-\pi, \pi]$ and $v \in \mathbb{R}$. The face is bounded by two linear and two circular edges. The right-hand image shows the parameter range $\mathcal{U}_i$ which consists of two rectangular areas. The boundaries of these areas correspond to the linear edges of $\mathcal{F}_i$, pieces of the circular edges of $\mathcal{F}_i$ and pieces of the boundary of the parameter space of the cylinder.

In order to have an easier notation when we apply Green's theorem to the surface integrals in (2.36) we define $f_i(u,v) = \boldsymbol{V}(\mathbf{r}_i(u,v))^T\mathbf{n}_i(u,v)$. We can write the right-hand side of (2.37) with $\mathcal{B} = \mathcal{U}_i$ as

$$\sum_j \int_{\mathcal{E}_{ij}} S_i du + T_i dv,$$

where $\boldsymbol{W}_i = [S_i, T_i]^T$ is chosen in such a way that $f_i = \frac{\partial T_i}{\partial u} - \frac{\partial S_i}{\partial v}$. Let the edge $\mathcal{E}_{ij}$ be parameterized by $\mathbf{p}_{ij}(t)$ and let $\mathcal{T}_{ij} \subset \mathbb{R}$ be the parameter interval of $\mathcal{E}_{ij}$. If we substitute $[u,v]^T$ by $\mathbf{p}_{ij}(t)$ then we have $[du, dv]^T = \dot{\mathbf{p}}_{ij}dt$. With this, the jth addend in the above summation can be written as

$$\int_{\mathcal{T}_{ij}} \boldsymbol{W}_i^T \dot{\mathbf{p}}_{ij} dt.$$

With this, (2.37) becomes

$$\iint_{\mathcal{U}_i} f_i du dv = \sum_j \int_{\mathcal{T}_{ij}} \boldsymbol{W}_i^T \dot{\mathbf{p}}_{ij} dt. \tag{2.38}$$

We have to take care to follow the boundary of $\mathcal{U}_i$ in mathematically positive direction during the integration. We choose the vector field $\boldsymbol{W}_i = [S_i, T_i]^T$ in the following way. We set $T_i \equiv 0$ and choose $-S_i$ to be any antiderivative of f_i with respect to v, i.e.

$$S_i(u,v) = -\int_{v_0}^{v} f_i(u,\bar{v}) d\bar{v}$$

for an arbitrarily chosen value v_0. With this, the condition $f_i = \frac{\partial T_i}{\partial u} - \frac{\partial S_i}{\partial v}$ is obviously fulfilled. The function $S_i(u,v)$ can always be written in closed form, which can be seen as follows. From the definition of f_i and from the fact that we use surface parameterizations of the form shown in table 2.2 and equation (2.4) if follows that $f_i(u,v)$ viewed as a function of the variable v is either rational, a sum of powers of sine and cosine functions or a sum of powers of hyperbolic functions. Using the substitutions (2.7) or (2.9) the integral over f_i can always be transformed into an integral over a rational function. But such an integral can be solved in closed form by means of decomposition into partial fractions.

In order to use (2.38) we have to determine the boundary of each $\mathcal{U}_i$ and integrate along it in mathematically positive direction, i.e. the interior of $\mathcal{U}_i$ must always lie to the left of the integration path. We describe briefly how this can be done. First, we see that in order to compute the integral (2.38) we need for each edge $\mathcal{E}_{ij}$ of the boundary of $\mathcal{U}_i$ a parameterization $\mathbf{p}_{ij}$. Moreover, we must be able to compute the derivative $\dot{\mathbf{p}}_{ij}(t)$

for any $t \in \mathcal{T}_{ij}$. Each $\mathcal{E}_{ij}$ is either a part of the boundary of the parameter space of the surface $\mathcal{S}_i$ containing the face $\mathcal{F}_i$ or corresponds to a part of an edge of $\mathcal{F}_i$. If we use the parameterizations given by table 2.2 and equation (2.4), the parameter space is either $\mathbb{R}^2$ or an area in $\mathbb{R}^2$ that is bounded by straight lines or segments of straight lines. Hence, it is easy to determine a parameterization $\mathbf{p}_{ij}$ of an edge $\mathcal{E}_{ij}$ corresponding to a part of that boundary as well as the derivative $\dot{\mathbf{p}}_{ij}$. This parameterization must be chosen in such a way that $\mathcal{E}_{ij}$ is oriented such that the interior of the parameter range lies to its left. If $\mathcal{E}_{ij}$ corresponds to an edge of $\mathcal{F}_i$, then let $\mathbf{q}(t)$ be a parameterization of the curve containing that edge in $\mathbb{R}^3$. We have seen in the section about curves and surfaces that the surface parameterization $\mathbf{r}_i$ is one-to-one and onto if the parameter range is restricted appropriately. Hence, we can parameterize $\mathcal{E}_{ij}$ in the parameter space of the face by defining $\mathbf{p}_{ij}(t) = \mathbf{r}_i^{-1}(\mathbf{q}(t))$. The derivative $\dot{\mathbf{p}}_{ij}$ can be computed as follows. By applying $\mathbf{r}_i$ to $\mathbf{p}_{ij}(t)$ we obtain $\mathbf{r}_i(\mathbf{p}_{ij}(t)) = \mathbf{q}(t)$ and derivating this yields

$$\left[\frac{\partial \mathbf{r}_i}{\partial u}(\mathbf{p}_{ij}(t)), \frac{\partial \mathbf{r}_i}{\partial v}(\mathbf{p}_{ij}(t))\right] \cdot \dot{\mathbf{p}}_{ij}(t) = \dot{\mathbf{q}}(t).$$

This is a system of linear equations for $\dot{\mathbf{p}}_{ij}(t)$. The rank of this system is two since we do not allow faces with singularities. Thus, there is a unique solution that we can find using Cramer's rule, which gives

$$\dot{\mathbf{p}}_{ij}(t) = \frac{1}{\mathbf{n}_i(\mathbf{p}_{ij}(t))^2} \cdot \left[\frac{\partial \mathbf{r}_i}{\partial v}(\mathbf{p}_{ij}(t)) \times \mathbf{n}_i(\mathbf{p}_{ij}(t)),\ \mathbf{n}_i(\mathbf{p}_{ij}(t)) \times \frac{\partial \mathbf{r}_i}{\partial u}(\mathbf{p}_{ij}(t))\right]^T \dot{\mathbf{q}}(t).$$

This is an improvement of the approach described in [Imm01] where this derivative is computed numerically. Now we are able to evaluate the integrand functions in the summation on the right-hand side of (2.38) for each value of t. We can solve these integrals using numerical methods such as Romberg integration which can be found in [PTVF94].

In order to perform the integration along the whole boundary of $\mathcal{U}_i$ and in mathematically positive direction we proceed as follows. We use figure 2.9 to illustrate the different steps. The dashed horizontal lines symbolize the boundary of the parameter space of the surface containing the face. We can assume that $\mathcal{F}_i$ has edges. This is because the only cases where a face can have no edges is if it is a complete ellipsoid or torus. But then, the object $\mathcal{O}$ consists only of this face. But there are closed formulae for the volume, center of mass and moments of inertia of the ellipsoid and the torus.

- First, we compute the intersection points $\mathbf{a}_1, \ldots, \mathbf{a}_k$ between those $\mathcal{E}_{ij}$ that correspond to edges of $\mathcal{F}_i$ and the boundary of the parameter space of the surface $\mathcal{S}_i$. In figure 2.9 there are five such points. The boundary of the parameter space corresponds to conics in $\mathbb{R}^3$. Therefore, the computation of the intersection points can be performed in the same way as described for the ray shooting approach in the point-in-face test in section 3.2. This is an improvement of [Imm01] where these intersection points are determined numerically. Let us call an edge of $\mathcal{F}_i$ *crossing* if its counterpart in the parameter space of $\mathcal{S}_i$ intersects the boundary of

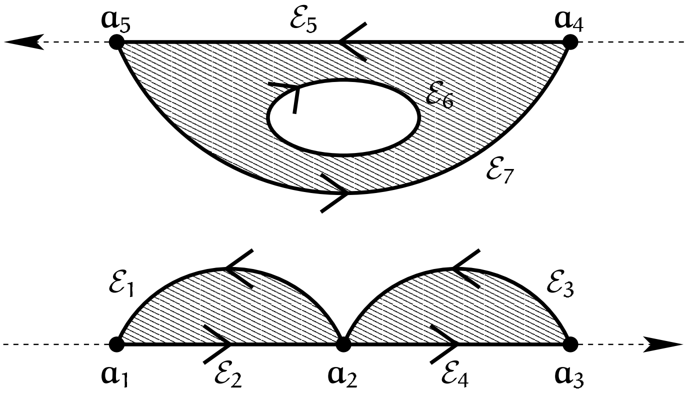


Figure 2.9: Sketch of a parameter range of a face.

this parameter space, and *non-crossing* otherwise. Let us furthermore call a loop of $\mathcal{F}_i$ crossing if one of its edges is crossing, and non-crossing otherwise.

- Next, we integrate along the non-crossing loops of $\mathcal{F}_i$. More precisely, this means we iterate through the edges of the non-crossing loops and integrate along their counterparts in the parameter space. The direction of the integration is implicitly given by the orientation of the edges. In figure 2.9 this means we integrate along the edge denoted by $\mathcal{E}_6$.

- Now, we integrate along those components of $\mathcal{U}_i$ that involve the boundary of the parameter space. Therefore, we choose one of the intersection points, say $\mathbf{a}_{l_0}$. Let $\mathbf{u}$ be the direction of the boundary of the parameter space in that point and let $\mathbf{t}$ be the tangent of the $\mathcal{E}_{ij}$ intersecting the boundary in that point. If $\mathcal{E}_{ij}$ leaves the parameter space at $\mathbf{a}_{l_0}$, then the integration path follows the boundary of the parameter space. Otherwise, we follow $\mathcal{E}_{ij}$. The edge leaves the parameter space iff $t_1u_2 - t_2u_1 > 0$. Let us choose the point $\mathbf{a}_1$ in figure 2.9. Since $\mathcal{E}_1$ leaves the parameter space at that point we follow the boundary along $\mathcal{E}_2$. If we reach a point $\mathbf{a}_l$ with $l \neq l_0$, then we 'turn left' with respect to our current direction. In the figure, the next point we reach is $\mathbf{a}_2$ and our current direction is the tangent of $\mathcal{E}_2$ in $\mathbf{a}_2$. The leftmost direction that we can follow from that point is that of $\mathcal{E}_1$. We omit the details of how to choose the 'leftmost' direction which is a rather simple task. We repeat this until we reach again the starting point $\mathbf{a}_{l_0}$. If there are still unvisited intersection points, then we choose one of them, say $\mathbf{a}_{l_1}$ and proceed similarly. This is repeated until all $\mathbf{a}_l$'s have been visited.

- In the case that the parameter space is a finite area in $\mathbb{R}^2$, i.e. $\mathcal{S}_i$ is an ellipsoid or a torus, we have to decide whether the boundary of the parameter space belongs to $\mathcal{U}_i$ or not. Figure 2.10 illustrates the two cases. In order to make this decision, we randomly choose a point $\mathbf{a}$ on the boundary of the parameter space that does not lie on one of the $\mathcal{E}_{ij}$ along which we have already integrated. We construct a

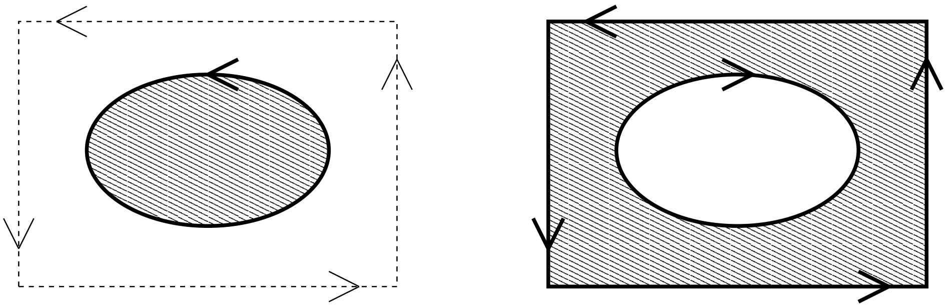


Figure 2.10: If the parameter space of $\mathcal{S}_i$ is a finite area, its boundary might or might not belong to the boundary of $\mathcal{U}_i$.

path $\mathcal{L}$ in the interior of the parameter space of $\mathcal{S}_i$ starting at $\mathbf{a}$ to an $\mathcal{E}_{ij}$ that corresponds to an edge of $\mathcal{F}_i$. We compose this path of horizontal and vertical line segments. We determine the first point where $\mathcal{L}$ intersects one of the $\mathcal{E}_{ij}$. Since the segments that $\mathcal{L}$ consists of correspond to segments of conics in $\mathbb{R}^3$, this can easily be done. Let $\mathbf{b}$ be this point. Let $\mathbf{t}$ denote the tangent of $\mathcal{E}_{ij}$ in $\mathbf{b}$ and let $\mathbf{u}$ be the direction of the segment of $\mathcal{L}$ that passes through $\mathbf{b}$. Then $\mathcal{L}$ leaves the range $\mathcal{U}_i$ at the point $\mathbf{b}$ iff $u_1 t_2 - u_2 t_1 > 0$. In this case, $\mathbf{a}$ belongs to $\mathcal{U}_i$ and otherwise it does not.

3 Collision Detection

In the context of dynamics simulations the objects in a virtual environment are not stationary. They rather move around caused by user interaction or interactions with each other. An important invariant that has to be maintained is that there is no interpenetration between two objects. This means that two objects are either disjoint or their boundaries intersect, but their interiors are disjoint. A collision detection algorithm is designed to decide whether there are interpenetrations between two objects. In this chapter we present two different kinds of collision detection algorithms for objects with curved surfaces. The *static* collision detection decides for two stationary objects whether they interpenetrate or not. The algorithm that we present also reports a collision if only the boundaries of the objects intersect. Starting with a valid configuration, i.e. a collision-free configuration, such an algorithm can be used at discrete points in time to decide whether the configuration is still valid. If it decides that it is not, then the last motion step is rejected. The *dynamic* collision test decides whether two moving objects will interpenetrate during a given time interval. It also computes the point in time in this interval when the collision occurs.

Before we start describing these algorithms, we present heuristics for fast feature culling. One of these heuristics is the space partitioning which is used to quickly reduce the number of object pairs that have to be checked for collision. This heuristic exploits the fact that two objects that have a large distance compared to their relative velocity cannot collide. The second heuristic that we present is the bounding volume hierarchy which reduces the number of object parts that have to be considered during the collision test.

3.1 Heuristics for Fast Feature Culling

3.1.1 Bounding Volume Hierarchies

A bounding volume hierarchy is a data structure that is used to quickly reduce the number of faces of the objects that have to be considered during a collision test. Let $\mathfrak{F}_1$ and $\mathfrak{F}_2$ be two sets of faces, and let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two volumes containing $\mathfrak{F}_1$ and $\mathfrak{F}_2$, respectively. The basic idea is that if $\mathcal{B}_1$ and $\mathcal{B}_2$ do not intersect, then neither do $\mathfrak{F}_1$ and $\mathfrak{F}_2$.

A bounding volume hierarchy for an object $\mathcal{O}$ is a tree structure. A node of this tree is a pair $(\mathfrak{F}, \mathcal{B})$, where $\mathfrak{F}$ is a subset of $\mathcal{O}$'s faces and $\mathcal{B}$ is a bounding volume of $\mathfrak{F}$. The set of faces in the root node is the set of all faces of $\mathcal{O}$. The sets of faces of the children of a node $(\mathfrak{F}, \mathcal{B})$ form a partition of $\mathfrak{F}$. In this way, each level of the tree represents a partition of all faces of $\mathcal{O}$. In the following we concentrate on binary hierarchies, i.e. each inner node has exactly two children.

A collision test for two objects $\mathcal{O}_1$ and $\mathcal{O}_2$ using bounding volume hierarchies works as follows. First, the bounding volumes in the root nodes $\mathfrak{p}_1$ and $\mathfrak{p}_2$ of the objects are tested for collision. If they do not collide, then neither do $\mathcal{O}_1$ and $\mathcal{O}_2$. Otherwise, the volumes in the children of $\mathfrak{p}_1$ are tested pairwise against the volumes in the children of $\mathfrak{p}_2$. This is recursively repeated. If $\mathfrak{p}_1$ is an inner node and $\mathfrak{p}_2$ is a leaf, then only $\mathfrak{p}_1$ is expanded. If they are both leaves, then the faces in $\mathfrak{p}_1$ are tested pairwise against the faces in $\mathfrak{p}_2$ using the basic collision tests described in the sections 3.2 and 3.3.

There are different types of bounding volumes that can be used in such a hierarchy. The most important ones are

- spheres [Qui94, Hub95, PG95],
- oriented boxes [GLM96],
- k-DOPS [HKM$^+$96],
  special case: iso-boxes [ZF95, Zac97].

The k-DOPS are convex polyhedra with fixed directions. They are described by the minimal and maximal extent in $k/2$ directions. If $k = 6$ and the three directions are the coordinate axes one obtains the axis aligned boxes or iso-boxes as an important special case. Two such k-DOPS can be tested for collision very efficiently but since the directions are fixed, they have to be recomputed whenever the orientation of the object changes. This means a great computational effort especially if the boundary of the object is curved. Therefore we concentrate on spheres and oriented boxes in the following.

A hierarchy of bounding volumes can be uniform or hybrid. In a uniform hierarchy the same bounding volume type is used in each node whereas in the hybrid case different nodes may have bounding volumes of different types. We only consider uniform hierarchies here and moreover, we assume that for all objects in the scene the same bounding volume type is used.

In the following we describe how a hierarchy of bounding volumes can be constructed for an object of one of the classes defined in 2.2.3. This construction consists of two steps, namely the computation of a bounding volume for a given set of faces and the partitioning of a set of faces into two subsets. Finally, we describe both a static and a dynamic collision test for two bounding volumes of the same type.

Computing a bounding volume for a given set of faces

Let $\mathfrak{F}$ denote the set of faces, $\mathfrak{E}$ the set of all edges in $\mathfrak{F}$ and $\mathfrak{V}$ the set of all vertices in $\mathfrak{F}$. First, we give a general description how to compute a smallest bounding sphere or a smallest bounding box for $\mathfrak{F}$. Afterwards we describe how the basic computations can be performed for our classes of objects. The methods we present here are in more detail described in [Rei01] and were published in [LRSW02].

Smallest bounding sphere Our aim is to compute an enclosing sphere for the set $\mathfrak{F}$ with minimal volume. Since the volume of a sphere is a monotone function in its radius this is equivalent to computing an enclosing sphere with minimal radius. We define the function $\mathsf{d}(\mathbf{p})$, which computes for a point $\mathbf{p}$ the maximal distance between $\mathbf{p}$ and any point of the faces. This function is given by

$$\mathsf{d}(\mathbf{p}) = \max_{\mathbf{x}\in\mathfrak{F}} |\mathbf{p} - \mathbf{x}|, \tag{3.1}$$

where we wrote $\mathbf{x} \in \mathfrak{F}$ instead of $\mathbf{x} \in \mathcal{F}, \mathcal{F} \in \mathfrak{F}$ for the sake of simplicity. Obviously, the radius of a smallest sphere enclosing $\mathfrak{F}$ with center $\mathbf{c}$ is given by $\mathsf{d}(\mathbf{c})$. Thus, in order to compute an enclosing sphere with minimal radius we must find a center $\mathbf{c}$ such that $\mathsf{d}(\mathbf{c})$ is minimal.

Lemma 3.1. *The maximal distance* $\mathsf{d}(\mathbf{p})$ *is a convex function of* $\mathbf{p}$.

Proof. First, we prove that the function $|\mathbf{p} - \mathbf{x}|$ is convex in $\mathbf{p}$.

$$\begin{aligned}
|t\mathbf{p} + (1-t)\mathbf{q} - \mathbf{x}| &= |t\mathbf{p} + (1-t)\mathbf{q} - t\mathbf{x} - (1-t)\mathbf{x}| \\
&= |t(\mathbf{p}-\mathbf{x}) + (1-t)(\mathbf{q}-\mathbf{x})| \\
&\leq t|\mathbf{p}-\mathbf{x}| + (1-t)|\mathbf{q}-\mathbf{x}| \quad \text{for } t \in [0,1]
\end{aligned}$$

by the triangle inequality. In the following we omit the subscript $\mathbf{x} \in \mathfrak{F}$ for better readability. For $t \in [0,1]$ we have

$$\begin{aligned}
\mathsf{d}(t\mathbf{p} + (1-t)\mathbf{q}) &= \max |t\mathbf{p} + (1-t)\mathbf{q} - \mathbf{x}| \\
&\leq \max \big(t|\mathbf{p}-\mathbf{x}| + (1-t)|\mathbf{q}-\mathbf{x}|\big) \\
&\leq \max \big(t|\mathbf{p}-\mathbf{x}|\big) + \max \big((1-t)|\mathbf{q}-\mathbf{x}|\big) \\
&= t\mathsf{d}(\mathbf{p}) + (1-t)\mathsf{d}(\mathbf{q}).
\end{aligned}$$

□

Provided that we can evaluate the function $\mathsf{d}(\mathbf{p})$, this lemma ensures that we can find the optimal solution using local optimization methods such as the *downhill simplex method* according to Nelder and Mead or *Powell's method*, both described in [PTVF94]. We will see later that the maximal distance function can be evaluated by solving univariate polynomial equations.

Smallest bounding box Now we want to find a box of minimal volume that encloses the set $\mathfrak{F}$. We define a box by a corner $\mathbf{c}$, an orientation and the three lengths δ_1, δ_2 and δ_3 of the axes. Assume the orientation is given by three orthogonal unit vectors

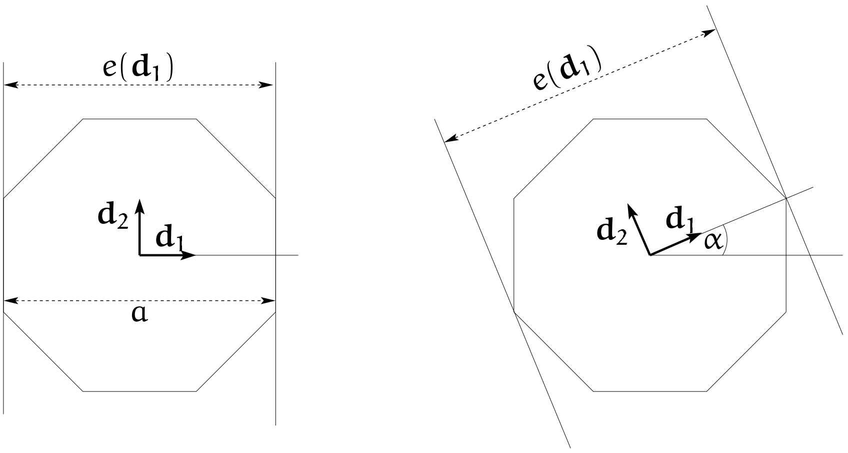


Figure 3.1: Extent of an octagon.

$\mathbf{d}_1, \mathbf{d}_2$ and $\mathbf{d}_3$. Then all corners of the box are given by $\mathbf{c} + \sum_{i \in \mathcal{I}} \delta_i \mathbf{d}_i$, $\mathcal{I} \subseteq \{1,2,3\}$. If we know the vectors $\mathbf{d}_i$, then we can compute the corner $\mathbf{c}$ and the lengths δ_i of the smallest enclosing box with that orientation in the following way. We define the *extent* of $\mathfrak{F}$ in direction $\mathbf{d}$ for $|\mathbf{d}| = 1$ as

$$\begin{aligned} e(\mathbf{d}) &= e_{max}(\mathbf{d}) - e_{min}(\mathbf{d}), \quad \text{where} \\ e_{max}(\mathbf{d}) &= \max_{x \in \mathfrak{F}} \mathbf{d}^T \mathbf{x}, & (3.2) \\ e_{min}(\mathbf{d}) &= \min_{x \in \mathfrak{F}} \mathbf{d}^T \mathbf{x}. & (3.3) \end{aligned}$$

The corner $\mathbf{c}$ of the enclosing box is given by the solution of the system $\mathbf{d}_i^T \mathbf{x} = e_{min}(\mathbf{d}_i), i = 1,2,3$. The lengths can be computed as $\delta_i = e(\mathbf{d}_i), i = 1,2,3$. We need to find an orientation $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$ such that the resulting volume $\delta_1 \delta_2 \delta_3$ is minimal. But in contrast to the sphere case, this objective function is not convex in general, as the following example shows.

Example 3.2. *For the sake of simplicity we present a two dimensional example. But this can immediately be generalized to three dimensions.*

We consider a regular octagon, whose opposite edges have distance a. *Figure 3.1 shows the extent* $e(\mathbf{d}_1)$ *of the octagon for two different angles* α. *By the left-hand picture, for* $\alpha = 0$ *we have* $e(\mathbf{d}_1) = a$. *Obviously, the same holds for* $\alpha = \frac{\pi}{4}$. *In the right-hand picture we chose* $\alpha = \frac{\pi}{8}$ *which gives the extent*

$$e(\mathbf{d}_1) = \frac{a}{\cos \frac{\pi}{8}} \approx 1.082a > a.$$

Because of the symmetry of the octagon it holds for all values of α *that* $e(\mathbf{d}_1) = e(\mathbf{d}_2)$. *Thus, the volume of the enclosing box has two local minima and hence is not convex. In this example both local minima have the same value. But it is easy to create a situation where this is not the case, e.g. by considering an octagon with different edge lengths.*

A consequence of the fact that the volume of the enclosing box is not a convex function in the orientation parameters is that local optimization methods such as the downhill simplex method or Powell's method might end up in local minima. In [Rei01] it is shown that a good heuristic to overcome this problem is to run a local optimization method for many different starting values.

Another problem in optimizing the volume of the bounding box is the high redundancy in our way to represent orientations. In order to uniquely describe the orientation of an object in three dimensions only three parameters are needed. A well known representation for orientations by three parameters uses the so-called Z-Y-X Euler angles $(\alpha, \beta, \gamma) \in [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [-\pi, \pi]$. For a detailed description of this representation including the computation of the axes $(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)$ from the Euler angles and vice versa we refer to [Cra89].

In the following we describe how to evaluate the functions (3.1), (3.2) and (3.3). Obviously, we can write the function $d(\mathbf{p})$ as

$$d(\mathbf{p}) = \max\{\max_{\mathcal{F}\in\mathfrak{F}} \max_{\mathbf{x}\in\mathcal{F}} |\mathbf{p}-\mathbf{x}|,\ \max_{\mathcal{E}\in\mathfrak{E}} \max_{\mathbf{x}\in\mathcal{E}} |\mathbf{p}-\mathbf{x}|,\ \max_{\mathbf{x}\in\mathfrak{V}} |\mathbf{p}-\mathbf{x}|\}.$$

This suggests the following procedure. For each face $\mathcal{F} \in \mathfrak{F}$ compute all local maxima of the function $|\mathbf{p} - \mathbf{x}|$ for $\mathbf{x} \in \mathcal{S}$, where $\mathcal{S}$ is the surface containing $\mathcal{F}$. For each such maximum check whether the corresponding point $\mathbf{x}$ is contained in $\mathcal{F}$. If it is not we neglect that maximum. This *point-in-face test* will be described in detail in section 3.2. Then, for each edge $\mathcal{E} \in \mathfrak{E}$ compute all local maxima of the function $|\mathbf{p}-\mathbf{x}|$ for $\mathbf{x} \in \mathcal{C}$, where $\mathcal{C}$ is the curve containing $\mathcal{E}$. Again, for each such maximum check whether the corresponding point $\mathbf{x}$ is contained in $\mathcal{E}$. If this is not the case we neglect that maximum. This *point-on-edge test* will also be described in section 3.2. Finally compute the maximal distance $|\mathbf{p}-\mathbf{x}|$ for $\mathbf{x} \in \mathfrak{V}$. The value of $d(\mathbf{p})$ is the maximum of all these extrema.

Similarly, we observe that we can write the functions $e_{max}(\mathbf{d})$ and $e_{min}(\mathbf{d})$ as

$$\begin{aligned} e_{max}(\mathbf{d}) &= \max\{\max_{\mathcal{F}\in\mathfrak{F}} \max_{\mathbf{x}\in\mathcal{F}} \mathbf{d}^T\mathbf{x},\ \max_{\mathcal{E}\in\mathfrak{E}} \max_{\mathbf{x}\in\mathcal{E}} \mathbf{d}^T\mathbf{x},\ \max_{\mathbf{x}\in\mathfrak{V}} \mathbf{d}^T\mathbf{x}\}, \\ e_{min}(\mathbf{d}) &= \min\{\min_{\mathcal{F}\in\mathfrak{F}} \min_{\mathbf{x}\in\mathcal{F}} \mathbf{d}^T\mathbf{x},\ \min_{\mathcal{E}\in\mathfrak{E}} \min_{\mathbf{x}\in\mathcal{E}} \mathbf{d}^T\mathbf{x},\ \min_{\mathbf{x}\in\mathfrak{V}} \mathbf{d}^T\mathbf{x}\}. \end{aligned}$$

This suggests an analogous procedure as in the case of the maximal distance. The only two differences are that we do not only compute local maxima but all local extrema and that the objective function is now $\mathbf{d}^T\mathbf{x}$.

We give now brief descriptions how to compute the local extrema of the distance function and the extent function. With respect to the object classes that we consider in this thesis, the surfaces that we look at in the following are quadrics and tori and the curves are conics and quadric intersection curves.

Local maxima of the distance between a point and a curve Let $\mathcal{C}$ be the curve in question. If $\mathcal{C}$ is a conic section, then let $\mathbf{x}(t)$ be a rational parameterization. We compute the local maxima of the squared distance between $\mathbf{p}$ and $\mathbf{x}(t)$ by solving

$$\frac{d}{dt}(\mathbf{p}-\mathbf{x}(t))^2 = 0.$$

We notice that this can be written as

$$\frac{d}{dt}\frac{f(t)}{g(t)^2} = \frac{\dot{f}(t)g(t) - 2f(t)\dot{g}(t)}{g(t)^3} = 0,$$

where $f(t)$ and $g(t)$ are polynomials of degree at most four and at most two, respectively. In order to get an upper bound for the degree of the numerator of this equation we assume that the degree of f is four and the degree of g is two. Thus, five is clearly an upper bound. Let the highest coefficients of f and g be denoted by f_4 and g_2, respectively. Then the coefficient of t^5 in the numerator is $4f_4 \cdot g_2 - 2f_4 \cdot 2g_2 = 0$. Hence, the resulting degree is at most four.

If $\mathcal{C}$ is a quadric intersection curve then we assume that it is the intersection between an L-quadric defined by the matrix $\mathbf{A}_H$ and an arbitrary quadric defined by $\mathbf{B}_H$. We assume further that the coordinate system has been transformed in such a way that $\mathbf{A}_H$ has normal form as shown in table 2.1. Since $\mathcal{C}$ is not a conic we know that the L-quadric is not planar. Hence, it must be a hyperbolic or parabolic cylinder or a hyperbolic paraboloid. Let $\mathbf{x}(s,t)$ be a parameterization of the L-quadric. In order for $\mathbf{x}(s,t)$ to have a nice form we rewrite the implicit form of the L-quadric in the following way.

$$\begin{aligned}\text{hyperbolic cylinder:}\quad & \frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} = 1,\\ \text{parabolic cylinder:}\quad & \frac{x_2^2}{b^2} - x_1 = 0,\\ \text{hyperbolic paraboloid:}\quad & \frac{x_1^2}{a^2} - \frac{x_2^2}{b^2} - x_3 = 0.\end{aligned}$$

With this we get the following parameterizations.

$$\mathbf{x}(s,t) = \begin{cases} \left[\pm a\frac{1+t^2}{1-t^2},\ b\frac{2t}{1-t^2},\ s\right]^T & \text{(hyperbolic cylinder)},\\ \left[t^2,\ bt,\ s\right]^T & \text{(parabolic cylinder)},\\ \left[a(s+t),\ b(s-t),\ 4st\right]^T & \text{(hyperbolic paraboloid)}.\end{cases} \tag{3.4}$$

In order to find the maximal distance between a point on $\mathcal{C}$ and $\mathbf{p}$ we maximize the function $(\mathbf{x}(s,t)-\mathbf{p})^2$ under the constraint $\mathbf{x}(s,t)^T\mathbf{B}\mathbf{x}(s,t) + 2\mathbf{x}(s,t)^T\mathbf{b} + b_0 = 0$. This can be done using the Lagrange formalism (see [Ber96]). We introduce the Lagrange multiplier λ and define the function

$$L(s,t,\lambda) = \big(\mathbf{x}(s,t)-\mathbf{p}\big)^2 + \lambda\big(\mathbf{x}(s,t)^T\mathbf{B}\mathbf{x}(s,t) + 2\mathbf{x}(s,t)^T\mathbf{b} + b_0\big).$$

The system of equations that we have to solve is

$$\frac{\partial L}{\partial s}(s,t,\lambda) = \frac{\partial L}{\partial t}(s,t,\lambda) = \frac{\partial L}{\partial \lambda}(s,t,\lambda) = 0$$

which we write as $L_s(s,t,\lambda) = L_t(s,t,\lambda) = L_\lambda(s,t,\lambda) = 0$. We observe that L_s and L_t are both linear in λ and L_λ does not depend on λ at all. Thus, we eliminate λ from L_s and L_t by computing the resultant w.r.t. λ. Using some vector algebra we find that the resulting equation can be written in the from

$$f(s,t) = \big(\mathbf{x}(s,t) - \mathbf{p}\big)^T \Big(\big(\mathbf{x}_s(t) \times \mathbf{x}_t(s,t)\big) \times \big(\mathbf{B}\mathbf{x}(s,t) + \mathbf{b}\big)\Big) = 0.$$

This equation can be interpreted geometrically. The cross-product between the two partial derivatives of $\mathbf{x}$ is for all values of s and t parallel to the normal of the L-quadric in the point $\mathbf{x}(s,t)$. Thus, if $\mathbf{x}(s,t)$ lies on the intersection curve $\mathcal{C}$, then the right-hand side of the dot-product of the above equation is parallel to the cross-product of the two normals of the intersecting surfaces in that point. But this cross-product is parallel to the tangent of $\mathcal{C}$ in $\mathbf{x}(s,t)$. Hence, this equation is satisfied for a point $\mathbf{x}(s,t)$ on $\mathcal{C}$ if and only if the vector pointing from $\mathbf{p}$ to $\mathbf{x}(s,t)$ is perpendicular to the tangent of $\mathcal{C}$ in $\mathbf{x}(s,t)$.

In order to find the parameters of the points on $\mathcal{C}$ with extremal distance we compute the resultant of $f(s,t)$ and $g(s,t) = \mathbf{x}(s,t)^T\mathbf{B}\mathbf{x}(s,t) + 2\mathbf{x}(s,t)^T\mathbf{b} + b_0$ with respect to s. Although in the case that $\mathbf{x}(s,t)$ parameterizes a hyperbolic cylinder there is a denominator, this involves only the parameter t. The functions f and g are always polynomials in s of degree three and two, respectively. Let $r(t) = \text{res}_s(f,g)(t)$. In the case of $\mathbf{x}(s,t)$ parameterizing a parabolic cylinder we can compute $r(s,t)$ symbolically using Maple®. The resulting degree in t is ten. If $\mathbf{x}(s,t)$ parameterizes a hyperbolic cylinder Maple® is also able to compute the resultant symbolically and the degree of the numerator of the result is twelve. In the case of a hyperbolic paraboloid we are not able to compute the resultant symbolically. But the degree of g in t is two and the degree of f in t is three. Thus, the Sylvester matrix of f and g with respect to s has three rows of degree two in t and two rows of degree three in t. Hence, the degree of $r(t)$ is at most twelve in t.

Local maxima of the distance between a point and a surface Let $\mathcal{S}$ be the surface in question. First we assume that $\mathcal{S}$ is a quadric. If $\mathcal{S}$ is ruled, then there are no local maxima between $\mathcal{S}$ and a point $\mathbf{p}$. This can be justified as follows. Let $\mathbf{A}_H$ be the matrix defining the quadratic form of $\mathcal{S}$. A point $\mathbf{x}$ on $\mathcal{S}$ has extremal distance from the point $\mathbf{p}$ if the vector pointing from $\mathbf{x}$ to $\mathbf{p}$ is parallel to the normal of $\mathcal{S}$ in $\mathbf{x}$, i.e. there is a real value for λ such that

$$\mathbf{A}\mathbf{x} + \mathbf{a} = \lambda(\mathbf{p} - \mathbf{x}). \tag{3.5}$$

If $\mathcal{S}$ is ruled there is a straight line $\mathcal{L}$ on $\mathcal{S}$ that passes through $\mathbf{x}$. Equation (3.5) implies that the vector $\mathbf{p}-\mathbf{x}$ is perpendicular to $\mathcal{L}$. This means that $\mathbf{x}$ is also a point of extremal distance between $\mathcal{L}$ and $\mathbf{p}$. But the only local extremum of the distance between a point and a straight line is a minimum. Hence, $\mathbf{x}$ cannot be a point of maximal distance between $\mathbf{p}$ and $\mathcal{S}$.

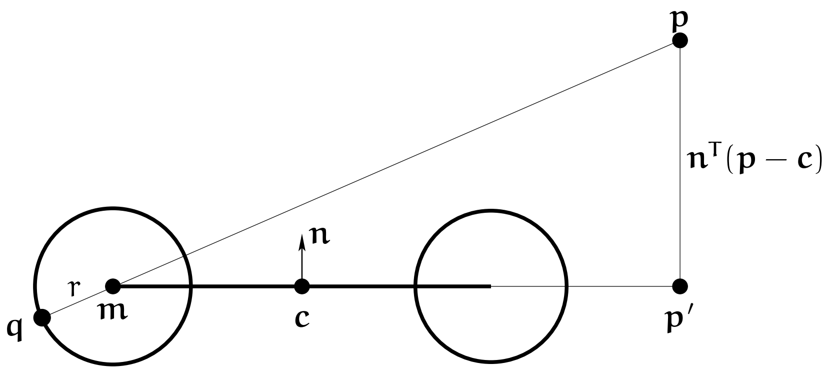


Figure 3.2: Point $\mathbf{q}$ on the torus with maximal distance from $\mathbf{p}$.

Now we consider the case that $\mathcal{S}$ is not ruled. Equation (3.5) is equivalent to $(\lambda\mathbf{E} + \mathbf{A})\mathbf{x} = \lambda\mathbf{p} - \mathbf{a}$. We multiply this by adj $(\lambda\mathbf{E} + \mathbf{A})$ and obtain

$$|\lambda\mathbf{E} + \mathbf{A}|\mathbf{x} = \overline{\lambda\mathbf{E} + \mathbf{A}}(\lambda\mathbf{p} - \mathbf{a}), \tag{3.6}$$

where we wrote $|\mathbf{M}|$ for the determinant of $\mathbf{M}$ and $\overline{\mathbf{M}}$ for the adjoint of $\mathbf{M}$. Multiplying the quadratic form of $\mathcal{S}$ by $\det(\lambda\mathbf{E} + \mathbf{A})$ and inserting (3.6) we obtain a polynomial equation of degree at most six in λ. In order to compute the points on $\mathcal{S}$ with extremal distance to $\mathbf{p}$ we compute the solutions of these equations and use (3.5) to compute the corresponding point $\mathbf{x}$.

Let now $\mathcal{S}$ be a torus with center $\mathbf{c}$, normal $\mathbf{n}$ with $|\mathbf{n}| = 1$, and radii R and r with $R > r$. If $\mathbf{n}^T(\mathbf{p} - \mathbf{c}) = 0$, i.e. $\mathbf{p}$ lies in the main plane of the torus, then the maximal distance can be computed as follows. If $\mathbf{p} = \mathbf{c}$, then the desired distance is simply $R+r$. Otherwise, the point on $\mathcal{S}$ with maximal distance to $\mathbf{p}$ lies on the line through $\mathbf{p}$ and $\mathbf{c}$. In this case the desired distance is given by $R + r$ plus the distance between $\mathbf{p}$ and $\mathbf{c}$, which gives $R + r + |\mathbf{p} - \mathbf{c}|$. Now assume that $\mathbf{p}$ does not lie in the main plane of $\mathcal{S}$. With the notation as in figure 3.2 the maximal distance $|\mathbf{p} - \mathbf{q}|$ is equal to $r + |\mathbf{p} - \mathbf{m}|$. The distance between $\mathbf{p}$ and $\mathbf{m}$ can be computed as

$$|\mathbf{p} - \mathbf{m}| = \sqrt{\left(\mathbf{n}^T(\mathbf{p} - \mathbf{c})\right)^2 + \left(R + |\mathbf{p}' - \mathbf{c}|\right)^2}.$$

Finally, we have $|\mathbf{p}' - \mathbf{c}| = |\mathbf{p} - \mathbf{c} - \mathbf{n}^T(\mathbf{p} - \mathbf{c})\mathbf{n}|$.

Local extrema of the extent of a curve Let $\mathcal{C}$ be the curve in question and let $\mathbf{d}$ be a unit vector defining the direction in which the extent of $\mathcal{C}$ is to be computed. First, we assume that $\mathcal{C}$ is a conic. We do not need to consider straight lines since in that case the extrema are attained at the vertices of the edge embedded on $\mathcal{C}$. Let $\mathbf{x}(t)$ be a parameterization of $\mathcal{C}$. We compute the local extrema of the extent by solving

$$\frac{d}{dt}\mathbf{d}^T\mathbf{x}(t) = 0.$$

We observe that we can write this as

$$\frac{d}{dt}\frac{f(t)}{g(t)} = \frac{\dot{f}(t)g(t) - f(t)\dot{g}(t)}{g(t)^2},$$

where f and g are polynomials of degree at most two. In order to get an upper bound for the degree of the numerator of this equation we assume that the degree of both of them is two. Obviously, three is an upper bound for this degree. Let the highest coefficients of f and g be denoted by f_2 and g_2, respectively. Then the coefficient of t^3 in the numerator is $2f_2 \cdot g_2 - f_2 \cdot 2g_2 = 0$. Hence, the resulting degree is at most two.

Next, we consider the case that $\mathcal{C}$ is a quadric intersection curve. We assume that $\mathcal{C}$ is the intersection between an L-quadric and an arbitrary quadric. Furthermore, we suppose that the coordinate system has been transformed in such a way that the L-quadric is given by one of the parameterizations in (3.4). Let $\mathbf{B}_H$ be the matrix defining the implicit form of the other quadric. We have to compute the local extrema of the function $\mathbf{d}^T\mathbf{x}(s,t)$ subject to the constraint $g(s,t) = \mathbf{x}(s,t)^T\mathbf{B}\mathbf{x}(s,t) + 2\mathbf{x}(s,t)^T\mathbf{b} + b_0 = 0$. We introduce the Lagrange multiplier λ and define the function

$$L(s,t,\lambda) = \mathbf{d}^T\mathbf{x}(s,t) + \lambda g(s,t).$$

We have to solve the system $L_s(s,t,\lambda) = L_t(s,t,\lambda) = L_\lambda(s,t,\lambda) = 0$, where L_s, L_t and L_λ denote the partial derivatives of L. We notice that both L_s and L_t are linear in λ and that $L_\lambda(s,t) = g(s,t)$ does not depend on λ at all. We eliminate λ from L_s and L_t by computing the resultant. Using some vector algebra, we see that the resulting equation can be written as

$$f(s,t) = 2\mathbf{d}^T\Big(\big(\mathbf{x}_s(t) \times \mathbf{x}_t(s,t)\big) \times \big(\mathbf{B}\mathbf{x}(s,t) + \mathbf{b}\big)\Big) = 0.$$

As in the paragraph about the computation of the maximal distance between a point and a quadric intersection curve, this equation has a nice geometric interpretation. The right-hand side of the dot-product is again parallel to the tangent of $\mathcal{C}$ in the point $\mathbf{x}(s,t)$, provided that this point lies on $\mathcal{C}$. Hence, this equation is satisfied for a point on the curve if and only if the tangent in this point is perpendicular to the direction vector $\mathbf{d}$.

In order to find the parameters of the points on $\mathcal{C}$ where the extent attains a local extremum we compute the resultant $r(t) = \mathrm{res}_s(f,g)(t)$. This can be done symbolically using Maple®. The resulting degree is six in the case of a parabolic cylinder and eight in the case of a hyperbolic paraboloid. If the L-quadric is a hyperbolic cylinder, then $r(t)$ is a rational function whose numerator has degree eight.

Local extrema of the extent of a surface Let $\mathcal{S}$ be the surface in question. First, we assume that $\mathcal{S}$ is a quadric. If $\mathcal{S}$ is ruled we do not have to compute the local extrema of its extent. This can be explained as follows. Let $\mathbf{x}$ be a point on $\mathcal{S}$ where the extent with respect to $\mathbf{d}$ is locally maximal. This implies that $\mathbf{d}$ is parallel to the normal of $\mathcal{S}$ in $\mathbf{x}$. Hence, $\mathbf{d}$ is perpendicular to the straight line $\mathcal{L}$ on $\mathcal{S}$ through $\mathbf{x}$. We assume that $\mathbf{x}$ lies inside the face $\mathcal{F}$ that we are considering, because otherwise this point will be neglected. Thus, if we follow the line $\mathcal{L}$ starting in $\mathbf{x}$ we will eventually reach a point $\mathbf{p}$ on an edge $\mathcal{E}$ of $\mathcal{F}$. Since $\mathbf{d}$ is perpendicular to $\mathcal{L}$, it holds that $\mathbf{d}^\mathsf{T}\mathbf{p} = \mathbf{d}^\mathsf{T}\mathbf{x}$. This means we know that the extent of $\mathcal{E}$ is at least $\mathbf{d}^\mathsf{T}\mathbf{x}$ and thus, this value is not a better lower bound for the maximal extent of the face set than the extent of the edge $\mathcal{E}$. The same argument holds if $\mathbf{x}$ is a point on $\mathcal{S}$ where the extent is locally minimal.

Let the quadratic form of $\mathcal{S}$ be given by the matrix $\mathbf{A}_\mathsf{H}$. If $\mathcal{S}$ attains an extremal extent in a point $\mathbf{x} \in \mathcal{S}$, then the normal in $\mathbf{x}$ must be parallel to the direction $\mathbf{d}$, i.e. there is a real value for λ such that $\mathbf{A}\mathbf{x} = \lambda\mathbf{d} - \mathbf{a}$. Multiplying this by adj $\mathbf{A}$ we get

$$\det \mathbf{A} \cdot \mathbf{x} = \operatorname{adj} \mathbf{A} \cdot (\lambda\mathbf{d} - \mathbf{a}). \tag{3.7}$$

If we multiply the quadratic form of $\mathcal{S}$ by $\det \mathbf{A}$ and insert (3.7) we obtain

$$(\lambda\mathbf{d} - \mathbf{a})^\mathsf{T}\overline{\mathbf{A}}(\lambda\mathbf{d} - \mathbf{a}) + 2(\lambda\mathbf{d} - \mathbf{a})^\mathsf{T}\overline{\mathbf{A}}\mathbf{a} + |\mathbf{A}|a_0 = 0.$$

This is a quadratic equation in λ. In order to determine the points of extremal extent we solve this equation for λ and use $\mathbf{A}\mathbf{x} = \lambda\mathbf{d}$ to compute the corresponding point $\mathbf{x}$.

Now, let $\mathcal{S}$ be a torus with center $\mathbf{c}$. Let the unit vector $\mathbf{n}$ be the normal of the main plane and let the radii be R and r with $R > r$. If $\mathbf{d}$ is parallel to $\mathbf{n}$, then the local extrema of the extent are $\mathbf{d}^\mathsf{T}\mathbf{c} \pm r$. Otherwise, the local extrema of $\mathbf{d}^\mathsf{T}\mathbf{x}$ for $\mathbf{x} \in \mathcal{S}$ can be computed by first determining the extremal points on the main circle and then adding the vector $\pm r\mathbf{d}$. In order to compute the extremal points on the main circle we first project the direction $\mathbf{d}$ into the main plane, which gives $\mathbf{d}' = \mathbf{d} - (\mathbf{d}^\mathsf{T}\mathbf{n})\mathbf{n}$. The squared length of this projection is $|\mathbf{d}'|^2 = 1 - (\mathbf{d}^\mathsf{T}\mathbf{n})^2$. The extremal points on the main circle are given by $\mathbf{c} \pm R(\mathbf{d}'/|\mathbf{d}'|)$. Thus, the extrema of the extent function on the torus are the four points

$$\mathbf{c} \pm R\frac{\mathbf{d} - (\mathbf{d}^\mathsf{T}\mathbf{n})\mathbf{n}}{\sqrt{1 - (\mathbf{d}^\mathsf{T}\mathbf{n})^2}} \pm r\mathbf{d}.$$

Partitioning a set of faces

Let $\mathfrak{F}$ denote the set of faces again. Our aim is to partition $\mathfrak{F}$ into two subsets $\mathfrak{F}_1$ and $\mathfrak{F}_2$ such that the sum of the volumina of their bounding volumes is minimal. If the bounding volumes are spheres, then this problem is closely related to the problem of covering a set of points in $\mathbb{R}^d$ by k minimal spheres. This is called the *euclidian* k*-center problem* and is known to be NP-complete for all dimensions $d \geq 2$ ([MS81]). In [Eck99] a branch-and-bound algorithm is described to solve the partitioning problem optimally. This algorithm maintains a set of variable faces which still have to be assigned to one of the subsets $\mathfrak{F}_1$ or $\mathfrak{F}_2$. Initially, all faces in $\mathfrak{F}$ are variable. A branching step consists of choosing one of the variable faces and assigning it to one of the subsets. In this way each node in the decision tree represents a partial covering of $\mathfrak{F}$. A lower bound in such a node is computed by taking the sum of the volumina of the bounding volumes of the current subsets. An upper bound is obtained by assigning the remaining variable faces to the subsets using some heuristic. This algorithm computes the optimal solution to our problem, however the author of the above mentioned thesis states that the running time is very high even for small instances of polygonal faces.

In the case of oriented boxed as bounding volumes, [GLM96] describes a simple heuristic that produces quite good results and is very efficient. Let the set $\mathfrak{F}$ be bounded by the box $\mathcal{B}$. Let for each face $\mathcal{F} \in \mathfrak{F}$ the point $\mathbf{c}_{\mathcal{F}}$ be a reference point associated with that face. The set $\mathfrak{F}$ is partitioned by defining a cutting plane $\mathcal{P}$ that splits the longest edge of $\mathcal{B}$. All faces whose reference points lie on the same side of $\mathcal{P}$ are assigned to the same subset. If the longest axis of $\mathcal{B}$ cannot be subdivided, the second longest axis is chosen. If this also fails, the shortest one is used. According to our experience it makes sense to try each axis of $\mathcal{B}$ to subdivide $\mathfrak{F}$ and finally use the one for which the volume of the resulting bounding volumes is minimal. In addition to the axis of $\mathcal{B}$ we must choose a point $\mathbf{p}$ for the definition of the plane $\mathcal{P}$. The above mentioned publication suggests to take the mean of the reference points

$$\mathbf{p} = \frac{1}{|\mathfrak{F}|} \sum_{\mathcal{F} \in \mathfrak{F}} \mathbf{c}_{\mathcal{F}},$$

which leads to good results. It remains to describe how to obtain the reference points. In [GLM96], all faces are triangles and it is suggested to take the mean of the vertices of a triangle as its reference point. We experienced that also in the case of general polygons this is a good choice. If a face $\mathcal{F}$ is curved, we compute a smallest bounding sphere for $\mathcal{F}$ and take the center of this sphere as the reference point.

In the case of spheres as bounding volumes we simply compute a smallest bounding box for $\mathcal{S}$ and use the techniques that we just described to partition the set of faces.

Static collision test for bounding volumes

In order to use a bounding volume hierarchy to speed up a static collision detection algorithm we must be able to test whether the two bounding volumes intersect. In the case of spheres this is considerably easy. Two spheres intersect if the distance of their centers is smaller than the sum of their radii.

For the case of boxes as bounding volumes we briefly describe the approach presented in [GLM96]. Let $\mathcal{A}$ and $\mathcal{B}$ be two boxes. If we project both boxes onto an axis in space, then each of them forms an interval on that axis. If these intervals are disjoint, then so are the boxes. In this case, the axis is called a *separating axis.* If the intervals overlap then the boxes may or may not be disjoint. The *separating axis theorem* which is proven in [Got96] states that two convex polytopes are disjoint iff there exists a separating axis that is orthogonal to

- a face of one of the two polytopes or
- an edge from each of the two polytopes.

Each box has three unique face orientations and three unique axis directions. It follows that there are 15 potential separating axes, namely three face normals from box $\mathcal{A}$, three

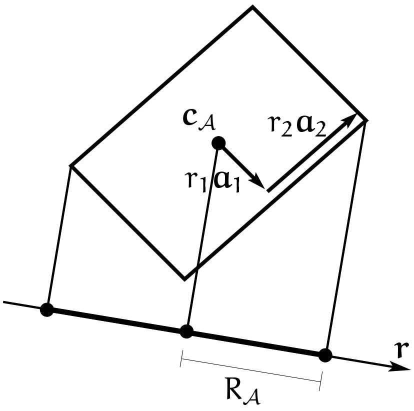


Figure 3.3: Projecting a box onto an axis.

face normals from box $\mathcal{B}$ and nine cross-products of an edge from $\mathcal{A}$ and an edge from $\mathcal{B}$. For each of these 15 axes we check whether it is separating or not. If $\mathcal{A}$ and $\mathcal{B}$ intersect then obviously no separating axis exists. Otherwise, the theorem states that one of these axes is separating. To perform the test for a given axis we project the centers of the boxes onto it, which gives us the centers of the intervals. Then we compute the radii of the intervals. The intervals are disjoint iff the distance between their centers is greater than the sum of their radii.

Let $\mathbf{c}_\mathcal{A}$ and $\mathbf{c}_\mathcal{B}$ be the centers of $\mathcal{A}$ and $\mathcal{B}$, respectively. Moreover, let the unit vectors $\mathbf{a}_i$ and $\mathbf{b}_i$ for $i = 1, 2, 3$ be the axis directions of $\mathcal{A}$ and $\mathcal{B}$, respectively. Let the half-dimension of $\mathcal{A}$ in the direction $\mathbf{a}_i$ be r_i. Similarly, let the half-dimensions of $\mathcal{B}$ be s_i. We denote the direction of the axis onto which the boxes are to be projected by the unit vector $\mathbf{r}$. Figure 3.3 illustrates in two dimensions that the radius of the interval produced by box $\mathcal{A}$ is given by

$$R_\mathcal{A} = \sum_{i=1}^{3} |r_i \mathbf{a}_i^T \mathbf{r}|.$$

The respective expression for $\mathcal{B}$ is similar. The distance of the centers of the intervals is given by $|(\mathbf{c}_\mathcal{B} - \mathbf{c}_\mathcal{A})^T \mathbf{r}|$. Hence, the intervals are disjoint iff

$$|(\mathbf{c}_\mathcal{B} - \mathbf{c}_\mathcal{A})^T \mathbf{r}| > \sum_{i=1}^{3} |r_i \mathbf{a}_i^T \mathbf{r}| + \sum_{i=1}^{3} |s_i \mathbf{b}_i^T \mathbf{r}|. \tag{3.8}$$

The summations simplify if the vector $\mathbf{r}$ is a box axis or a cross-product of box axes. For instance, if $\mathbf{r} = \mathbf{a}_j$, then in the first summation all terms with $i \neq j$ are zero. In [GLM96] a detailed analysis of the number of arithmetic operations can be found that are required to evaluate the above expression.

Dynamic collision test for bounding volumes

In order to speed up a dynamic collision detection algorithm using a bounding volume hierarchy we need to be able to perform a dynamic collision test between the bounding volumes themselves. This means, given linear and angular velocities, we must be able to decide whether two bounding volumes will intersect during a given time interval. In section 3.3 we will show how this can be done for quadratic complexes and for natural quadratic complexes. Since our bounding volumes fall in the class of natural quadratic complexes, we could simply use the results presented there. However, the idea of a bounding volume hierarchy is to have a filtering technique that allows to quickly exclude those parts of the objects that definitely cannot be involved in a collision. Thus, the collision test for the bounding volumes should be as simple and efficient as possible and we do not mind there being some false positive tests, i.e. tests where the algorithm decides that the bounding volumes collide even though they do not. In [Eck99] it is suggested to approximate the swept volume of the bounding spheres or boxes. This is done by computing a bounding volume of the same type that contains the moving volume during the whole time interval. Then, a static collision test between these approximations is performed.

We want to describe a different approach here which is based on interval arithmetic. As we will do in section 3.3, we assume that object $\mathcal{O}_1$ is fixed in time and object $\mathcal{O}_2$ moves with linear velocity $\boldsymbol{v}$ and angular velocity $\boldsymbol{\omega}$. Thus, for every point $\mathbf{p}(t) \in \mathcal{O}_2(t)$ we have

$$\mathbf{p}(t) = \mathbf{R}(t)(\tilde{\mathbf{p}} - \tilde{\mathbf{c}}) + \tilde{\mathbf{c}} + \boldsymbol{v}t, \tag{3.9}$$

where $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{p}}$ are the coordinates of $\mathbf{c}$ and $\mathbf{p}$ at time $t = 0$. The rotation matrix is given by Rodrigues' formula

$$\mathbf{R}(t) = \cos(|\boldsymbol{\omega}|t) \cdot \mathbf{E} + \big(1 - \cos(|\boldsymbol{\omega}|t)\big) \cdot \boldsymbol{w}\boldsymbol{w}^T + \sin(|\boldsymbol{\omega}|t) \cdot \boldsymbol{w}^{\times},$$

with $\boldsymbol{w} = \boldsymbol{\omega}/|\boldsymbol{\omega}|$ if $\boldsymbol{\omega} \neq 0$ and $\boldsymbol{w} = 0$ otherwise. We assume that the time interval that is to be checked for collision is normalized to $[0, 1]$.

Let us first assume that we use spheres as bounding volumes. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be the bounding spheres that are currently to be tested for collision. We denote the centers of $\mathcal{S}_1$ and $\mathcal{S}_2$ by $\mathbf{c}_1$ and $\mathbf{c}_2$, respectively. The radii are denoted by r_1 and r_2. With this notation, the spheres collide iff there is a $t \in [0, 1]$ such that

$$f(t) = |\mathbf{c}_2(t) - \mathbf{c}_1| - r_1 - r_2 < 0.$$

Since $\mathcal{S}_2$ moves synchronously to $\mathcal{O}_2$, we use the relation (3.9) for $\mathbf{c}_2(t)$. Now, we use interval arithmetic to evaluate $[a, b] = f([0, 1])$. In the case $a \geq 0$ we know that the spheres do not collide. Otherwise, we recursively subdivide the time interval until either we decide that there is no collision or the length of the time interval is smaller than a given threshold.

Now, suppose that we use boxes as bounding volumes. The method that we present now is described in [RKC02]. Let $\mathcal{A}$ and $\mathcal{B}$ be the boxes associated with the objects $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. We use the same notations as in the paragraph about static collision detection for bounding boxes. For the 15 potential separating axes we perform the separating axis test (3.8) using interval arithmetic. Since box $\mathcal{B}$ moves synchronously to object $\mathcal{O}_2$, we use relation (3.9) for $\mathbf{c}_{\mathcal{B}}$ and replace the vectors $\mathbf{b}_i$ by $\mathbf{R}(t)\tilde{\mathbf{b}}_i$. Let the left-hand side of (3.8) be denoted by $l(t)$ and the right-hand side by $r(t)$. We start by evaluating both sides for the time interval $[0,1]$ to obtain $[a,b] = l([0,1])$ and $[c,d] = r([0,1])$. If for one of the 15 axes we find that $a - d > 0$, then this axis is separating over the whole interval and we know that the boxes do not collide. If all 15 tests fail, then we have one of the following situations.

- The boxes collide.
- The boxes do not collide, but the separating axis changes over time.
- The boxes do not collide, but the bounds computed by the interval arithmetic are not tight enough.

Again, we recursively subdivide the time interval until its length falls below a given threshold or we find a separating axis.

3.1.2 Space Partitioning

In order to check a scene consisting of n objects for collision, one basically has to test all $O(n^2)$ pairs of objects. Space partitioning is a heuristic that helps to quickly reduce the number of pairs that have to be considered. We will briefly describe an approach published in [Mir97] and [Len00].

Suppose we are given n iso-boxes $\mathcal{B}_1, \ldots, \mathcal{B}_n$ in 3-space and want to find all overlapping pairs $(\mathcal{B}_i, \mathcal{B}_j)$. An iso-box is a box parallel to the coordinate axes. The box $\mathcal{B}_i$ is given by three intervals $I_{ik} = [a_{ik}, b_{ik}]$ for $k = 1,2,3$ and consists of all points $\mathbf{x} \in \mathbb{R}^3$ with $x_k \in I_{ik}$ for $k = 1,2,3$. We define the vectors $\mathbf{a}_i = [a_{i1}, a_{i2}, a_{i3}]^T$ and $\mathbf{b}_i = [b_{i1}, b_{i2}, b_{i3}]^T$. These are the corners of $\mathcal{B}_i$ with minimal and maximal coordinates, respectively. It is a simple observation that $\mathcal{B}_i$ and $\mathcal{B}_j$ overlap if and only if for each $k = 1,2,3$ the intervals I_{ik} and I_{jk} overlap. This gives us a simple static collision test for two iso-boxes. However, in order to find all pairs of overlapping boxes we have to perform $O(n^2)$ such tests. In order to alleviate this all-pairs-weakness, we partition the space into cubical tiles. This is done by a tiling map $\tau : \mathbb{R}^3 \longrightarrow \mathbb{Z}^3$ with resolution ρ which is defined as

$$\tau(\mathbf{x}) = \boldsymbol{w} = \begin{bmatrix} \lfloor x_1/\rho \rfloor \\ \lfloor x_2/\rho \rfloor \\ \lfloor x_3/\rho \rfloor \end{bmatrix}.$$

We identify a tile $\tau^{-1}(\boldsymbol{w})$ with the vector $\boldsymbol{w}$. In order to identify all tiles that intersect an iso-box $\mathcal{B}$ defined by the vectors $\mathbf{a}$ and $\mathbf{b}$ we make the following observation. Let $\mathbf{u} = \tau(\mathbf{a})$ and $\boldsymbol{v} = \tau(\mathbf{b})$ be the tiles containing the vertices of $\mathcal{B}$ with minimal and maximal coordinates, respectively. Then the set of tiles that are intersected by $\mathcal{B}$ is given by

$$\mathcal{W}(\mathcal{B}) = \{\boldsymbol{w} \in \mathbb{Z}^3 \,|\, u_i \leq w_i \leq v_i\}.$$

Clearly, if for two boxes $\mathcal{B}_i$ and $\mathcal{B}_j$ the sets $\mathcal{W}(\mathcal{B}_i)$ and $\mathcal{W}(\mathcal{B}_j)$ are disjoint, then $\mathcal{B}_i$ and $\mathcal{B}_j$ are disjoint, as well. We can now use a hash table to efficiently identify those pairs of iso-boxes that share at least one tile. In this table we represent all tiles that are occupied by at least one box. We use the vector $\boldsymbol{w}$ as key for the hashing. Let us denote the hash function by h. For each tile $\boldsymbol{w} \in \mathcal{W}(\mathcal{B}_i)$ we represent a pointer to $\mathcal{B}_i$ in the hash bucket $h(\boldsymbol{w})$. For any given box $\mathcal{B}_i$ we can now find all boxes $\mathcal{B}_j$ that share a tile with $\mathcal{B}_i$ by inspecting the hash buckets $h(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{W}(\mathcal{B}_i)$.

The quality of this method depends on the choice of the resolution ρ. If ρ is chosen to small, then most non-overlapping pairs of boxes are excluded. However, each box is stored in many hash buckets, which extremely slows down the algorithm. On the other hand, if ρ is chosen to large, then the approach is not very successful in reducing the number of pairs. To solve this problem, we introduce the concept of the *hierarchical space partitioning.* We define the size of an iso-box $\mathcal{B}_i$ as the length of its diagonal, i.e. $s_i = |\mathbf{b}_i - \mathbf{a}_i|$. We want to assign a resolution ρ to each box in such a way that the ratio s_i/ρ is bounded from below and above by α and β, respectively, where $0 < \alpha < 1 \leq \beta$. More precisely, for given values of α and β we are looking for resolutions $\rho_1 > \ldots > \rho_k$ such that for each box $\mathcal{B}_i$ there is an index j_i with

$$\alpha \leq \frac{s_i}{\rho_{j_i}} \leq \beta.$$

Let the minimal and maximal size be denoted by s_{min} and s_{max}, respectively. Then we define the minimal resolution as $\tilde{\rho}_1 = s_{min}/\alpha$. We define the resolution $\tilde{\rho}_{i+1}$ by multiplying $\tilde{\rho}_i$ by β/α, such that we get

$$\tilde{\rho}_i = \left(\frac{\beta}{\alpha}\right)^{i-1} \frac{s_{min}}{\alpha}.$$

By considering the largest box, we find that for the number k of resolutions the relation

$$\log_{\frac{\beta}{\alpha}} \frac{s_{max}}{s_{min}} \leq k \leq 1 + \log_{\frac{\beta}{\alpha}} \frac{s_{max}}{s_{min}}$$

must hold. Thus, we choose

$$k = \left\lceil \log_{\frac{\beta}{\alpha}} \frac{s_{max}}{s_{min}} \right\rceil.$$

In order to have the resolutions in ascending order we set $\rho_i = \tilde{\rho}_{k-i+1}$. We define the box-resolution $\mathtt{res}(i)$ of the iso-box $\mathcal{B}_i$ to be the smallest index j such that

$$\alpha \leq \frac{s_i}{\rho_j} \leq \beta.$$

The idea of the hierarchical space partitioning is to store each box $\mathcal{B}_i$ only in the hash table with resolution $\rho_{\mathtt{res}(i)}$, since the ratio between this resolution and the size of the box is appropriate. We can identify the potentially intersecting pairs of iso-boxes as follows. The k hash tables are initially empty. Now we insert the boxes into the data structure in the order of non-decreasing box-resolutions. Insertion means to store a pointer to the box $\mathcal{B}_i$ in the hash table with resolution $\rho_{\mathrm{res}(i)}$ in all buckets with keys $\boldsymbol{w}$ for $\boldsymbol{w} \in \mathcal{W}(\mathcal{B}_i)$. If we insert box $\mathcal{B}_i$, then we locate it in all hash tables with resolution ρ_l with $l \leq \mathrm{res}(i)$. If there is a pointer to a box $\mathcal{B}_j$ in any visited bucket, then we report the pair $(\mathcal{B}_i, \mathcal{B}_j)$ as potentially overlapping.

We want to bound the number of buckets that we visit if we insert a box $\mathcal{B}_i$. Since $s_i \leq \beta\rho_{\mathrm{res}(i)}$, the box $\mathcal{B}_i$ can intersect at most $(\beta+1)^3$ tiles with edge length $\rho_{\mathrm{res}(i)}$. For box-resolutions $l < \mathrm{res}(i)$ this number cannot be larger, since the tiles with resolution ρ_l have longer edges. We have to locate $\mathcal{B}_i$ in at most k hash tables, hence the number of visited buckets is at most

$$(\beta+1)^3 \left\lceil \log_{\frac{\beta}{\alpha}} \frac{s_{max}}{s_{min}} \right\rceil . \tag{3.10}$$

Since β is a constant, we have proven the following lemma.

Lemma 3.3. *Given a scene with* n *iso-boxes. With hierarchical space partitioning and using perfect hashing we can identify the pairs of potentially overlapping boxes in time* $O(n \log R + c)$, *where* R *is the ratio of the largest to the smallest box and* c *is the number of potentially overlapping pairs.*

Two boxes $\mathcal{B}_i$ and $\mathcal{B}_j$ with $\mathrm{res}(i) \leq \mathrm{res}(j)$ are reported as potentially overlapping if they share a tile at resolution $\rho_{\mathrm{res}(i)}$. The maximal distance between any two points in such a tile is $\sqrt{3}\rho_{\mathrm{res}(i)}$. Hence, for the distance δ between $\mathcal{B}_i$ and $\mathcal{B}_j$ we have

$$\delta \leq \frac{\sqrt{3}\rho_{\mathrm{res}(i)}}{s_i} s_i \leq \frac{1}{\alpha}\sqrt{3}s_i .$$

Conversely, this means that if the distance between two boxes is larger than this value, they will not be reported as potentially overlapping. We can now describe the tradeoffs involved in choosing the constants α and β. By (3.10), the smaller we choose β, the fewer buckets have to be visited when we store a box. The larger we choose α, the smaller is the distance that two non-intersecting boxes must have to not be reported as potentially overlapping. The larger the ratio β/α, the smaller is the number k of resolutions required to store all boxes.

Apparently, the idea of space partitioning makes sense if the number c of potentially overlapping boxes is expected to be small.

In the following we describe how the hierarchical space partitioning can be applied to our collision detection problems.

Application to the collision detection problem

In order to make the above results applicable to the static collision detection problem we must bound each object $\mathcal{O}$ by an iso-box $\mathcal{B}$. We want to make this box large enough such that it does not change its shape if the object rotates. To this end, we compute a bounding sphere for $\mathcal{O}$ with minimal radius that is centered at the center of mass $\mathbf{c}$ of $\mathcal{O}$. We have seen in section 3.1.1 how this can be done. We choose $\mathcal{B}$ to be the smallest iso-box containing this sphere. The corners of $\mathcal{B}$ with minimal and maximal coordinates, respectively, are $\mathbf{c} \pm r[1,1,1]^T$. Since the radius of a bounding sphere does not change if $\mathcal{O}$ rotates, this is also true for the shape of $\mathcal{B}$. Now, the space partitioning approach described above can be applied directly to the set of these boxes. Since the ratio of the largest to the smallest box is a constant, lemma 3.3 says that we can find all pairs of potentially overlapping boxes in time $O(n + c)$.

Now suppose that we want to perform a dynamic collision test. Each object has constant linear and angular velocities and we want to know which pairs of objects potentially collide in the time interval $[0, 1]$. Again, we assume that we know a bounding sphere centered at the center of mass for each object. Moreover, we assume that each point $\mathbf{p}$ of an object $\mathcal{O}$ moves according to equation (3.9). For the center of mass $\mathbf{c}$ this means that it moves along a straight line, namely $\mathbf{c}(t) = \tilde{\mathbf{c}} + \mathbf{v}t$. Hence, the sphere containing $\mathcal{O}$ moves along a straight trajectory. We bound the volume swept by this sphere by an iso-box. Let the radius of the sphere bounding $\mathcal{O}$ be r. Then, we can choose the box whose corners with minimal and maximal coordinates are

$$\mathbf{a} = \begin{bmatrix} \min\{c_1(0), c_1(1)\} - r \\ \min\{c_2(0), c_2(1)\} - r \\ \min\{c_3(0), c_3(1)\} - r \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} \max\{c_1(0), c_1(1)\} + r \\ \max\{c_2(0), c_2(1)\} + r \\ \max\{c_3(0), c_3(1)\} + r \end{bmatrix},$$

respectively. We apply the hierarchical space partitioning method to the set of so defined boxes. The size of the smallest box is at least as large as in the static case. It is quite obvious that the size of the largest box is bounded by a linear function of the maximal velocity $|\mathbf{v}_{max}|$. Thus, lemma 3.3 says that we can find all pairs of potentially colliding objects in time $O(n \log(|\mathbf{v}_{max}| + 1) + c)$.

3.2 Static Collision Test

A static collision detection algorithm tests whether two stationary objects $\mathcal{O}_1$ and $\mathcal{O}_2$ intersect, i.e. whether $\mathcal{O}_1 \cap \mathcal{O}_2 \neq \emptyset$. Therefore we call such a test an intersection test. We first present a generic algorithm and then describe precisely how this algorithm can be specialized for the classes of objects defined in section 2.2.3.

3.2.1 A Generic Algorithm

Now we describe a generic algorithm for the intersection test between two objects. Therefore, we first test whether the boundaries of the objects intersect. If this is the case, then we report a collision. Otherwise, we know that the objects are either disjoint or one of them is contained in the other one. In order to decide in which case we are, we choose a point on the first solid and test whether it is inside or outside the other one, and vice versa. This point-in-solid-test can be done by sending a ray from the query point

Algorithm 1 FACETEST($\mathcal{F}_1, \mathcal{F}_2$)

```
1: for all edges $\mathcal{E}$ of $\mathcal{F}_1$ do
2:   if $\mathcal{E}$ intersects $\mathcal{F}_2$ then
3:     return INTERSECTION
4:   end if
5: end for
6: for all edges $\mathcal{E}$ of $\mathcal{F}_2$ do
7:   if $\mathcal{E}$ intersects $\mathcal{F}_1$ then
8:     return INTERSECTION
9:   end if
10: end for
11: for all loops $\mathcal{L}$ of $\mathcal{S}_1 \cap \mathcal{S}_2$ do
12:   determine a point $\mathbf{p}$ on $\mathcal{L}$
13:   if ($\mathcal{F}_1$ contains $\mathbf{p}$) and ($\mathcal{F}_2$ contains $\mathbf{p}$) then
14:     return INTERSECTION
15:   end if
16: end for
17: return NO_INTERSECTION
```

to infinity and counting the number of intersections with the object. If this number is odd, then the point is inside, otherwise it is outside.

In order to check whether the boundaries of two objects intersect we proceed as follows. For each face $\mathcal{F}_1$ of object 1 and each face $\mathcal{F}_2$ of object 2 we check whether $\mathcal{F}_1$ and $\mathcal{F}_2$ intersect. If we find a pair of intersecting faces then the objects intersect, otherwise they do not. In order to reduce the number of pairs of faces to be checked for intersection we use the techniques described in section 3.1. Algorithm 1 is a generic algorithm for the intersection test between two faces $\mathcal{F}_1$ and $\mathcal{F}_2$ that are embedded on surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. First, all edges of the boundary of $\mathcal{F}_1$ are tested for intersection with $\mathcal{F}_2$ and vice versa (lines 1 to 10). If one of these tests is positive we have obviously found an intersection. Otherwise we know that in the case that $\mathcal{F}_1$ and $\mathcal{F}_2$ intersect there is a loop of the intersection curve between the surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$ that lies completely inside both $\mathcal{F}_1$ and $\mathcal{F}_2$. Thus, it suffices to determine one point on each loop and test whether it is contained in both $\mathcal{F}_1$ and $\mathcal{F}_2$ (lines 11 to 16). If we find one such point, we have found an intersection, otherwise we can decide that $\mathcal{F}_1$ and $\mathcal{F}_2$ do not intersect.

The test in the lines 2 and 7 of algorithm 1 consists of two steps. First the curve on which the edge is embedded is intersected with the surface containing the face. Then for each intersection point (if any) one has to check whether it is contained in both the edge and the face. So altogether the following four tasks have to be performed.

1. Test whether a point on a curve is contained in an edge embedded on that curve (point-on-edge-test).

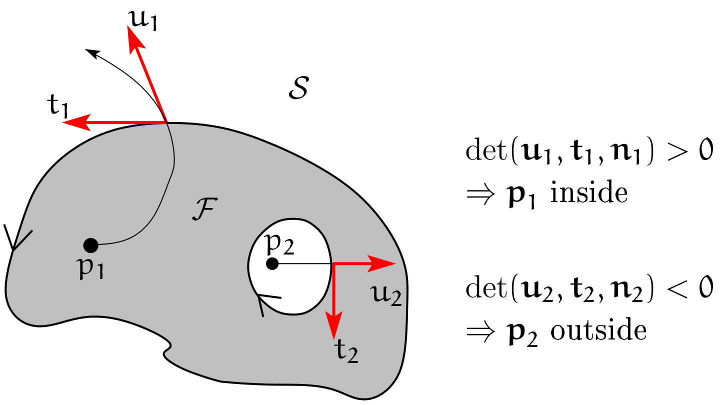

u_1
t_1
$\mathcal{S}$
$\mathcal{F}$
p_1
p_2
u_2
t_2
$\det(\mathbf{u}_1, \mathbf{t}_1, \mathbf{n}_1) > 0$
$\Rightarrow \mathbf{p}_1$ inside
$\det(\mathbf{u}_2, \mathbf{t}_2, \mathbf{n}_2) < 0$
$\Rightarrow \mathbf{p}_2$ outside

Figure 3.4: Point-In-Face-Test

2. Test whether a point on a surface is contained in a face embedded on that surface (point-in-face-test).

3. Compute the points of intersection between a curve and a surface (curve-surface-intersection).

4. Compute (at least) one point on each loop of the intersection curve between two surfaces (surface-surface-intersection).

We will now briefly describe generically how to solve the first two of these tasks. More details as well as descriptions of the remaining two tasks will be given in the next section, where we will have a closer look at special classes of objects.

Point-on-Edge-Test Let the edge $\mathcal{E}$ be embedded on a curve $\mathcal{C}$. Additionally we are given a point $\mathbf{p}$ on $\mathcal{C}$. Since we know a parameterization of $\mathcal{C}$ and since edges are given by a parameter interval $[a, b]$ it is easy to decide whether $\mathcal{E}$ contains $\mathbf{p}$ provided that we know the parameter t of $\mathbf{p}$ on $\mathcal{C}$. We simply have to check the condition $a \leq t \leq b$.

Point-in-Face-Test Let the face $\mathcal{F}$ be embedded on the surface $\mathcal{S}$ and let $\mathbf{p}$ be a point on $\mathcal{S}$. We know that the orientation of the boundary of $\mathcal{F}$ is induced by the orientation of $\mathcal{F}$ (definition 2.5). So, we can use the following approach in order to check whether $\mathbf{p}$ lies inside of outside of $\mathcal{F}$. We send a ray within $\mathcal{S}$ from $\mathbf{p}$ to the boundary of $\mathcal{F}$ (Note that we use the term *ray* even for curved paths). If $\mathcal{F}$ has no boundary, which means that $\mathcal{S}$ is bounded and $\mathcal{F} = \mathcal{S}$, then $\mathbf{p}$ is obviously inside $\mathcal{F}$. Otherwise, we compute the tangent $\mathbf{u}$ to the ray, the tangent $\mathbf{t}$ to the boundary of $\mathcal{F}$ and the normal $\mathbf{n}$ of $\mathcal{S}$ in the point where the ray first hits a bounding edge. Because of the orientation of the boundary the point $\mathbf{p}$ lies inside $\mathcal{F}$ iff $\det(\mathbf{u}, \mathbf{t}, \mathbf{n}) > 0$. Figure 3.4 illustrates the two cases. The normal in the figure is supposed to point out of the drawing plane. The face sketched there has two loops. The point $\mathbf{p}_1$ lies in the interior, $\mathbf{p}_2$ in the exterior of $\mathcal{F}$.

For unbounded surfaces $\mathcal{S}$ there is another approach. We can define a ray within $\mathcal{S}$ from $\mathbf{p}$ to infinity. We count the intersection points between this ray and the boundary of $\mathcal{F}$. If none of these points is a tangential intersection point, then $\mathbf{p}$ lies inside $\mathcal{F}$, iff this number is odd. In both approaches we have to perform a point-on-edge-test whenever we intersect a ray with the boundary of $\mathcal{F}$.

We perform the point-in-face-test in 3-space rather than in the parameter space of the particular surface. The reason for this is that the curves that we allow as boundaries of the faces have a relatively easy representation in 3-space. If we change to the parameter space then it is not guaranteed that this is still true. A conic section in 3-space for instance is in general not a conic section in the parameter space of the surface.

In the next section we will describe how the generic algorithm for static collision detection can be specialized for the classes of objects that we defined in section 2.2.3. This means that we have to describe how the four tasks mentioned above can be performed for these specific object classes. For a subclass of the quadratic complexes, namely the natural quadratic complexes, we describe two evaluation examples.

3.2.2 Specialization for Quadratic Complexes

We will now prove the following theorem.

Theorem 3.4. *The static collision detection problem for QCs can be reduced to solving polynomial equations of degree at most four.*

We will show that for QCs, the tasks described in the previous section can be performed by solving only polynomial equations of degree at most four. This means that there are closed form solutions for all occurring equations. Hence, we can use Cardano's and Ferrari's formulae to solve them. Experiments showed that the accuracy of the solutions found by these formulae is sufficient. Moreover, we experienced that the numerical methods we used for comparison were at least by a factor of ten slower.

Point-on-Edge-Test

We have seen that this task is trivial, provided that we know the curve parameter of the query point. We show now how to compute this parameter for a point $\mathbf{p}$ on a conic $\mathcal{C}$. If $\mathcal{C}$ is a straight line (as a special case of a conic), then we have a parameterization $\mathbf{x}(t) = \mathbf{a} + t\mathbf{u}$ and the parameter of $\mathbf{p}$ is simply

$$t_0 = \frac{(\mathbf{p} - \mathbf{a})^T \mathbf{u}}{\mathbf{u}^2}.$$

Otherwise, $\mathcal{C}$ is given by a parameterization of the form

$$\mathbf{x}(t) = \mathbf{c} + \mathbf{u}x_1(t) + \mathbf{v}x_2(t),$$

where $\mathbf{u}$ and $\mathbf{v}$ are orthogonal unit vectors and with $x_1(t)$ and $x_2(t)$ as in (2.6). The normal of the plane containing $\mathcal{C}$ is $\mathbf{n} = \mathbf{u} \times \mathbf{v}$. We define the rotation matrix $\mathbf{R} =$

$[\mathbf{u}, \mathbf{v}, \mathbf{n}]$. If we set

$$\begin{aligned} \tilde{\mathbf{x}}(t) &= \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad \text{and} \\ \begin{bmatrix} \tilde{\mathbf{p}} \\ 0 \end{bmatrix} &= \begin{bmatrix} \tilde{p}_1 \\ \tilde{p}_2 \\ 0 \end{bmatrix} = \mathbf{R}^T(\mathbf{p}-\mathbf{c}) = \begin{bmatrix} \mathbf{u}^T(\mathbf{p}-\mathbf{c}) \\ \mathbf{v}^T(\mathbf{p}-\mathbf{c}) \\ 0 \end{bmatrix}, \end{aligned}$$

then we have $\mathbf{x}(t_0) = \mathbf{p}$ iff $\tilde{\mathbf{x}}(t_0) = \tilde{\mathbf{p}}$.

Case 1: $\mathcal{C}$ is an ellipse. If the edge consists of the whole ellipse, then we do not have to compute the parameter of $\mathbf{p}$, since it certainly lies on the edge. By the parameterization (2.8) we have

$$x_1(t) = a\frac{\gamma(1-t^2) - 2\delta t}{1+t^2} \quad \text{and} \quad x_2(t) = b\frac{\delta(1-t^2) + 2\gamma t}{1+t^2}.$$

If $\tilde{\mathbf{p}} = [-a\gamma, -b\delta, 0]^T$, then it does not lie on the edge, since the parameterization of $\mathcal{C}$ was chosen in that way. We have

$$\begin{aligned} \tilde{x}_1(t_0) = \tilde{p}_1 &\Leftrightarrow a(-\gamma t_0^2 - 2\delta t_0 + \gamma) = \tilde{p}_1(1+t_0^2) \\ &\Leftrightarrow t_0 = \frac{-a\delta \pm \sqrt{a^2 - \tilde{p}_1^2}}{a\gamma + \tilde{p}_1}. \end{aligned}$$

Here we used the fact that for the values γ and δ the relation $\gamma^2 + \delta^2 = 1$ holds. Similarly, we obtain

$$\tilde{x}_2(t_0) = \tilde{p}_2 \Leftrightarrow t_0 = \frac{b\gamma \pm \sqrt{b^2 - \tilde{p}_2^2}}{b\delta + \tilde{p}_2}.$$

Case 2: $\mathcal{C}$ is a hyperbola. Then, $x_1(t) = \sigma a \frac{1+t^2}{1-t^2}$ and $x_2 = b\frac{2t}{1-t^2}$ for $\sigma = \pm 1$.

$$\begin{aligned} \tilde{\mathbf{x}}(t_0) = \tilde{\mathbf{p}} &\Leftrightarrow \sigma a(1+t_0^2) = \tilde{p}_1(1-t_0^2) \quad \wedge \quad 2bt_0 = \tilde{p}_2(1-t_0^2) \\ &\Leftrightarrow t_0 = \pm\sqrt{\frac{\tilde{p}_1 - \sigma a}{\tilde{p}_1 + \sigma a}} \quad \wedge \quad 2bt_0 = \tilde{p}_2(1-t_0^2). \end{aligned}$$

Case 3: $\mathcal{C}$ is a parabola. Then $x_1(t) = t$ and $x_2(t) = at^2$. In this case we have $\tilde{\mathbf{x}}(t_0) = \tilde{\mathbf{p}}$ iff $t_0 = \tilde{\mathbf{p}}_1$. In all cases we can compute the parameter of $\mathbf{p}$ by solving only linear and quadratic equations.

Before we continue with the point-in-face-test, we show how the points of intersection between a straight line and a conic $\mathcal{C}$ can be computed by solving only linear and quadratic equations. Let the straight line be given by the parameterization $\mathbf{l}(\lambda) = \mathbf{p} + \lambda\mathbf{r}$. If $\mathcal{C}$ is also a straight line given by $\mathbf{x}(t) = \mathbf{a} + t\mathbf{u}$, then we assume, that $\mathbf{r} \times \mathbf{u} \neq 0$, i.e. the lines are not parallel. We have an intersection point $\mathbf{l}(\lambda_0)$ between the two lines iff

$$\mathbf{p} + \lambda_0\mathbf{r} = \mathbf{a} + t\mathbf{u} + \nu(\mathbf{r} \times \mathbf{u})$$

has a solution with $\nu = 0$. This is a linear system of equations with the unknowns λ_0, t and ν and by Cramer's rule we find

$$\begin{aligned} \nu = 0 \;\Leftrightarrow\; & \mathbf{r}^T(\mathbf{u} \times (\mathbf{a} - \mathbf{p})) = 0, \\ \lambda_0 \;=\; & \frac{(\mathbf{a} - \mathbf{p})^T(\mathbf{u} \times (\mathbf{r} \times \mathbf{u}))}{(\mathbf{r} \times \mathbf{u})^2}. \end{aligned}$$

If $\mathcal{C}$ is not a straight line, then it is given by a parameterization of the form $\mathbf{x}(t) = \mathbf{c} + \mathbf{u}x_1(t) + \mathbf{v}x_2(t)$. We set $\mathbf{n} = \mathbf{u} \times \mathbf{v}$ and $\mathbf{R} = [\mathbf{u}, \mathbf{v}, \mathbf{n}]$. We also know $\mathbf{A} \in \mathbb{R}^{2\times 2}$, $\mathbf{a} \in \mathbb{R}^2$ and $a_0 \in \mathbb{R}$ such that

$$[x_1, x_2]\mathbf{A} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + 2[x_1, x_2]\mathbf{a} + a_0 = 0.$$

Case 1: $\mathbf{n}^T\mathbf{r} = 0$ and $\mathbf{n}^T(\mathbf{p} - \mathbf{c}) \neq 0$. Then the line does not intersect $\mathcal{C}$ because it lies in a plane that is parallel but not equal to the plane containing $\mathcal{C}$.

Case 2: $\mathbf{n}^T\mathbf{r} = \mathbf{n}^T(\mathbf{p} - \mathbf{c}) = 0$. Then the line lies in the same plane as $\mathcal{C}$. We express this line in coordinates of that plane by defining

$$\begin{bmatrix} \tilde{\mathbf{l}}(\lambda) \\ 0 \end{bmatrix} := \mathbf{R}^T(\mathbf{l}(\lambda) - \mathbf{c}) = \begin{bmatrix} \mathbf{u}^T(\mathbf{p} - \mathbf{c}) + \lambda\mathbf{u}^T\mathbf{r} \\ \mathbf{v}^T(\mathbf{p} - \mathbf{c}) + \lambda\mathbf{v}^T\mathbf{r} \\ 0 \end{bmatrix}.$$

Then, $\mathbf{l}(\lambda_0)$ is an intersection point between the straight line and $\mathcal{C}$, iff λ_0 is a solution of the quadratic equation $\tilde{\mathbf{l}}(\lambda)^T\mathbf{A}\tilde{\mathbf{l}}(\lambda) + 2\tilde{\mathbf{l}}(\lambda)^T\mathbf{a} + a_0 = 0$.

Case 3: $\mathbf{n}^T\mathbf{r} \neq 0$. Then, the line intersects the plane containing $\mathcal{C}$ in exactly one point $\mathbf{q} = \mathbf{l}(\lambda_0)$:

$$\mathbf{n}^T(\mathbf{l}(\lambda_0) - \mathbf{c}) = 0 \quad \Leftrightarrow \quad \lambda_0 = \frac{\mathbf{n}^T(\mathbf{c} - \mathbf{p})}{\mathbf{n}^T\mathbf{r}}.$$

We express $\mathbf{q}$ in coordinates of the plane containing $\mathcal{C}$ by setting $[\tilde{\mathbf{q}}^T, 0]^T = \mathbf{R}^T(\mathbf{q} - \mathbf{c})$. Then, $\mathbf{q}$ lies on $\mathcal{C}$ iff $\tilde{\mathbf{q}}^T\mathbf{A}\tilde{\mathbf{q}} + 2\tilde{\mathbf{q}}^T\mathbf{a} + a_0 = 0$.

Point-in-Face-Test

Let $\mathcal{F}$ be a face that lies on a quadric $\mathcal{S}$ defined by the matrix $\mathbf{A}_H$ and let $\mathbf{p}$ be a point on $\mathcal{S}$. We show, that we can decide whether $\mathbf{p}$ lies in $\mathcal{F}$ by solving only linear and quadratic equations. We make a case distinction according to the type of $\mathcal{S}$.

Case 1: $\mathcal{S}$ is an ellipsoid or an elliptic cylinder. By 2.3.1 we may assume that $\mathcal{S}$ is given in the form $\mathbf{x}^T\mathbf{A}\mathbf{x} = \sigma$ for $\sigma = \pm 1$ (The sign depends on the orientation of $\mathcal{S}$). If $\mathcal{S}$ is an ellipsoid and $\mathcal{F}$ has no boundary, then obviously $\mathbf{p} \in \mathcal{F}$. Otherwise we choose a point $\mathbf{q}$ on the boundary of $\mathcal{F}$ and show that we can find a ray in $\mathcal{S}$ from $\mathbf{p}$ to $\mathbf{q}$. We compute the first point where this ray intersects the boundary of $\mathcal{F}$ and use the determinant criterion described in 3.2.1 to decide whether $\mathbf{p}$ lies in $\mathcal{F}$ or not. The following lemma is useful to determine such a ray.

Lemma 3.5. *Let the quadratic form* $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - 1 = 0$ *define an ellipsoid or an elliptic cylinder and let* $\mathbf{p}$ *and* $\mathbf{q}$ *be points with* $\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{p} = \mathbf{q}^\mathsf{T}\mathbf{A}\mathbf{q} = 1$*. Then it holds that* $|\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q}| \leq 1$*.*

Proof. First, let $\mathbf{A}$ define an ellipsoid. Then, $\det \mathbf{A} > 0$ and $\mathbf{A}$ is positive definite. If $\mathbf{p} = \pm\mathbf{q}$, then the claim is obviously true. Thus, we assume that $\mathbf{p} \neq \pm\mathbf{q}$. Then we know that $\mathbf{p}$ and $\mathbf{q}$ are linearly independent because otherwise the line through $\mathbf{p}, -\mathbf{p}, \mathbf{q}$ and $-\mathbf{q}$ would intersect the ellipsoid in four points. We define the vector $\mathbf{n} = \mathbf{A}^{-1}(\mathbf{p} \times \mathbf{q})$ and the matrix $\mathbf{T} = [\mathbf{p}, \mathbf{q}, \mathbf{n}]$. The determinant $(\mathbf{p} \times \mathbf{q})^\mathsf{T}\mathbf{A}^{-1}(\mathbf{p} \times \mathbf{q})$ of $\mathbf{T}$ is positive because $\mathbf{A}^{-1}$ is positive definite. Clearly, $\det(\mathbf{T}^\mathsf{T}\mathbf{A}\mathbf{T}) > 0$.

$$\begin{aligned}\det(\mathbf{T}^\mathsf{T}\mathbf{A}\mathbf{T}) &= \det\begin{bmatrix} 1 & \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q} & 0 \\ \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q} & 1 & 0 \\ 0 & 0 & \mathbf{n}^\mathsf{T}\mathbf{A}\mathbf{n} \end{bmatrix} \\ &= \mathbf{n}^\mathsf{T}\mathbf{A}\mathbf{n} \cdot \big(1 - (\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})^2\big).\end{aligned}$$

Hence, $\det(\mathbf{T}^\mathsf{T}\mathbf{A}\mathbf{T}) > 0$ is equivalent to $|\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q}| < 1$ and the lemma follows. Next, let $\mathbf{A}$ define an elliptic cylinder. Then, two eigenvalues of $\mathbf{A}$ are positive and the third one is zero. Let $\boldsymbol{w}$ be a unit eigenvector of $\mathbf{A}$ associated with the eigenvalue zero. If we define $\tilde{\mathbf{A}} = \mathbf{A} + \boldsymbol{w}\boldsymbol{w}^\mathsf{T}$ and the vectors $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}^\mathsf{T}\boldsymbol{w} \cdot \boldsymbol{w}$ and $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}^\mathsf{T}\boldsymbol{w} \cdot \boldsymbol{w}$, then $\tilde{\mathbf{A}}$ is positive definite and we easily verify that $\tilde{\mathbf{p}}^\mathsf{T}\tilde{\mathbf{A}}\tilde{\mathbf{p}} = \tilde{\mathbf{q}}^\mathsf{T}\tilde{\mathbf{A}}\tilde{\mathbf{q}} = 1$. Thus, we are in the ellipsoid case and we already know that $\tilde{\mathbf{p}}^\mathsf{T}\tilde{\mathbf{A}}\tilde{\mathbf{q}} \leq 1$. But it is also easy to verify that $\tilde{\mathbf{p}}^\mathsf{T}\tilde{\mathbf{A}}\tilde{\mathbf{q}} = \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q}$. □

Obviously, this lemma is still true if we change the orientation of the ellipsoid or elliptic cylinder.

By looking more closely at this proof we see that $|\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q}| = 1$ means in the case of $\mathcal{S}$ being an ellipsoid that $\mathbf{p} = \pm\mathbf{q}$ and if $\mathcal{S}$ is an elliptic cylinder that $\mathbf{p}$ and $\mathbf{q}$ lie on the same straight line or on opposite straight lines on $\mathcal{S}$. We choose the point $\mathbf{q}$ randomly in such a way, that this is not the case. Now we are ready to compute a ray in $\mathcal{S}$ from $\mathbf{p}$ to $\mathbf{q}$. Therefore, we define a plane $\mathcal{P}$ containing the points $\mathbf{p}$ and $\mathbf{q}$ by the parameterization $\mathbf{x}(s,t) = \mathbf{p} + s(\mathbf{q} - \mathbf{p}) + t(\mathbf{q} + \mathbf{p})$. As $\mathbf{p}$ and $\mathbf{q}$ are linearly independent, the vectors $\mathbf{q} - \mathbf{p}$ and $\mathbf{q} + \mathbf{p}$ are linearly independent, as well. Inserting $\mathbf{x}(s,t)$ into the quadratic form of $\mathcal{S}$ we obtain the quadratic equation

$$(\sigma + \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})t^2 + (\sigma + \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})t + s(s-1)(\sigma - \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q}) = 0.$$

The discriminant of this equation w.r.t. t is $D(s) = (\sigma + \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})^2 + 4s(s-1)\big((\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})^2 - 1\big)$. Since $(\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})^2 - 1 < 0$ we have $D(s) > 0$ for $0 \leq s \leq 1$. Thus, for all values of s in the interval $[0, 1]$ we find values t that satisfy the quadratic equation. One solution is

$$t(s) = -\frac{1}{2} + \frac{\sigma\sqrt{D(s)}}{2(\sigma + \mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{q})}.$$

Since $\mathrm{sign}(1+\mathbf{p}^T\mathbf{A}\mathbf{q})=\sigma$, we have $t(0)=t(1)=0$ and hence $\mathbf{x}(0,t(0))=\mathbf{p}$ and $\mathbf{x}(1,t(1))=\mathbf{q}$. So we define the ray $\mathcal{L}$ from $\mathbf{p}$ to $\mathbf{q}$ by $\mathbf{l}(s)=\mathbf{x}(s,t(s))$ for $s\in[0,1]$. Since we chose $\mathbf{q}$ randomly, we can assume that $\mathcal{L}$ does not intersect any bounding edge of $\mathcal{F}$ tangentially.

Now, we have to find the intersection point $\mathbf{l}(s_0)$ between $\mathcal{L}$ and the boundary of $\mathcal{F}$ with minimal value $s_0\in(0,1]$. This is equivalent to finding the intersection point $\mathbf{x}(s_0,t_0)$ between the plane $\mathcal{P}$ and the boundary of $\mathcal{F}$ with minimal value $s_0\in(0,1]$ and $t_0\geq-\frac{1}{2}$. Let $\mathcal{E}$ be an edge of $\mathcal{F}$ that is embedded on a conic $\mathcal{C}$ with parameterization $\mathbf{c}(\lambda)$. In order to find the intersection points between $\mathcal{C}$ and $\mathcal{P}$ we solve the system of equations

$$\mathbf{p}+s\mathbf{u}+t\mathbf{v}+r\mathbf{n}_{\mathcal{P}}=\mathbf{c}(\lambda),$$

where $\mathbf{u}=\mathbf{q}-\mathbf{p}$, $\mathbf{v}=\mathbf{q}+\mathbf{p}$ and $\mathbf{n}_{\mathcal{P}}=\mathbf{u}\times\mathbf{v}$ is the normal of $\mathcal{P}$. We are interested in solutions of this system with $r=0$. Thus we have to solve the equation $\mathbf{n}_{\mathcal{P}}^T(\mathbf{c}(\lambda)-\mathbf{p})=0$. Because of the parameterizations (2.6) and the substitutions (2.7) and (2.9) this is a quadratic equation in λ. For each solution of this equation we obtain a point $\mathbf{x}=\mathbf{c}(\lambda)$, for which we perform a point-on-edge-test to check whether $\mathbf{x}\in\mathcal{E}$. By Cramer's rule, for each such point our system of equations has the solution

$$s=\frac{(\mathbf{x}-\mathbf{p})^T(\mathbf{v}\times\mathbf{n}_{\mathcal{P}})}{\mathbf{n}_{\mathcal{P}}^2},\quad t=\frac{(\mathbf{x}-\mathbf{p})^T(\mathbf{n}_{\mathcal{P}}\times\mathbf{u})}{\mathbf{n}_{\mathcal{P}}^2}.$$

If $s\in(0,1]$ and $t>-\frac{1}{2}$, then $\mathbf{x}$ lies on $\mathcal{L}$ between $\mathbf{p}$ and $\mathbf{q}$ (or is equal to $\mathbf{q}$). We iterate over all edges of $\mathcal{F}$ to determine that point $\mathbf{x}$ with minimal s-parameter. Then we use the determinant criterion described in 3.2.1 to decide whether $\mathbf{p}\in\mathcal{F}$ or not. Therefore we need the tangent $\mathbf{t}_{\mathcal{L}}$ of the ray in $\mathbf{x}$, the tangent $\mathbf{t}_{\mathcal{E}}$ of the edge $\mathcal{E}$ of $\mathcal{F}$ on which $\mathbf{x}$ lies and the normal $\mathbf{n}_{\mathcal{F}}$ of $\mathcal{F}$ in $\mathbf{x}$. The tangent $\mathbf{t}_{\mathcal{E}}$ can simply be computed as the derivation of the parameterization of the curve containing $\mathcal{E}$ and by taking care of the orientation of the coedge of $\mathcal{E}$ that belongs to a loop of $\mathcal{F}$. The normal of $\mathcal{F}$ is given by $\mathbf{n}_{\mathcal{F}}=\mathbf{A}\mathbf{x}$. Finally, the tangent of $\mathcal{L}$ in $\mathbf{x}$ is $\mathbf{t}_{\mathcal{L}}=\mathbf{l}'(s)=\mathbf{q}-\mathbf{p}+t'(s)(\mathbf{q}+\mathbf{p})$.

In each of the following cases we compute a ray in $\mathcal{S}$ from $\mathbf{p}$ to infinity and count the number of intersections between the ray and the boundary of $\mathcal{F}$. The ray will always been chosen randomly, so that we can assume that it will not intersect any bounding edge of $\mathcal{F}$ tangentially.

Case 2: $\mathcal{S}$ is a one- or two-sheet hyperboloid or a cone. By 2.3.1 we can assume that $\mathcal{S}$ is given by $\mathbf{x}^T\mathbf{A}\mathbf{x}+a_0=0$ and that we know the eigenvalues ξ_1,ξ_2 and ξ_3 of $\mathbf{A}$ as well as unit eigenvectors $\mathbf{u},\mathbf{v}$ and $\mathbf{w}$ associated with ξ_1,ξ_2 and ξ_3, respectively. W.l.o.g. we assume that $\xi_1,\xi_2>0$ and $\xi_3<0$. We want do define a random ray $\mathcal{L}$ in $\mathcal{S}$ from $\mathbf{p}$ to infinity and count the number of intersections between $\mathcal{L}$ and the boundary of $\mathcal{F}$. Therefore, we choose random values λ,μ (not both zero) and define $\mathbf{r}=\lambda\mathbf{u}+\mu\mathbf{v}$. The ray $\mathcal{L}$ will be a part of the intersection curve between $\mathcal{S}$ and the plane $\mathcal{P}$ parameterized as $\mathbf{x}(s,t)=\mathbf{p}+s\mathbf{w}+t\mathbf{r}$. Inserting this parameterization into the quadratic form of $\mathcal{S}$

we obtain the quadratic equation

$$\mathbf{r}^T\mathbf{A}\mathbf{r}t^2 + 2\mathbf{p}^T\mathbf{A}\mathbf{r}t + s(\boldsymbol{w}^T\mathbf{A}\boldsymbol{w}s + 2\mathbf{p}^T\mathbf{A}\boldsymbol{w}) = 0.$$

The discriminant of this equation w.r.t. t is

$$D(s) = 4\big((\mathbf{p}^T\mathbf{A}\mathbf{r})^2 - s\mathbf{r}^T\mathbf{A}\mathbf{r}(\boldsymbol{w}^T\mathbf{A}\boldsymbol{w}s + 2\mathbf{p}^T\mathbf{A}\boldsymbol{w})\big).$$

We observe that $D(0) = 4(\mathbf{p}^T\mathbf{A}\mathbf{r})^2 \geq 0$ and that because of $-\mathbf{r}^T\mathbf{A}\mathbf{r} \cdot \boldsymbol{w}^T\mathbf{A}\boldsymbol{w} > 0$ there is a minimum of $D(s)$ at $s_{min} = -(\mathbf{p}^T\mathbf{A}\boldsymbol{w})/(\boldsymbol{w}^T\mathbf{A}\boldsymbol{w})$. Since $\boldsymbol{w}^T\mathbf{A}\boldsymbol{w} < 0$, the sign of s_{min} is equal to the sign of $\mathbf{p}^T\mathbf{A}\boldsymbol{w}$. We set $\sigma = -\text{sign}(\mathbf{p}^T\mathbf{A}\boldsymbol{w})$ if this value is non-zero and $\sigma = 1$ otherwise. Then, for each value of $s \geq 0$ it holds that $D(\sigma s) \geq 0$ and hence, there are values of t such that t and σs solve the quadratic equation. One solution is

$$t(\sigma s) = \frac{-\mathbf{p}^T\mathbf{A}\mathbf{r} + \tau\sqrt{\frac{1}{4}D(\sigma s)}}{\mathbf{r}^T\mathbf{A}\mathbf{r}}, \tag{3.11}$$

where we set $\tau = \text{sign}(\mathbf{p}^T\mathbf{A}\mathbf{r})$ if that value is non-zero and $\tau = 1$ otherwise. With this choice of τ we ensure that $t(0) = 0$ and hence $\mathbf{x}(0, t(0)) = \mathbf{p}$. We define the ray $\mathcal{L}$ from $\mathbf{p}$ to infinity by $\mathbf{l}(s) = \mathbf{x}(\sigma s, t(\sigma s))$ for $s \geq 0$.

Now we must determine the intersection points $\mathbf{l}(s)$ between $\mathcal{L}$ and the boundary of $\mathcal{F}$ with $s > 0$. This is equivalent to finding the intersection points $\mathbf{x}(\sigma s, t)$ between the plane $\mathcal{P}$ and the boundary of $\mathcal{F}$ with $s > 0$ and

- $t \geq -(\mathbf{p}^T\mathbf{A}\mathbf{r})/(\mathbf{r}^T\mathbf{A}\mathbf{r})$ if $\tau = 1$ and
- $t \leq -(\mathbf{p}^T\mathbf{A}\mathbf{r})/(\mathbf{r}^T\mathbf{A}\mathbf{r})$ if $\tau = -1$.

If the number of these points is odd, then $\mathbf{p}$ lies on $\mathcal{F}$ and otherwise it does not. The computation of the intersection points works analogously to case 1.

Case 3: $\mathcal{S}$ is an elliptic or hyperbolic paraboloid. We can assume that $\mathcal{S}$ is given by $\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{a} = 0$, where $\mathbf{A}\mathbf{a} = \mathbf{0}$. As in the previous case we can assume that we know that $\mathbf{A}$ has eigenvalues ξ_1, ξ_2 and 0 with associated unit eigenvectors $\mathbf{u}, \mathbf{v}$ and $\boldsymbol{w}$. Thus, the vectors $\boldsymbol{w}$ and $\mathbf{a}$ are parallel. Again, we define a random ray $\mathcal{L}$ from $\mathbf{p}$ to infinity and count the number of intersections between $\mathcal{L}$ and the boundary of $\mathcal{F}$. Like before, the ray is a part of the intersection curve between a plane $\mathcal{P}$ and $\mathcal{S}$. The plane is defined as in the previous case, namely by the parameterization $\mathbf{x}(s,t) = \mathbf{p} + s\boldsymbol{w} + t\mathbf{r}$, where $\mathbf{r} = \lambda\mathbf{u} + \mu\mathbf{v}$ for randomly chosen values λ and μ. Inserting this into the quadratic form of $\mathcal{S}$ we obtain the quadratic equation

$$\mathbf{r}^T\mathbf{A}\mathbf{r}t^2 + 2\mathbf{p}^T\mathbf{A}\mathbf{r}t + 2s\boldsymbol{w}^T\mathbf{a} = 0.$$

The value $\mathbf{r}^T\mathbf{A}\mathbf{r}$ is zero iff $\lambda^2\xi_1 + \mu^2\xi_2 = 0$. But the set of (λ, μ) fulfilling this equation forms two intersecting lines in the plane, hence the probability of randomly chosen real values to belong to this set is zero. We notice that the above equation is linear in the parameter s and $\mathbf{w}^T\mathbf{a} \neq 0$. Thus, we solve for s in this case and obtain

$$s(t) = \frac{\mathbf{r}^T\mathbf{A}\mathbf{r}t^2 + 2\mathbf{p}^T\mathbf{A}\mathbf{r}t}{2\mathbf{w}^T\mathbf{a}}.$$

Since $s(0) = 0$, it holds that $\mathbf{x}(s(0), 0) = \mathbf{p}$. We define the ray $\mathcal{L}$ from $\mathbf{p}$ to infinity by $\mathbf{l}(t) = \mathbf{x}(s(t), t)$ for $t \geq 0$. Finding the intersection points between $\mathcal{L}$ and the boundary of $\mathcal{F}$ works analogously to the previous case.

Case 4: $\mathcal{S}$ is a hyperbolic or parabolic cylinder. Again, we assume that $\mathcal{S}$ is given by the quadratic form $\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{x}^T\mathbf{a} + a_0 = 0$, where $\mathbf{A}\mathbf{a} = \mathbf{0}$. As before, we also assume that we know the eigenvalues ξ_1, ξ_2 and 0 of $\mathbf{A}$ with associated unit eigenvectors $\mathbf{u}, \mathbf{v}$ and $\mathbf{w}$, respectively. W.l.o.g., we assume $\xi_1 \geq 0$ and $\xi_2 < 0$. We define a ray $\mathcal{L}$ in $\mathcal{S}$ from $\mathbf{p}$ to infinity as a part of the intersection curve between $\mathcal{S}$ and the plane $\mathcal{P}$ defined by the parameterization $\mathbf{x}(s, t) = \mathbf{p} + s\mathbf{u} + t\mathbf{r}$. The vector $\mathbf{r}$ is defined as $\lambda\mathbf{v} + \mu\mathbf{w}$ for random values λ and μ. With this definition of $\mathbf{r}$ we have $\mathbf{r}^T\mathbf{A}\mathbf{r} < 0$. Inserting the parameterization of $\mathcal{P}$ into the quadratic form of $\mathcal{S}$ we obtain the quadratic equation

$$\mathbf{r}^T\mathbf{A}\mathbf{r}t^2 + 2(\mathbf{p}^T\mathbf{A}\mathbf{r} + \mathbf{r}^T\mathbf{a})t + s(s\mathbf{u}^T\mathbf{A}\mathbf{u} + 2\mathbf{p}^T\mathbf{A}\mathbf{u} + 2\mathbf{u}^T\mathbf{a}) = 0.$$

1. $\xi_1 > 0$, i.e. $\mathcal{S}$ is a hyperbolic cylinder. Then, by 2.3.1 we can assume that $\mathbf{a} = \mathbf{0}$. The discriminant of the quadratic equation w.r.t. t is $D(s) = 4\big((\mathbf{p}^T\mathbf{A}\mathbf{r})^2 - s\mathbf{r}^T\mathbf{A}\mathbf{r}(s\mathbf{u}^T\mathbf{A}\mathbf{u} + 2\mathbf{p}^T\mathbf{A}\mathbf{u})\big)$. Since $\mathbf{r}^T\mathbf{A}\mathbf{r} \cdot \mathbf{u}^T\mathbf{A}\mathbf{u} < 0$, the discriminant has a minimum at $s_{min} = -(\mathbf{p}^T\mathbf{A}\mathbf{u})/(\mathbf{u}^T\mathbf{A}\mathbf{u})$. The sign of s_{min} is equal to the sign of $-\mathbf{p}^T\mathbf{A}\mathbf{u}$. We set $\sigma = \text{sign}(\mathbf{p}^T\mathbf{A}\mathbf{u})$. Then, for each value of $s \geq 0$ it holds that $D(\sigma s) \geq 0$ and hence, there are values of t such that t and σs solve the quadratic equation. Equation (3.11) is again a solution of that equation, where $\tau = \text{sign}(\mathbf{p}^T\mathbf{A}\mathbf{r})$ ensures that $t(0) = 0$ and hence $\mathbf{x}(0, t(0)) = \mathbf{p}$. As before, we define the ray $\mathcal{L}$ by $\mathbf{l}(s) = \mathbf{x}(\sigma s, t(\sigma s))$ for $s \geq 0$. Finding the intersection points between $\mathcal{L}$ and the boundary of $\mathcal{F}$ works analogously to the previous cases.

2. $\xi_1 = 0$, i.e. $\mathcal{S}$ is a parabolic cylinder. By 2.3.1 we can assume that $\mathbf{a}$ is a multiple of $\mathbf{u}$ and thus $\mathbf{u}^T\mathbf{a} \neq 0$ and $\mathbf{r}^T\mathbf{a} = 0$. The discriminant of the quadratic equation w.r.t. t is $D(s) = 4\big((\mathbf{p}^T\mathbf{A}\mathbf{r})^2 - s\mathbf{r}^T\mathbf{A}\mathbf{r} \cdot \mathbf{u}^T\mathbf{a}\big)$. The discriminant has a root at $s_0 = (\mathbf{p}^T\mathbf{A}\mathbf{r})^2/(2\mathbf{r}^T\mathbf{A}\mathbf{r} \cdot \mathbf{u}^T\mathbf{a})$. If we set $\sigma = \text{sign}(\mathbf{u}^T\mathbf{a})$, then for each value of $s \geq 0$ it holds that $D(\sigma s) \geq 0$ and hence there are values of t such that t and σs solve the quadratic equation. One solution is again given by (3.11) with τ being defined exactly as in the case of a hyperbolic cylinder. The ray $\mathcal{L}$ is again given by $\mathbf{l}(s) = \mathbf{x}(\sigma s, t(\sigma s))$.

Case 5: $\mathcal{S}$ is a plane defined by the implicit form $\mathbf{n}^T\mathbf{x} = n_0$. Then we choose $\mathbf{r}$ to be any random vector satisfying $\mathbf{n}^T\mathbf{r} = 0$. We define a ray $\mathcal{L}$ in $\mathcal{S}$ from $\mathbf{p}$ to infinity by $\mathbf{l}(t) = \mathbf{p} + t\mathbf{r}$ for $t \geq 0$. The points of intersection between $\mathcal{L}$ and the boundary of $\mathcal{F}$ can be determined as the points of intersection between the boundary of $\mathcal{F}$ and the plane defined by $(\mathbf{n} \times \mathbf{r})^T\mathbf{x} = (\mathbf{n} \times \mathbf{r})^T\mathbf{p}$. This is done analogously to the previous cases.

Curve-Surface-Intersection

Let $\mathcal{C}$ be a conic and $\mathcal{S}$ be a quadric. We show that the points of intersection between $\mathcal{C}$ and $\mathcal{S}$ can be computed by solving only polynomial equations of degree at most four. We assume that $\mathcal{C}$ is given by one of the parameterizations (2.6), with the trigonometric and hyperbolic functions substituted by their rational parameterizations (2.7) and (2.9). Let $\mathbf{c}(t)$ be this parameterization. We observe that this can be written as

$$\mathbf{c}(t) = \frac{1}{p(t)}\tilde{\mathbf{c}}(t),$$

where the components of $\tilde{\mathbf{c}}(t)$ as well as $p(t)$ are at most quadratic polynomials in t. Let $\mathcal{S}$ be given by the quadratic form (2.1). If we insert $\mathbf{c}(t)$ into this form we obtain

$$\tilde{\mathbf{c}}(t)^T\mathbf{A}\tilde{\mathbf{c}}(t) + 2p(t)\tilde{\mathbf{c}}^T\mathbf{a} + p(t)^2a_0 = 0,$$

which is a polynomial equation of degree at most four. The points $\mathbf{c}(t_0)$ for t_0 being a solution of this equation are the points of intersection between $\mathcal{C}$ and $\mathcal{S}$.

Surface-Surface-Intersection

In this paragraph we will show that we can determine at least one point on each loop of the intersection curve between two quadrics by solving polynomial equations of degree at most four. Let $\mathcal{A}$ and $\mathcal{B}$ be two quadrics. We have shown in section 2.3.4 that the intersection curve between two quadrics always lies on an L-quadric. We have also seen that such an L-quadric can be computed by solving polynomial equations of degree at most three. Thus, we can replace one of the two quadrics, say $\mathcal{B}$, by an L-quadric. We have described how to determine a transformation matrix $\mathbf{T}_H$ and a parameterization in homogeneous coordinates $\mathbf{x}(t,\lambda)$ of the transformed L-quadric such that $\mathbf{x}(t,\lambda)$ is linear in λ and the intersection curve is given by

- $\mathbf{c}(t) = \mathbf{T}_H\mathbf{x}(t,\lambda(t))$, where $\lambda(t)$ is obtained by solving a quadratic equation

  $$f(t,\lambda) = \alpha(t)\lambda^2 + \beta(t)\lambda + \gamma(t) = 0$$

  for λ. The polynomials α, β and γ are of degree at most four. In the case $\alpha \not\equiv 0$ the function $\lambda(t)$ is given by equation (2.13). We write $\lambda_+(t)$ if we choose the plus-sign in that equation, and $\lambda_-(t)$ otherwise. We use $\mathbf{c}_+(t)$ and $\mathbf{c}_-(t)$ for the corresponding parts of the intersection curve. We have seen that the degree of the discriminant $D(t)$ is at most four. We can determine the intervals $\mathcal{I}_i$ on which $D(t) \geq 0$ by computing the roots of $D(t)$. We divide each such interval into subintervals $\mathcal{I}_{ij}$ for which $\alpha(t) \neq 0$. Clearly, all points $\mathbf{c}_+(t)$ with $t \in \mathcal{I}_{ij}$ lie on the same loop of the intersection curve, and so do all points $\mathbf{c}_-(t)$. On the other hand, for each point $\mathbf{p}$ on the intersection curve there is a value t such that $\mathbf{p} = \mathbf{c}_+(t)$ or $\mathbf{p} = \mathbf{c}_-(t)$. Hence, we choose a value t_{ij} from each interval $\mathcal{I}_{ij}$ and compute points $\mathbf{p}_{ij+} = \mathbf{c}_+(t_{ij})$ and $\mathbf{p}_{ij-} = \mathbf{c}_-(t_{ij})$. In this way we obtain at least one point from each loop. If $\alpha \equiv 0$, then $\lambda(t)$ is given by equation (2.14). Then we choose a value t from each interval where $\beta(t) \neq 0$ and compute the corresponding point $\mathbf{c}(t)$. In this way we also get at least one point on each loop.

- the union of the up to four straight lines $\mathbf{c}_i(\lambda) = \mathbf{T}_H\mathbf{x}(t_i, \lambda)$, where the t_i are the roots of the polynomial $\gamma(t)$, which has degree at most four. For each t_i we choose a value λ and compute the corresponding point $\mathbf{c}_i(\lambda)$. This gives us one point on each loop of the intersection curve.

Two Evaluation Examples

We implemented the collision test prototypically for a subclass of the quadratic complexes, namely the natural quadratic complexes. In order to test this implementation we constructed two scenes consisting of toy objects. In both scenes we performed a sequence of motions. We stored these sequences to files in order to make them reproducible. Then, we performed two tests for each scene. First, we performed the stored sequence of motions on the curved objects and measured the overall time spent in the collision detection module. Then, we replaced the curved objects by polyhedral approximations and our collision detection module by the package SWIFT++ (see [EL01] for details). We performed the same sequence of motions and again measured the time consumed by the collision test. Figure 3.5 shows snapshots from both scenes. These tests are not to be taken as fair comparisons between our method and SWIFT++. We can always approximate the objects such fine that our method is faster. But nevertheless, the tests show that it is possible to implement our method in such a way that the performance is good compared to algorithms that work on polyhedral objects. The tests also justify that it makes sense to work with the curved objects directly instead of using approximations.

The first scene (the upper images in figure 3.5) consisted of twice the same object. The boundary of this object consisted of 32 faces (eight cylindrical/conical, five spherical, 19 planar), 59 edges (26 circular, 33 straight) and 86 vertices. The approximated object consisted of 11914 triangles, 17871 edges and 5959 vertices. We used a 850MHz machine with 256MB ram for the test. In the curved scene, the overall time spent in the collision detection module was 0.33s, and in the approximated scene it was 28.67s.

The moving object in the second scene (the lower images in figure 3.5) was the boolean union of nine cylindrical objects. The second object was the boolean difference between a cylindrical object and an enlarged version of the first object. The boundary of the first object consisted of 18 faces (16 cylindrical and two planar), 48 edges (32 circular, 16 straight) and 32 vertices. The boundary of the second object had 19 faces (17 cylindrical, two planar), 50 edges (34 circular, 16 straight) and 34 vertices. We approximated the first object with 1372 triangles, 2058 edges and 688 vertices and the second object with 1632 triangles, 2448 edges and 816 vertices. The overall time consumed by the collision detection was 6.58s in the curved scene and 8.34s in the approximated scene.

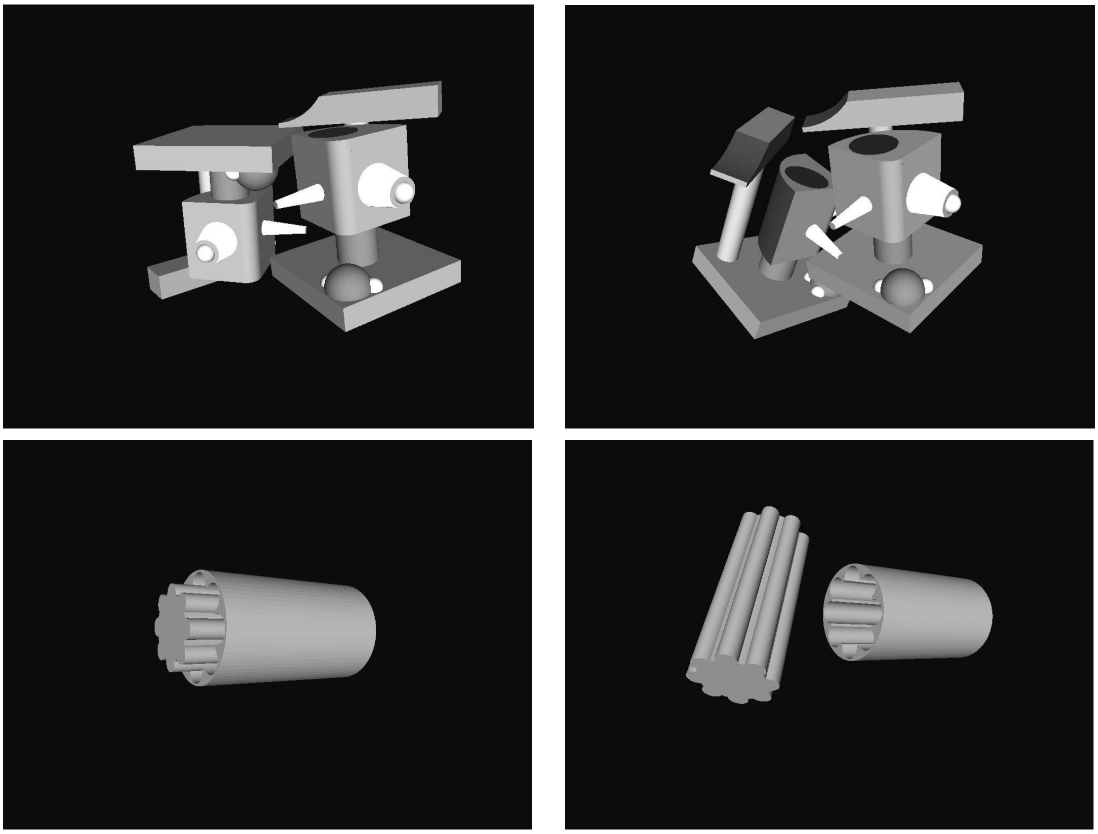


Figure 3.5: Two snapshots from both evaluation scenes.

3.2.3 Specialization for Natural Quadratic Complexes plus Torus

In the following, we will prove a similar result as in the case of QCs.

Theorem 3.6. *The static collision detection problem for the class NQC+T can be reduced to solving polynomial equations of degree at most four.*

We prove this by showing that also for objects from the class NQC+T the four computational tasks of our static collision detection algorithm can be performed by solving only polynomial equations of degree at most four. Since the edges of these objects are straight line segments and circle segments, we already know how to perform the point-on-edge-test.

Point-in-Face-Test

For faces that are embedded on quadrics we have already described that this test can be performed by solving only linear and quadratic equations. So, let the face $\mathcal{F}$ be embedded on a torus $\mathcal{T}$, and let $\mathbf{p}$ be a point on $\mathcal{T}$. If $\mathcal{F}$ has no boundary, then clearly $\mathbf{p}$ lies inside $\mathcal{F}$. Otherwise, we choose a point $\mathbf{q}$ randomly on the boundary of $\mathcal{F}$.

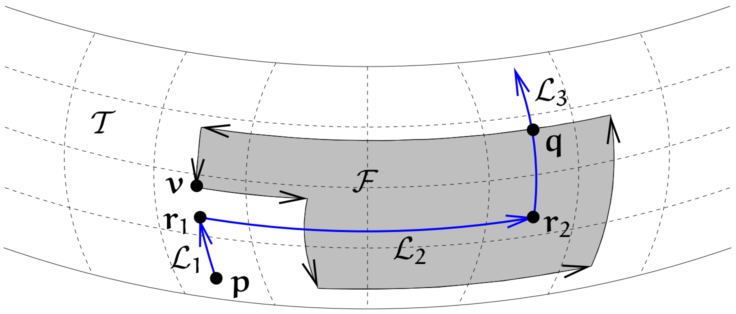


Figure 3.6: Construction of a ray from the query point $\mathbf{p}$ to a boundary point $\mathbf{q}$ of a face $\mathcal{F}$ embedded on a torus $\mathcal{T}$.

We construct a ray $\mathcal{L}$ on $\mathcal{T}$ from $\mathbf{p}$ to $\mathbf{q}$ and use the determinant criterion described in 3.2.1 to decide whether $\mathbf{p}$ lies inside or outside $\mathcal{F}$. This ray will be composed of three parts $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ which are segments of cross-sectional circles and profile circles. If the edge containing $\mathbf{q}$ lies on a profile circle, i.e. the normal of the plane containing that edge is parallel to the normal of the main plane of $\mathcal{T}$, then we proceed as follows (see figure 3.6). We follow the cross-sectional circle $CSC(\mathcal{T},\mathbf{p})$ starting at $\mathbf{p}$. If that circle contains vertices of $\mathcal{F}$, then let $\mathbf{v}$ be the first one that we reach, and let $\mathbf{r}_1$ be a random point on $CSC(\mathcal{T},\mathbf{p})$ between $\mathbf{p}$ and $\mathbf{v}$. Otherwise, let $\mathbf{r}_1$ be any random point on $CSC(\mathcal{T},\mathbf{p})$. The first part $\mathcal{L}_1$ of the ray starts at $\mathbf{p}$, follows $CSC(\mathcal{T},\mathbf{p})$ and ends at $\mathbf{r}_1$. Because of the choice of $\mathbf{r}_1$ we know that $\mathcal{L}_1$ does not hit any vertex. Moreover, because $\mathbf{r}_1$ was randomly chosen we may assume that the profile circle $PFC(\mathcal{T},\mathbf{r}_1)$ also contains no vertices. Since $\mathbf{q}$ was randomly chosen, as well, the same holds for $CSC(\mathcal{T},\mathbf{q})$. Let $\mathbf{r}_2$ be the intersection point of $PFC(\mathcal{T},\mathbf{r}_1)$ and $CSC(\mathcal{T},\mathbf{q})$. The second part $\mathcal{L}_2$ of the ray starts at $\mathbf{r}_1$ and follows $PFC(\mathcal{T},\mathbf{r}_1)$ to the point $\mathbf{r}_2$. Finally, $\mathcal{L}_3$ starts at $\mathbf{r}_2$ and follows $CSC(\mathcal{T},\mathbf{q})$ through $\mathbf{q}$.

If the edge containing $\mathbf{q}$ does not lie on a profile circle, then it must lie on a cross-sectional circle or a Villarceau circle. In that case we proceed similarly: The part $\mathcal{L}_1$ of the ray follows the profile circle through $\mathbf{p}$ to a point that is randomly chosen in such a way that no vertices are hit. Then, $\mathcal{L}_2$ follows the cross-sectional circle through that point until it intersects the profile circle through $\mathbf{q}$. Finally $\mathcal{L}_3$ follows that profile circle until it reaches $\mathbf{q}$.

Let $\mathcal{C}$ be a cross-sectional or profile circle such that a part $\mathcal{L}_i$ of the ray starts at point $\mathbf{a}$ and follows $\mathcal{C}$. Then we parameterize $\mathcal{C}$ in the following way. By section 2.3.2 we know the center $\mathbf{c}_\mathcal{C}$, the normal $\mathbf{n}_\mathcal{C}$ and the radius ρ of $\mathcal{C}$. Since $\mathbf{a} \in \mathcal{C}$ it holds that $|\mathbf{c}_\mathcal{C} - \mathbf{a}| = \rho$ and we define the unit vectors $\mathbf{u} = (\mathbf{c}_\mathcal{C} - \mathbf{a})/\rho$ and $\mathbf{v} = \mathbf{n}_\mathcal{C} \times \mathbf{u}$. As parameterization of $\mathcal{C}$ we choose

$$\mathbf{x}(t) = \mathbf{c}_\mathcal{C} + \rho\frac{1-t^2}{1+t^2}\mathbf{u} + \rho\frac{2t}{1+t^2}\mathbf{v}.$$

With this definition we have $\lim_{t\to\pm\infty}\mathbf{x}(t) = \mathbf{a}$. Thus, finding the first point on $\mathcal{C}$ starting at $\mathbf{a}$ with a certain property is equivalent to finding the smallest value $t_0 \in \mathbb{R}$ such that $\mathbf{x}(t_0)$ has that property. In order to construct the ray as well as to intersect the ray with the boundary of $\mathcal{F}$ we must be able to find the points of intersection between $\mathcal{C}$ and a given circle as well as to check whether a given point lies on $\mathcal{C}$ and if it does, compute the parameter t_0 of that point with respect to the above parameterization. But these tasks work analogously to the point-on-edge-test and the curve-surface-intersection for quadratic complexes that we described earlier in this section.

Curve-Surface-Intersection

We have already described how to find the intersection points between conics and quadrics. Thus, we only have to show that the points of intersection between straight lines and circles and the torus can be computed by solving only quartic equations. For the straight line this is obvious, since we can simply insert its parameterization of the form $\mathbf{x}(t) = \mathbf{a} + t\mathbf{u}$ into the implicit form (2.3) of the torus. This leads to a polynomial equation of degree four in the parameter t. The solutions of this equation are the parameters of the intersection points between the line and the torus.

Now, let $\mathcal{C}$ be a circle with center $\mathbf{c}$, normal $\mathbf{n}$ and radius ρ. By applying a coordinate transformation we can assume that the center of the torus $\mathcal{T}$ lies in the origin and that the normal of its main plane is $[0,0,1]^T$. Let the major and minor radius be given by R and r, respectively. In this situation, the implicit form (2.3) of $\mathcal{T}$ can be rewritten in the form

$$(x_1^2 + x_2^2 + x_3^2 + R^2 - r^2)^2 - 4R^2(x_1^2 + x_2^2) = 0. \tag{3.12}$$

As shown in [Kim98], the points of intersection between $\mathcal{C}$ and $\mathcal{T}$ can be computed as follows. We determine unit vectors $\mathbf{u}$ and $\mathbf{v}$ such that $\mathbf{u} \times \mathbf{v} = \mathbf{n}$. Then we can parameterize the plane containing $\mathcal{C}$ in the form $\mathbf{x}(s,t) = \mathbf{c} + s\mathbf{u} + t\mathbf{v}$. Inserting this into equation (3.12) leads to

$$(s^2 + t^2 + 2s\mathbf{c}^T\mathbf{u} + 2t\mathbf{c}^T\mathbf{v} + R^2 - r^2)^2 - 4R^2\big((c_1 + su_1 + tv_1)^2(c_2 + su_2 + tv_2)^2\big) = 0.$$

In parameters of the plane the circle $\mathcal{C}$ is given by $s^2 + t^2 - \rho^2 = 0$. Thus, we can replace all occurrences of $s^2 + t^2$ in the above equation by ρ^2 and then replace all remaining occurrences of t^2 by $\rho^2 - s^2$. This results in an equation of the form

$$\alpha_4 s^2 + \alpha_3 st + \alpha_2 s + \alpha_1 t + \alpha_0 = 0$$

with constant values $\alpha_0, \ldots, \alpha_4$. Together with the condition $s^2 + t^2 - \rho^2 = 0$ this is a system of equations the solutions of which are the parameters of the intersection points between $\mathcal{C}$ and $\mathcal{T}$. This system can be reduced to

$$\begin{aligned}(\alpha_3^2 + \alpha_4^2)s^4 + 2(\alpha_4\alpha_2 + \alpha_3\alpha_1)s^3 + {}& (2\alpha_0\alpha_4 - \alpha_3^2\rho^2 + \alpha_1^2 + \alpha_2^2)s^2 \\ &+ \ 2(\alpha_0\alpha_2 - \alpha_3\alpha_1\rho^2)s - \rho^2\alpha_1^2 \ = 0 \quad \text{and}\end{aligned}$$

$$t = -\frac{\alpha_4 s^2 + \alpha_2 s + \alpha_0}{\alpha_3 s + \alpha_1}.$$

In this way, we have reduced the problem of finding the points of intersection between a circle and a torus to solving a quartic equation.

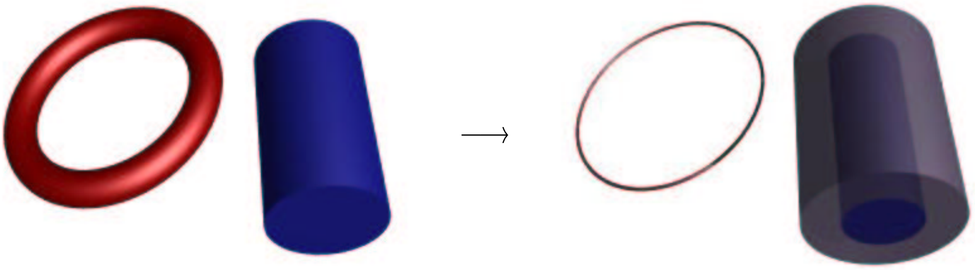


Figure 3.7: The torus is shrunk to a circle and the cylinder is 'blown up' to its C-space obstacle

Surface-Surface-Intersection

We already know that points on the loops of the intersection curve between two quadrics can be computed by solving polynomial equations of degree at most four. We must show that this is also true for the case of two tori and the case of a torus and a natural quadric.

Let the two surfaces to be tested for intersection be $\mathcal{S}_1$ and $\mathcal{S}_2$, and let $\mathcal{S}_1$ be a torus in the position and orientation given by equation (3.12). In the case of $\mathcal{S}_2$ being a sphere we can proceed as follows. If $\mathbf{c}$ is the center of the sphere and ρ is its radius, then its implicit form can be written as $(\mathbf{x} - \mathbf{c})^2 - \rho^2 = 0$. Solving this for $\mathbf{x}^2$ yields $\mathbf{x}^2 = \rho^2 + 2\mathbf{x}^\mathsf{T}\mathbf{c} - \mathbf{c}^2$. If we insert this into (3.12) and define $L = \rho^2 - \mathbf{c}^2 + R^2 - r^2$, then we obtain the quadratic form

$$\begin{aligned} \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + 2\mathbf{x}^\mathsf{T}\mathbf{a} + a_0 &= 0 \quad \text{with} \\ \mathbf{A} &= 4\big(\mathbf{c}\mathbf{c}^\mathsf{T} - \operatorname{diag}(R^2, R^2, 0)\big) \\ \mathbf{a} &= 2L\mathbf{c} \quad \text{and} \\ a_0 &= L^2. \end{aligned}$$

We replace the torus by the quadric defined by this form. Now we are in the surface-surface case for two quadrics.

Now let $\mathcal{S}_2$ be a circular cylinder, a circular cone or a torus. We will briefly describe the approach used in [Kim98] which indeed only requires solving at most quartic equations. For a detailed description of the algorithms we refer the reader to that work. The general idea is the following. If one of the two surfaces under consideration is the envelope surface of a moving ball with radius r, then this surface is replaced by the trajectory of that ball's center. The second surface is replaced by its so-called configuration space obstacle (C-space obstacle) which is bounded by the $\pm r$-offsets of the original surface. Figure 3.7 illustrates this for the case of a torus and a circular cylinder. The idea is now to reduce the surface-surface intersection problem to a curve-surface intersection problem.

We illustrate this in the case of $\mathcal{S}_2$ being a circular cylinder with radius ρ and axis through the point $\mathbf{a}$ with direction $\mathbf{u}$. We consider the case when $r < \rho$. In this case we perform the construction shown in figure 3.7. The torus is considered as the envelope surface of a moving ball and the cylinder as an obstacle. We replace the moving ball by its center and every point on the cylinder by a sphere of radius r. This means that the torus is shrunk to its main circle $\mathcal{C}$ whereas the cylinder is blown up to its C-space obstacle which is bounded by the $\pm r$-offsets of the original cylinder. These offsets are the two coaxial cylinders with radii $\rho + r$ and $\rho - r$, respectively, and with axis through $\mathbf{a}$ with direction $\mathbf{u}$. Figures 3.8 and 3.9 sketch different relative positions of the C-space obstacle and $\mathcal{C}$. The hatched region sketches the interior of the C-space obstacle. Figure 3.8 shows situations where the intersection between $\mathcal{C}$ and the obstacle is not the entire circle, whereas in figure 3.9 it is. The black dots show points of intersection between $\mathcal{C}$ and the boundary of the C-space obstacle. They result from tangential intersection points between the moving ball and $\mathcal{S}_2$. The points on $\mathcal{C}$ that lie in the interior of the obstacle result from positions where the moving ball properly intersects $\mathcal{S}_2$. Let $\mathcal{S}$ be a segment of the intersection between $\mathcal{C}$ and the C-space obstacle of the cylinder $\mathcal{S}_2$. Then in [Kim98] the following is shown.

- If S is not the entire main circle and not a single point then S results from a closed loop of the intersection curve. This is the case for the two segments sketched in figure 3.8(a). If S moreover touches the boundary of the C-space obstacle tangentially in k points then the loop has k singular points. This is sketched in figure 3.8(b).

- If S is a single point then it results from an isolated point of the intersection curve. This is sketched in figure 3.8(c).

- If S is the entire main circle and does not touch the boundary of the C-space obstacle then the intersection curve consists of two closed loops (figure 3.9(a)).

- If S is the entire main circle and touches the boundary of the C-space obstacle in $k > 0$ points then the intersection curve consists of one closed loop with k singular points (figure 3.9(b)).

Computing the intersection of $\mathcal{C}$ with the C-space obstacle means computing the intersections of $\mathcal{C}$ with the two coaxial cylinders. We know that this can be done by solving at most quartic equations. Once we have computed this intersection we must construct one point on each loop of the intersection curve.

- In the cases shown in figure 3.8 and figure 3.9(b) we first choose one of the intersection points between $\mathcal{C}$ and the C-space obstacle for each loop. If for any loop there is a tangential intersection point, then we choose that point. Let $\mathbf{q}$ be such a point. We know that $\mathbf{q}$ results from a situation where the moving ball touches the cylinder surface. We compute this contact point $\mathbf{q}'$. This can be done by dropping a perpendicular from $\mathbf{q}$ to the axis of the cylinder and computing its intersections with the cylinder surface, which involves only polynomials of degree at most two. Then we determine the intersection points between the line through $\mathbf{q}'$ with direction $\mathbf{u}$ and the torus. This requires solving an equation of degree four. As a point on the loop we can choose that intersection point which is closest to $\mathbf{q}'$.

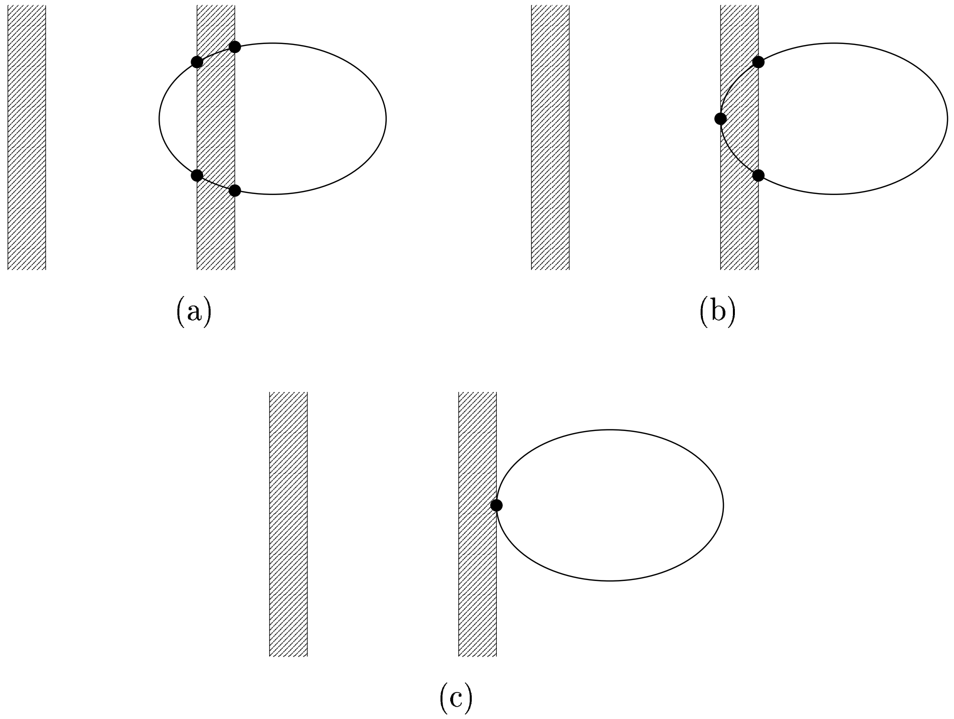


Figure 3.8: Intersection of the main circle of a torus and the C-space obstacle of a cylinder. The intersection is not the entire main circle.

- In the case shown in figure 3.9(a) we may choose any point on the main circle. Let $\mathbf{q}$ be such a point. We compute the two intersection points between the cross-sectional circle centered at $\mathbf{q}$ and the cylinder surface. These points lie on the two loops of the intersection curve. The computations of these intersection points requires solving an equation of degree four.

If the radius ρ of the cylinder is equal to r, then the inner cylinder of the C-space obstacle degenerates to a straight line. In this case one has to be more careful but the general idea still works. If $r > \rho$, then we consider the torus as the obstacle and the cylinder as the envelope surface of a moving ball. We shrink the cylinder to a straight line and replace the torus by its C-space obstacle which is bounded by the $\pm\rho$-offset-surfaces of the torus. Then we proceed similar to the case $r < \rho$.

Since the $\pm r$-offsets of natural quadrics are always composed of natural quadrics, this approach always leads to polynomials of degree at most four.

Remarks on extending the class NQC+T by arbitrary quadrics and conics

We want to discuss briefly how complex the equations become if we do not only consider natural quadrics and linear or circular edges but arbitrary quadrics and conics. The point-on-edge-test and point-in-face-test can still be performed by solving only quartic

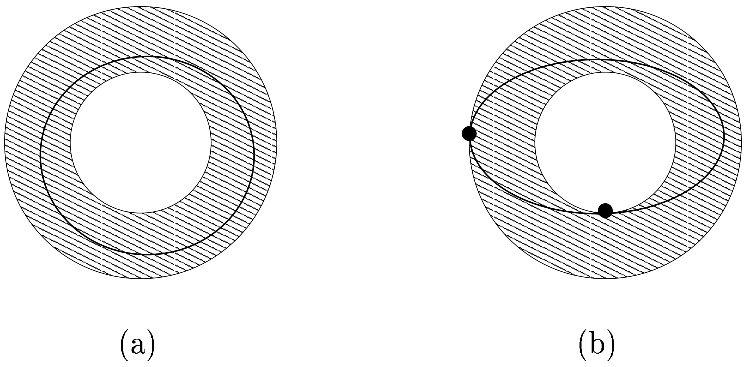


Figure 3.9: Intersection of the main circle of a torus and the C-space obstacle of a cylinder. The intersection is the entire main circle.

equations. But the curve-surface intersection and the surface-surface intersection are more difficult. If we insert the rational parameterization of an arbitrary conic into the implicit form of the torus and multiply with the denominator, then we obtain a polynomial equation of degree eight. Since there can be eight points of intersection between an ellipse and a torus, this degree is optimal.

We have seen that the surface-surface intersection can be reduced to polynomials of degree four in the case that both surfaces are quadrics. For the intersection of a torus and a quadric we want to derive the degrees that occur if we generalize the offset-approach described above. Afterwards, we present a sweep-line approach.

The offset-approach In order to generalize the approach described above, we consider the torus as the envelope surface of a moving ball and the quadric $\mathcal{Q}$ as an obstacle. As before, we shrink the torus to its main circle and replace the quadric to its C-space obstacle which is bounded by the ±r-offset of the quadric. We want to derive an implicit form of this offset surface which can be written as the set

$$\{\, x \pm r n(x) \mid x \in \mathcal{Q} \,\},$$

where $n(x)$ is the normal of $\mathcal{Q}$ in x and $|n(x)| = 1$. We assume that $\mathcal{Q}$ is given in the form (2.1) such that $\mathbf{A} = \operatorname{diag}(a_{11}, a_{22}, a_{33})$, $\mathbf{A}a = 0$ and $a_0 \in \{0, -1\}$. We write $a = [a_1, a_2, a_3]^T$. Let the normal in x be given by $n(x) = (\mathbf{A}x + a)/\lambda$ with

$$\lambda^2 - (\mathbf{A}x + a)^2 = 0 \tag{3.13}$$

We obtain any point y on the offset surface of $\mathcal{Q}$ by

$$y = x + r n(x) = \begin{bmatrix} (1 + a_{11}\mu)x_1 + a_1\mu \\ (1 + a_{22}\mu)x_2 + a_2\mu \\ (1 + a_{33}\mu)x_3 + a_3\mu \end{bmatrix}$$

with $\mu = r/\lambda$ for x fulfilling the quadratic form defining $\mathcal{Q}$. If we solve this for the components of x, then we obtain

$$\begin{aligned} x &= \frac{1}{\delta(\mu)}(\mathbf{D}y - \mu\mathbf{D}a) \quad \text{with} \\ \delta(\mu) &= (1 + a_{11}\mu)(1 + a_{22}\mu)(1 + a_{33}\mu) \quad \text{and} \\ \mathbf{D} &= \operatorname{diag}\big((1 + a_{22}\mu)(1 + a_{33}\mu),\ (1 + a_{11}\mu)(1 + a_{33}\mu),\ (1 + a_{11}\mu)(1 + a_{22}\mu)\big). \end{aligned}$$

We insert this into the quadratic form of $\mathcal{Q}$ and into equation (3.13) and multiply the results with the denominator and obtain the two polynomials

$$\begin{aligned} f_1(y, \mu) &= y^T\mathbf{A}\mathbf{D}^2y + 2\delta(\mu)y^T\mathbf{D}a - 2\mu\delta(\mu)a^T\mathbf{D}a + \delta(\mu)^2 a_0 = 0, & (3.14) \\ f_2(y, \mu) &= r^2\delta(\mu)^2 - a^2\mu^2\delta(\mu)^2 - \mu^2 y^T\mathbf{A}^2\mathbf{D}^2y = 0. & (3.15) \end{aligned}$$

If we compute the resultant of f_1 and f_2 w.r.t. μ, then we obtain an implicit from of the offset surface. This polynomial can be simplified by removing trivial factors. In order to determine the degree of the result, we make the following case distinction. We omit the cases where $\mathcal{Q}$ is a plane or a pair of planes. Let $r(y) = \operatorname{res}_\mu(f_1, f_2)(y)$.

Case 1: $\operatorname{rank}(\mathbf{A}) = 3, a_0 = -1$. Because we assume that $\mathbf{A}a = 0$ we have $a = 0$. We observe that both polynomials f_1 and f_2 have degree six in the variable μ. The coefficients of the powers of μ are polynomials of degree two in y. Thus, the Sylvester matrix w.r.t. μ has dimension 12×12 and its entries are of degree two. Therefore, the degree of $r(y)$ is 24. Furthermore, we see that for $i = 1, 2$ and $j = 1, 2, 3$ we can find polynomials r_{ij} and s_{ij} such that we can write

$$f_i(y, \mu) = y_j^2 r_{ij}(y, \mu) + \left(\mu + \frac{1}{a_{jj}}\right)^2 s_{ij}(y, \mu). \tag{3.16}$$

Then, by applying lemma 2.21 three times we conclude, that there is a degree 12 polynomial h such that $r(y) = (y_1y_2y_3)^4 \cdot h(y)$.

Case 2: $\operatorname{rank}(\mathbf{A}) = 3, a_0 = 0$. In this case, the degree of f_1 in μ is only four. Thus, the degree of r is only 20. As in the previous case we find a decomposition of the form (3.16). Therefore, the non-trivial factor h of the resultant has degree eight in this case.

Case 3: $\operatorname{rank}(\mathbf{A}) = 2, a \neq 0$. W.l.o.g. we assume that $a_{33} = 0$. Because of $\mathbf{A}a = 0$ we have $a = [0, 0, a_3]^T$. Now, the polynomials f_1 and f_2 have degree five and six in μ, respectively. The coefficients of the powers of μ are polynomials of degree at most two in y. Thus, the Sylvester matrix is of dimension 11×11 with entries of degree at most two. If we write

$$f_i(y, \mu) = \sum_{j=0}^{n_i} c_{i,j}(y)\mu^j \tag{3.17}$$

with $n_1 = 5$ and $n_2 = 6$, then we observe that $c_{1,4}$ has degree one and $c_{1,5}, c_{2,5}$ and $c_{2,6}$ have degree zero in $\mathbf{y}$. The transpose of the Sylvester matrix has the form

$$\begin{bmatrix} c_{1,n_1} & 0 & 0 & \cdots & 0 & c_{2,n_2} & 0 & 0 & \cdots & 0 \\ c_{1,n_1-1} & c_{1,n_1} & 0 & \cdots & 0 & c_{2,n_2-1} & c_{2,n_2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

Consequently, only nine columns of the Sylvester matrix have degree two and thus, the degree of the resultant r is only 18. We observe that for $i = 1, 2$ and $j = 1, 2$ we can find a decomposition of the form (3.16), such that we can write $r(\mathbf{y}) = (y_1 y_2)^4 \cdot h(\mathbf{y})$, where h has degree ten in $\mathbf{y}$.

Case 4: $\operatorname{rank}(\mathbf{A}) = 2, \mathbf{a} = \mathbf{0}, a_0 = -1$. In this case, both f_1 and f_2 have degree four in μ and the coefficients are of degree two in $\mathbf{y}$. Thus, the degree of the resultant r is 16. Since we can find a decomposition of the form (3.16), the degree of the non-trivial factor is eight in this case.

Case 5: $\operatorname{rank}(\mathbf{A}) = 1, \mathbf{a} \neq \mathbf{0}$. W.l.o.g. we assume that $a_{22} = a_{33} = 0$. Because of $\mathbf{A}\mathbf{a} = \mathbf{0}$ we have $\mathbf{a} = [0, a_2, a_3]^T$. The degrees of f_1 and f_2 in μ are three and four, respectively. If we write these polynomials in the form (3.17), then we observe that the degrees of the coefficients $c_{1,2}, c_{1,3}, c_{2,3}, c_{2,4}$ in $\mathbf{y}$ are less than two. With the same argument as in case 3 we conclude that the resultant has degree ten in $\mathbf{y}$. For $i = 1, 2$ and $j = 1$ we can again find a decomposition of the form (3.16), such that we can write $r(\mathbf{y}) = y_1^4 \cdot h(\mathbf{y})$, where the degree of h is six.

For special cases the degree of the implicit form of the offset surface can be smaller. For instance, if in case 1 two entries of $\mathbf{A}$ are equal, say $a_{11} = a_{22}$, then we can factor out the term $(1 + a_{11}\mu)^2$ from both equations (3.14) and (3.15). Then, the Sylvester matrix is a 8×8-matrix with entries of degree two in $\mathbf{y}$, such that the degree of the resultant is only 16. Again, we can find decompositions that allow us to apply lemma 2.21 and we find out that the degree of the non-trivial factor h of the resultant is eight. So, the degrees derived above are only valid in the general cases.

The implicit form h that we obtain by the above computation describes both the inner and the outer offset of $\mathcal{Q}$. One could hope to find a factorization $h = h_1 \cdot h_2$ such that h_1 is the implicit form of the inner offset and h_2 of the outer offset. But in [SS00] it is shown that the offset of a quadric is reducible if and only if the quadric is natural. If $\mathcal{Q}$ is a non-natural quadric, then its offset $\mathcal{O}$ is irreducible, which means that the ideal $I(\mathcal{O}) \subset \mathbb{R}[x_1, x_2, x_3]$ of all polynomials vanishing on $\mathcal{O}$ is a prime ideal. This means whenever $f \cdot g \in I(\mathcal{O})$ for two polynomials f and g, then $f \in I(\mathcal{O})$ or $g \in I(\mathcal{O})$. But then, such a factorization of h cannot exist because either h_1 or h_2 would lie in $I(\mathcal{O})$ and would therefore vanish on $\mathcal{O}$.

If we insert the rational parameterization of the main circle of the torus into the implicit form $h(\mathbf{x}) = 0$ of the offset surface of a general quadric, then we obtain a degree 24 polynomial in the worst case. The polynomial h is given by the determinant of a

12×12 matrix divided by a trivial factor. Computing the power series representation of h would be too costly. Therefore, it is more clever to insert the parameterization into the matrix and then use the methods described in 2.4.5 to find the roots of the determinant.

A sweep-line approach Let the quadric be given in general form by the matrix $\mathbf{A}_H$. We assume that the coordinate system has been transformed such that the main plane of the torus is the (x_1, x_2)-plane and its center is the origin. Then we can parameterize the torus rationally by

$$\mathbf{x}(s,t) = \begin{bmatrix} (R + r\frac{1-s^2}{1+s^2})\frac{1-t^2}{1+t^2} \\ (R + r\frac{1-s^2}{1+s^2})\frac{2t}{1+t^2} \\ r\frac{2s}{1+s^2} \end{bmatrix} \tag{3.18}$$

The set of points generated by this parameterization is the torus without two circles. These are the profile circle $\mathcal{C}_1$ centered at the origin with radius $R - r$ and the cross-sectional circle $\mathcal{C}_2$ centered at $[-R, 0, 0]$ with radius r. If we insert this parameterization into the quadratic form defining the quadric and multiply with the denominator, then we obtain an implicit form $f(s,t) = 0$ of the intersection curve between the torus and the quadric in the (s,t)-parameter space. Let $\mathcal{C} \subset \mathbb{R}^2$ be that curve. Since the torus is compact, we know that all loops of the intersection curve are closed. Thus, we have the following correspondence.

- Loops of the intersection curve that do not intersect one of the circles $\mathcal{C}_1$ or $\mathcal{C}_2$ correspond to closed loops of $\mathcal{C}$.

- Loops of the intersection curve that intersect $\mathcal{C}_1$ or $\mathcal{C}_2$ correspond to unbounded loops of $\mathcal{C}$.

- If a loop of the intersection curve is equal to $\mathcal{C}_1$ or $\mathcal{C}_2$, then it does not have a corresponding part in $\mathcal{C}$.

We can compute points on the loops of the second and the third type by intersecting the circles $\mathcal{C}_1$ and $\mathcal{C}_2$ with the quadric. We have seen that this leads to equations of degree at most four.

It remains to compute at least one point on each closed loop of the curve $\mathcal{C}$. This can be done by a sweep-line approach as described in [Gei02]. The idea of this approach is to move a straight line over the parameter space. The line stops whenever it reaches a so-called event point. Figure 3.10 illustrates this approach. The curve in the figure consists of two closed loops. The sweep line is parallel to the t-axis. The event points are the s-extremal points of the curve. Obviously, in this way we find at least one point on each closed loop. The s-extremal points of our curve $\mathcal{C}$ given by the implicit form $f(s,t) = 0$ form a subset of those points in which the normal $\nabla f(s,t)$ is parallel to the s-axis. Thus, the sweep-line has to stop in every position s for which there is a value t

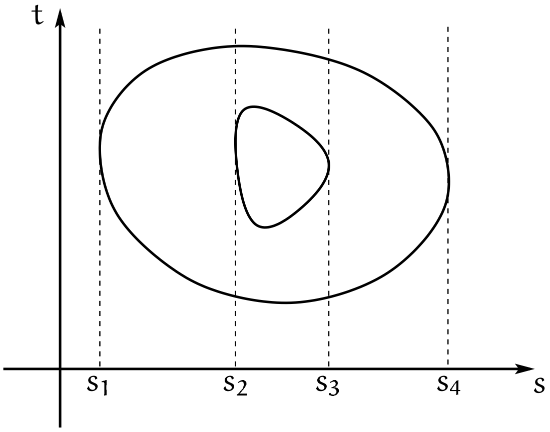


Figure 3.10: A sweep-line approach to find at least one point on each closed loop of a planar curve.

such that

$$\begin{aligned} f(s,t) &= 0 \quad \text{and} \\ \frac{\partial f}{\partial s}(s,t) &= 0. \end{aligned}$$

Computing the resultant of these two polynomials with respect to t leads to a polynomial whose roots are the values of s where the sweep-line has to stop. Figure 3.10 illustrates this approach. The corresponding values of t can then be computed by inserting s into the system of equations and then solving for t. Since the degree of both f and $\partial f/\partial s$ in t is four, the Sylvester matrix with respect to t has dimension eight. But it is also possible to eliminate the variable s first by computing the resultant with respect to s. Since the degree of f in s is four and the degree of $\partial f/\partial s$ in s is three, the Sylvester matrix has dimension seven in that case. Moreover, the only two non-zero entries in the first column of that matrix are the leading coefficients of f and $\partial f/\partial s$, which differ only by a factor of 4. Hence, by applying a row transformation we obtain a matrix with only one non-zero entry in its first column. In this way we can reduce the dimension of the matrix whose determinant has to be computed to six. We can compute this determinant symbolically using Maple®. After dividing by a trivial factor, the resulting degree in t is 16.

3.2.4 Specialization for Quadratic Complexes plus QIC

As in the two previous cases, we state a theorem that bounds the polynomial degrees of the equations occurring in the static collision test.

Theorem 3.7. *The static collision detection problem for the class QC+QIC can be reduced to solving polynomial equations of degree at most eight.*

Again, we show that for objects of the class QC+QIC the four computational tasks of our static collision detection algorithm can be performed by solving polynomial equations of degree at most eight. We have already discussed the point-in-face- test for quadric patches as well as the surface-surface-intersection for quadrics. Thus, we can concentrate on the point-on-edge-test and the curve-surface-intersection.

Point-on-Edge-Test

We know that this task is trivial provided that we know the curve parameter of the query point. Thus, we have to show how to compute this parameter t_0 for a point $\mathbf{p}$ lying on the curve $\mathcal{C}$. We already have described how this can be done if $\mathcal{C}$ is a conic. So we assume that $\mathcal{C}$ is a quadric intersection curve. As we have stated in section 2.3.4, we assume that we know a homogeneous matrix $\mathbf{T}_H$, a parameterization $\mathbf{x}(t, \lambda)$ in homogeneous coordinates and a function $\lambda(t)$ such that the parameterization of $\mathcal{C}$ is given by $\mathbf{c}(t) = \mathbf{T}_H\mathbf{x}(t, \lambda(t))$. If we compute the point

$$\mathbf{p}' = \mathbf{T}_H^{-1} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix},$$

then we have to compute t such that $\mathbf{x}(t, \lambda(t)) = \mathbf{p}'$. Since $\mathbf{T}_H$ is a product of rotation, translation and scaling matrices, we can be sure that $p_4' \neq 0$. Thus, we can assume that $\mathbf{p}'$ is homogenized, i.e. $p_4' = 1$. We have seen in 2.3.4 that

$$\mathbf{x}(t, \lambda) \in \left\{ \begin{bmatrix} 0 \\ t \\ \lambda \\ 1 \end{bmatrix}, \begin{bmatrix} \pm 1 \\ \lambda \\ t \\ 1 \end{bmatrix}, \begin{bmatrix} t \\ \lambda \\ -t^2 \\ 1 \end{bmatrix}, \begin{bmatrix} t \\ \lambda \\ t\lambda \\ 1 \end{bmatrix}, \begin{bmatrix} t \\ 0 \\ \lambda \\ 1 \end{bmatrix}, \begin{bmatrix} \pm(1+t^2) \\ 2t \\ \lambda \\ 1-t^2 \end{bmatrix} \right\}.$$

In the cases where the fourth component of $\mathbf{x}(t, \lambda)$ is one, we can determine t_0 and λ_0 from $\mathbf{p}'$ by just looking at the right component. Then we check whether $\lambda_0 = \lambda(t_0)$. If this is the case, then t_0 is the parameter that we wanted to compute. Otherwise the connected component of $\mathcal{C}$ containing $\mathbf{p}$ is different from the one containing the edge and thus, $\mathbf{p}$ does not lie on the edge. In the case where the fourth component of $\mathbf{x}(t, \lambda)$ is not constantly one, we use the second components of $\mathbf{p}'$ and $\mathbf{x}(t, \lambda)$ to compute t:

$$p_2' = \frac{2t}{1-t^2} \quad \Leftrightarrow \quad t = \begin{cases} \frac{-1 \pm \sqrt{1+p_2'^2}}{p_2'} & \text{if } p_2' \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

By the third components we have $\lambda_0 = (1 - t_0^2)p_3'$. Again, we check whether $\lambda_0 = \lambda(t_0)$.

Curve-Surface-Intersection

We have already seen that the curve-surface-intersection problem for a conic and a quadric leads to a polynomial equation of degree four. So let $\mathcal{C}$ be a QIC and let $\mathcal{Q}$ be a quadric. In order to compute the points of intersection between $\mathcal{C}$ and $\mathcal{Q}$ we could insert the parameterization of $\mathcal{C}$ into the quadratic form of $\mathcal{Q}$. If we look at the parameterization that we derived in section 2.3.4, we see that this can lead to a polynomial of degree 12 in the worst case. The following example shows that this degree can actually be reached.

Example 3.8. *Let $\mathcal{C}$ be a part of the intersection curve between the hyperbolic paraboloid parameterized by $\mathbf{x}(t, \lambda) = [t, \lambda, t\lambda, 1]^T$ and the ellipsoid defined by the homogeneous matrix*

$$\mathbf{A}_H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

By inserting $\mathbf{x}(t, \lambda)$ into the quadratic form defining the ellipsoid we find that the intersection curve can be parameterized by $\mathbf{c}(t) = [t, \lambda(t), t\lambda(t), 1]^T$ with

$$\lambda(t) = \frac{-t \pm \sqrt{2 - t^4}}{2 + t^2}.$$

Let $\mathcal{C}$ be the part of the curve that is parameterized if we take the plus sign in this equation. Now let $\mathcal{Q}$ be the sphere defined by the homogeneous matrix $\mathbf{B}_H = \mathrm{diag}(1, 1, 1, -r^2)$. If we insert $\mathbf{c}(t)$ into the quadratic form defining $\mathcal{Q}$ and clear the denominators, then we obtain the condition

$$(t^2 - r^2)(t^2 + 2) + (t^2 + 1)(t^2 - t^4 + 2) = 2t(t^2 + 1)\sqrt{2 - t^4}.$$

In order to obtain a polynomial we square this equation. The degree of the resulting polynomial is 12.

The intersection points between $\mathcal{C}$ and $\mathcal{Q}$ are a subset of the intersection points between three quadrics, namely between $\mathcal{Q}$ and the two quadrics defining $\mathcal{C}$. But these points are the common roots of the three quadratic forms defining these quadrics. Bezout's theorem tells us that there are only eight such common roots. So the question is whether this problem can be reduced to a polynomial equation of degree eight. In [CGM91] it is shown by means of multivariate resultants that this is indeed the case. They use a result from [Mac02] that expresses the resultant of k homogeneous polynomials in k variables as the quotient of two determinants $\det \mathbf{N} / \det \mathbf{D}$, where the denominator is a factor of the numerator and $\mathbf{D}$ is a submatrix of $\mathbf{N}$. In order to apply this to three arbitrary polynomials f_1, f_2 and f_3 in three variables x, y and z one considers the f_i as polynomials in only two variables x and y with coefficients being polynomials in z. Then one introduces a new variable, say w, to homogenize the f_i. Then, the resultant $\det \mathbf{N} / \det \mathbf{D}$ is a polynomial of degree eight in z the roots of which are the z-coordinates of the common roots of the f_i. In [CGM91] it is shown how to make this resultant expression division-free by means of appropriate matrix transformations. For each root z_0 of this polynomial the corresponding x and y coordinates are computed by substituting z_0 for z in the f_i and then solving three equations of degree two in two variables. This can be done using the Sylvester resultant which leads to equations of degree at most four.

In [ES03] a method is presented to express this multivariate resultant directly as the determinant of a matrix rather than the quotient of two determinants. Let for $i = 1, 2, 3$ the homogeneous polynomials $f_i(x, y, z)$ be given in the form

$$f_i(x, y, z) = \alpha_{i0}x^2 + \alpha_{i1}xy + \alpha_{i2}xz + \alpha_{i3}y^2 + \alpha_{i4}yz + \alpha_{i5}z^2.$$

These are the implicit forms of three conics in the projective plane. By the above mentioned publication, these curves have a common point iff the determinant of the matrix $\mathbf{M}(f_1, f_2, f_3)$ is zero. This matrix is given by

$$\begin{bmatrix} \alpha_{10} & \alpha_{20} & \alpha_{30} & [0,1,5] & 0 & [0,1,2] \\ \alpha_{11} & \alpha_{21} & \alpha_{31} & [0,3,5] & [0,3,4] & [0,1,4]-[0,2,3] \\ \alpha_{12} & \alpha_{22} & \alpha_{32} & [0,4,5]-[1,2,5] & [0,3,5] & [0,1,5] \\ \alpha_{13} & \alpha_{23} & \alpha_{33} & 0 & [1,3,4] & [0,3,4] \\ \alpha_{14} & \alpha_{24} & \alpha_{34} & [2,3,5] & [2,3,4]+[1,3,5] & [0,3,5] \\ \alpha_{15} & \alpha_{25} & \alpha_{35} & [2,4,5] & [2,3,5] & 0 \end{bmatrix},$$

where $[i, j, k]$ denotes the Plücker coordinate

$$[i, j, k] = \det \begin{bmatrix} \alpha_{1i} & \alpha_{2i} & \alpha_{3i} \\ \alpha_{1j} & \alpha_{2j} & \alpha_{3j} \\ \alpha_{1k} & \alpha_{2k} & \alpha_{3k} \end{bmatrix}.$$

Now let $g_i(x, y, z) = 0$ for $i = 1, 2, 3$ be the three quadrics to be intersected. We write these polynomials in the form $h_i(x, y) = 0$ with $h_i \in \mathbb{R}[z][x, y]$, i.e. the coefficients of h are polynomials in z. We introduce a new variable w to homogenize the polynomials h_i and obtain H_i. Then, the determinant of $\mathbf{M}(H_1, H_2, H_3)$ is a polynomial in z whose roots are the z-coordinates of the intersection points between the three quadrics. The degree matrix of $\mathbf{M}(H_1, H_2, H_3)$ is

$$\begin{bmatrix} 0 & 0 & 0 & 2 & -\infty & 1 \\ 0 & 0 & 0 & 2 & 1 & 1 \\ 1 & 1 & 1 & 3 & 2 & 2 \\ 0 & 0 & 0 & -\infty & 1 & 1 \\ 1 & 1 & 1 & 3 & 2 & 2 \\ 2 & 2 & 0 & 4 & 3 & -\infty \end{bmatrix}.$$

With the method described in 2.4.2 we can determine an upper bound for the degree of $\mathbf{M}(H_1, H_2, H_3)$ using this degree matrix. We find that this bound is indeed 8.

We now describe two alternative methods to find the intersection points between three quadrics. The first one uses a Groebner basis approach to derive the degree eight polynomial in z directly, i.e. without generating a quotient of determinants that has to be made division-free. The second approach uses the parameterization of the L-quadric in the pencil of two of the quadrics in order to derive a polynomial of degree eight in the curve parameter of the QIC.

A Groebner basis approach Let $f_i(x, y, z) = 0$ for $i = 1, 2, 3$ be the three quadratic forms to be solved. This system of equations can be written in the form

$$[\mathbf{M}, \mathbf{N}] \cdot [x^2, xy, y^2, x, y, 1]^T = 0$$

with $\mathbf{M} \in \mathbb{R}^{3\times3}$ and $\mathbf{N} \in \mathbb{R}[z]^{3\times3}$. Multiplying this equation from the left with a regular 3×3-matrix preserves the set of solutions. We know from linear algebra that each matrix $\mathbf{A}$ can be transformed into reduced row echelon form by a sequence of invertible row transformations, and that this form is uniquely determined by $\mathbf{A}$. Each invertible row transformation can be expressed as a multiplication from the left with a regular matrix. Thus, there exists a regular matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ such that $\mathbf{RM}$ is in reduced row echelon form. Therefore we may assume that $\mathbf{M}$ is already in that form. We see that $\mathbf{M}$ must have one of the following eight forms.

$$\mathrm{rank}\mathbf{M} = 3 : \quad \mathbf{E}$$

$$\mathrm{rank}\mathbf{M} = 2 : \quad \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \beta \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & \alpha & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{rank}\mathbf{M} = 1 : \quad \begin{bmatrix} 1 & \alpha & \beta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 1 & \alpha \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{rank}\mathbf{M} = 0 : \quad \mathbf{0}$$

with $\alpha, \beta \in \mathbb{R}$. Let the entries of the matrix $\mathbf{N}$ be denoted as a_{ij}. Now we treat α and β as well as the a_{ij} as indetermined coefficients and compute for each of the eight cases a Groebner basis $\mathcal{G} = [g_1, \ldots, g_k]$ using lex order with $x \prec y \prec a_{ij}$ for all i, j and some arbitrary order among the a_{ij}. Fortunately, the fact that $\mathbf{M}$ is in this special form has the effect that $\mathcal{G}$ can be computed within few minutes using some computer algebra system such as Maple®. The polynomial g_k does not contain the variables x and y and by substituting the symbols a_{ij} by the polynomials in z we obtain the desired polynomial of degree eight. The Groebner basis $\mathcal{G}$ also contains polynomials that are linear in x and y. They can be used to determine the corresponding values for x and y for each root of g_k.

An L-quadric approach Let the two quadrics containing the QIC $\mathcal{C}$ be denoted as $\mathcal{A}$ and $\mathcal{B}$. We know a parameterization $\mathbf{x}_H(t, \lambda)$, a matrix $\mathbf{T}_H$ and a function $\lambda(t)$ such that $\mathcal{C}$ is parameterized by $\mathbf{c}_H(t) = \mathbf{T}_H\mathbf{x}_H(t, \lambda(t))$. We saw in section 2.3.4 that $\mathbf{T}_H\mathbf{x}_H(t, \lambda)$ parameterizes an L-quadric $\mathcal{L}$ in the pencil of $\mathcal{A}$ and $\mathcal{B}$, and that we can write $\mathbf{x}_H(t, \lambda) = \mathbf{p}_H(t) + \lambda\mathbf{r}_H(t)$ such that the degree condition 2.12 is fulfilled. W.l.o.g. we assume that $\mathcal{L} \neq \mathcal{A}$. Let the quadrics $\mathcal{A}$ and $\mathcal{Q}$ be given by the homogeneous matrices $\mathbf{A}_H$ and $\mathbf{Q}_H$, respectively. The intersection between $\mathcal{L}$ and $\mathcal{A}$ and between $\mathcal{L}$ and $\mathcal{Q}$ in

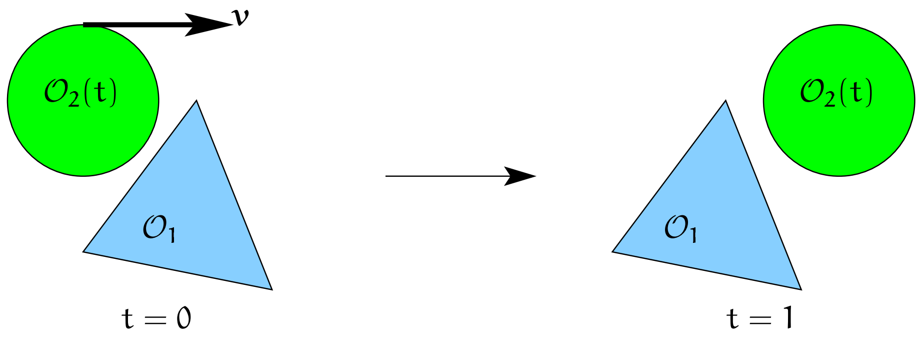


Figure 3.11: The objects do not intersect at $t = 0$ and $t = 1$. Thus, the static collision detection does not report a collision.

the parameters t and λ are given by the solutions of

$$\begin{aligned} f_1(t,\lambda) &= \mathbf{x}_H(t,\lambda)^T\mathbf{T}_H^T\mathbf{A}_H\mathbf{T}_H\mathbf{x}_H(t,\lambda) = 0 \quad \text{and} \\ f_2(t,\lambda) &= \mathbf{x}_H(t,\lambda)^T\mathbf{T}_H^T\mathbf{Q}_H\mathbf{T}_H\mathbf{x}_H(t,\lambda) = 0. \end{aligned}$$

We define the polynomial $g(t) = \text{res}_\lambda(f_1, f_2)(t)$. If for a real solution t_0 of $g(t) = 0$ and for $\lambda_0 = \lambda(t_0)$ the condition $f_1(t_0,\lambda_0) = f_2(t_0,\lambda_0) = 0$ is fulfilled, then the point $\mathbf{c}_H(t_0)$ is an intersection point between $\mathcal{C}$ and $\mathcal{Q}$. In order to see that the degree of $g(t)$ is at most eight we define $\tilde{\mathbf{A}}_H = \mathbf{T}_H^T\mathbf{A}_H\mathbf{T}_H$ and $\tilde{\mathbf{Q}}_H = \mathbf{T}_H^T\mathbf{Q}_H\mathbf{T}_H$ and write down the Sylvester matrix of f_1 and f_2 w.r.t. λ.

$$\mathbf{S} = \begin{bmatrix} \mathbf{r}^T\tilde{\mathbf{A}}\mathbf{r} & 2\mathbf{p}^T\tilde{\mathbf{A}}\mathbf{r} & \mathbf{p}^T\tilde{\mathbf{A}}\mathbf{p} & 0 \\ 0 & \mathbf{r}^T\tilde{\mathbf{A}}\mathbf{r} & 2\mathbf{p}^T\tilde{\mathbf{A}}\mathbf{r} & \mathbf{p}^T\tilde{\mathbf{A}}\mathbf{p} \\ \mathbf{r}^T\tilde{\mathbf{Q}}\mathbf{r} & 2\mathbf{p}^T\tilde{\mathbf{Q}}\mathbf{r} & \mathbf{p}^T\tilde{\mathbf{Q}}\mathbf{p} & 0 \\ 0 & \mathbf{r}^T\tilde{\mathbf{Q}}\mathbf{r} & 2\mathbf{p}^T\tilde{\mathbf{Q}}\mathbf{r} & \mathbf{p}^T\tilde{\mathbf{Q}}\mathbf{p} \end{bmatrix}.$$

We omitted the parameter t as well as the subscript H for the sake of readability. If we expand the determinant of $\mathbf{S}$, e.g. by cofactors of the first column, then we see that each addend has degree four in $\mathbf{p}$ as well as in $\mathbf{r}$. Thus, the degree of g in t is $4(\deg \mathbf{p} + \deg \mathbf{r}) \leq 8$ by condition 2.12. If the degree of f_1 and/or f_2 in λ is less than two, then the degree of g is obviously even smaller.

3.3 Dynamic Collision Test

The biggest problem of a static collision detection algorithm is the fact that one can easily miss collisions due to the discretization of the time. Figure 3.11 illustrates this problem. At time $t = 0$, the objects do not intersect. Object $\mathcal{O}_2$ moves with velocity $\boldsymbol{v}$. At time $t = 1$, the situation is again intersection-free. Thus, the static collision detection does not report a collision although this motion cannot be performed without intersection. The dynamic collision detection is a possibility to avoid this problem.

In this section we describe a dynamic collision test for quadratic complexes and natural quadratic complexes. To make things easier, we assume that the object $\mathcal{O}_1$ is stationary and $\mathcal{O}_2(t)$ is moving during the time interval $[0,1]$. We also assume that at time $t = 0$ we are in a valid situation, i.e. $\text{int}(\mathcal{O}_1) \cap \text{int}(\mathcal{O}_2(0)) = \emptyset$, where $\text{int}(\mathcal{O})$ denotes the interior of the set $\mathcal{O}$. The problem that we want to solve is the following. First, we want to decide whether there are values $t \in (0,1)$ such that $\text{int}(\mathcal{O}_1) \cap \text{int}(\mathcal{O}_2(t)) \neq \emptyset$. If this is the case, we want to compute the point in time $t_{\text{col}} \in [0,1)$ such that $\text{int}(\mathcal{O}_1) \cap \text{int}(\mathcal{O}_2(t)) = \emptyset$ for all $t \in [0, t_{\text{col}}]$ and there is an $\varepsilon > 0$ such that $\text{int}(\mathcal{O}_1) \cap \text{int}(\mathcal{O}_2(t)) \neq \emptyset$ for all $t \in (t_{\text{col}}, t_{\text{col}} + \varepsilon)$. In addition to the collision time t_{col} we want to compute a point $\mathbf{p}_{\text{col}} \in \partial\mathcal{O}_1$ that witnesses the penetration. We define what this means.

Definition 3.9. *Let $\mathbf{p}$ be a point on the boundary of $\mathcal{O}_1$. We say that $\mathbf{p}$ witnesses a penetration at time t_c, if there is an $\varepsilon > 0$ such that for each neighbourhood $\mathcal{N}$ of $\mathbf{p}$ the intersection $\mathcal{N} \cap \text{int}(\mathcal{O}_1) \cap \text{int}(\mathcal{O}_2(t)) \neq \emptyset$ for all $t \in (t_c, t_c + \varepsilon)$.*

Now, in addition to the collision time t_{col} we compute a point $\mathbf{p}_{\text{col}}$ that witnesses a penetration at time t_{col}.

We assume that the linear velocity $\boldsymbol{\nu}$ and the angular velocity $\boldsymbol{\omega}$ of $\mathcal{O}_2$ are constant over the time interval. Together with the assumption that only one object moves whereas the other one is fixed this makes it necessary to sequentialize the motions of the objects. The reason for this is that if two objects $\mathcal{O}_1$ and $\mathcal{O}_2$ rotate simultaneously, then the angular velocity of $\mathcal{O}_2$ seen from the local coordinate system of $\mathcal{O}_1$ is not constant in general, even if both angular velocities are constant in global coordinates. We propose two ways two sequentialize the motions. The first is to fix an ordering on the objects and then move object 1 first whereas the others are fixed, then move object 2, etc. At each step, the moving object performs a superposition of a translational and a rotational motion. We will see later that the degrees of the polynomial equations that have to be solved are much less for pure translations or pure rotations. Therefore, we propose a second way of sequentializing. First, we perform the translations of all objects at once. For the collision detection this means that it considers all pairs of objects $(\mathcal{O}_i, \mathcal{O}_j)$ and looks at the motion from the local coordinate system of $\mathcal{O}_i$. Next, we proceed as in the first approach and rotate the objects one at a time whereas the others are fixed.

A naive way to perform a dynamic collision test between two objects would be to use the velocities $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ and the diameter d of $\mathcal{O}_2$ and the distance δ between $\mathcal{O}_1$ and $\mathcal{O}_2(0)$ to compute a lower bound τ for the duration of the collision-free motion as shown in [Mir96b]. If $\tau \geq 1$, then we know that there is no collision in the interval $[0,1]$. Otherwise, we consider the interval $[\tau, 1]$ and compute a new lower bound, etc. We stop, if the difference of two successive values of τ is smaller than a given bound. But this would be an extremely time consuming approach, since the computation of the distance between two quadratic complexes is anything but a trivial task and requires finding the roots of polynomials of high degrees (see [Len04] for details).

In the following we will present a different approach to solve the dynamic collision detection problem. We will start with giving a generic algorithm and then continue with concrete descriptions of the parts of the algorithm.

3.3.1 A Generic Algorithm

Depending on the parts of the boundaries of the objects that contain the witness point $\mathbf{p}_{\text{col}}$ we can define four types of collision:

vertex-face: $\mathbf{p}_{\text{col}}$ is a vertex of $\mathcal{O}_1$ or $\mathcal{O}_2(\mathrm{t}_{\text{col}})$.

edge-edge: $\mathbf{p}_{\text{col}}$ lies both on an edge of $\mathcal{O}_1$ and an edge of $\mathcal{O}_2(\mathrm{t}_{\text{col}})$ and is not a vertex.

edge-face: $\mathbf{p}_{\text{col}}$ lies either on an edge of $\mathcal{O}_1$ or an edge of $\mathcal{O}_2(\mathrm{t}_{\text{col}})$ but not both.

face-face: Otherwise.

Definition 3.10.

- *If $\mathbf{a}$ is a vertex of $\mathcal{O}_1$ and $\mathcal{F}$ is a face of $\mathcal{O}_2$ embedded in the surface $\mathcal{S}$, then $\mathrm{t}_0 \in [0,1)$ is a potential collision time between $\mathbf{a}$ and $\mathcal{F}$ if $\mathbf{a} \in \mathcal{S}(\mathrm{t}_0)$. If additionally $\mathbf{a} \in \mathcal{F}(\mathrm{t}_0)$ and $\mathbf{a}$ witnesses a penetration at time t_0 we say that $\mathbf{a}$ collides with $\mathcal{F}$ at time t_0. The definition for the case that $\mathbf{a}$ is a vertex of $\mathcal{O}_2$ and $\mathcal{F}$ is a face of $\mathcal{O}_1$ is similar.*

- *If $\mathcal{E}_1$ and $\mathcal{E}_2$ are edges on $\mathcal{O}_1$ and $\mathcal{O}_2$ lying on the curves $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, then $\mathrm{t}_0 \in [0,1)$ is a potential collision time between $\mathcal{E}_1$ and $\mathcal{E}_2$ if $\mathcal{C}_1$ and $\mathcal{C}_2(\mathrm{t}_0)$ intersect in a point $\mathbf{p}$. If additionally $\mathbf{p}$ lies on both $\mathcal{E}_1$ and $\mathcal{E}_2(\mathrm{t}_0)$ and $\mathbf{p}$ witnesses a penetration at time t_0 we say that $\mathcal{E}_1$ collides with $\mathcal{E}_2$ in the point $\mathbf{p}$ at time t_0.*

- *If $\mathcal{E}$ is an edge of $\mathcal{O}_1$ and $\mathcal{F}$ is a face of $\mathcal{O}_2$, and $\mathcal{E}$ is embedded on the curve $\mathcal{C}$ and $\mathcal{F}$ is embedded on the surface $\mathcal{S}$, then $\mathrm{t}_0 \in [0,1)$ is a potential collision time between $\mathcal{E}$ and $\mathcal{F}$ if $\mathcal{C}$ and $\mathcal{S}(\mathrm{t}_0)$ have a tangential intersection point $\mathbf{p}$. If additionally $\mathbf{p}$ lies on both $\mathcal{E}$ and $\mathcal{F}(\mathrm{t}_0)$ and $\mathbf{p}$ witnesses a collision at time t_0 we say that $\mathcal{E}$ collides with $\mathcal{F}$ in the point $\mathbf{p}$ at time t_0. The definition for the case that $\mathcal{E}$ belongs to $\mathcal{O}_2$ and $\mathcal{F}$ is a face of $\mathcal{O}_1$ is similar.*

- *If $\mathcal{F}_1$ and $\mathcal{F}_2$ are faces on $\mathcal{O}_1$ and $\mathcal{O}_2$ lying on the surfaces $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively, then we say that $\mathrm{t}_0 \in [0,1)$ is a potential collision time between $\mathcal{F}_1$ and $\mathcal{F}_2$ if $\mathcal{S}_1$ and $\mathcal{S}_2(\mathrm{t}_0)$ have a tangential intersection point $\mathbf{p}$. If additionally $\mathbf{p}$ lies on both $\mathcal{F}_1$ and $\mathcal{F}_2(\mathrm{t}_0)$ and $\mathbf{p}$ witnesses a penetration at time t_0 we say that $\mathcal{F}_1$ collides with $\mathcal{F}_2$ in the point $\mathbf{p}$ at time t_0.*

With these definitions we can formulate a generic algorithm for the collision detection between two objects. For each pair $(\mathcal{F}_1, \mathcal{F}_2)$ of $\mathcal{O}_1$ and $\mathcal{O}_2$ we perform the following four types of tests according to the above described collision types.

1. Vertex-Face Test: For each vertex of $\mathcal{F}_1$ we check whether it collides with $\mathcal{F}_2$ and vice versa. We set t_{col} to the minimum of all the respective collision times.

2. Edge-Edge Test: For each edge of $\mathcal{F}_1$ we check whether it collides with an edge of $\mathcal{F}_2$. We set t_{col} to the minimum of the earliest among the respective collision times and the previously computed minimum.

3. Edge-Face Test: For each edge of $\mathcal{F}_1$ we check whether it collides with $\mathcal{F}_2$ and vice versa. Again, we set t_{col} to the minimum of its previous value and the earliest such collision time.

4. Face-Face Test: Finally, we check whether $\mathcal{F}_1$ collides with $\mathcal{F}_2$. As before, we set t_{col} appropriately.

Whenever a new collision time is computed, a corresponding point $\mathbf{p}_{col}$ is determined, as well. If none of the above tests detects a collision, then $\mathcal{O}_1$ and $\mathcal{O}_2$ do not collide. Instead of considering all pairs of faces, we use the techniques described in section 3.1 to exclude parts of the objects from the test. According to definition 3.10 we can split each of the four tests into three parts. If $\mathcal{G}_1$ and $\mathcal{G}_2$ are the features of the objects $\mathcal{O}_1$ and $\mathcal{O}_2$ to be checked for collision, then first compute the potential collision times between $\mathcal{G}_1$ and $\mathcal{G}_2$ and the corresponding intersection points. Then check for each such time t_0 and corresponding point $\mathbf{p}_0$ whether $\mathbf{p}_0$ lies on both features at time t_0. This second test means performing point-on-edge tests and/or point-in-face tests. Finally, check whether $\mathbf{p}_0$ witnesses a collision at time t_0. We call this the *penetration test.*

In the following, we examine how these tests can be performed in the case of quadratic complexes and natural quadratic complexes. We assume that object $\mathcal{O}_1$ is fixed in time and that $\mathcal{O}_2$ moves with linear velocity $\boldsymbol{v}$ and angular velocity $\boldsymbol{\omega}$. Thus, for every point $\mathbf{p}(t) \in \mathcal{O}_2(t)$ we have

$$\mathbf{p}(t) = \mathbf{R}(t)(\tilde{\mathbf{p}} - \tilde{\mathbf{c}}) + \tilde{\mathbf{c}} + \mathbf{s}(t),$$

where $\tilde{\mathbf{c}}$ is the center of mass of $\mathcal{O}_2(0)$. The rotation matrix is given by Rodrigues' formula

$$\mathbf{R}(t) = \cos(|\boldsymbol{\omega}|t) \cdot \mathbf{E} + \big(1 - \cos(|\boldsymbol{\omega}|t)\big) \cdot \boldsymbol{w}\boldsymbol{w}^T + \sin(|\boldsymbol{\omega}|t) \cdot \boldsymbol{w}^\times,$$

with $\boldsymbol{w} = \boldsymbol{\omega}/|\boldsymbol{\omega}|$ if $\boldsymbol{\omega} \neq 0$ and $\boldsymbol{w} = 0$ otherwise. The translation vector is defined as $\mathbf{s}(t) = \boldsymbol{v} \cdot t$.

In order to obtain polynomial equations, we use a similar substitution as in (2.7) to transform the trigonometric functions in rational ones. We substitute $|\boldsymbol{\omega}|t = 2\arctan(wt)$ and obtain

$$\cos(|\boldsymbol{\omega}|t) = \frac{1-(wt)^2}{1+(wt)^2}, \quad \text{and} \quad \sin(|\boldsymbol{\omega}|t) = \frac{2wt}{1+(wt)^2}.$$

This substitution is equivalent to

$$|\boldsymbol{\omega}| = \frac{2}{t}\arctan(wt) \quad \Leftrightarrow \quad w = \frac{1}{t}\tan\left(\frac{|\boldsymbol{\omega}|t}{2}\right) \approx \frac{|\boldsymbol{\omega}|}{2} \quad \text{for small values of } |\boldsymbol{\omega}|t.$$

Thus, we set $w = |\boldsymbol{\omega}|/2$ and assume that w is constant over the time interval. As we always normalize this interval to $[0,1]$, it can easily be shown that the relative deviation of the replacement of $|\boldsymbol{\omega}|$ from the original value is bounded by

$$1 - \frac{2\arctan(|\boldsymbol{\omega}|/2)}{|\boldsymbol{\omega}|}.$$

If we want this deviation to be less than one percent, $|\boldsymbol{\omega}|$ must be less than $0.35 s^{-1}$. In a real-time simulation with twenty frames per second this corresponds to approximately 400 degrees per second[1].

If a quadric associated with $\mathcal{O}_2(0)$ is defined by the quadratic form

$$S_0(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{x}^T\mathbf{a} + a_0 = 0,$$

then with the above definitions the quadratic form of the quadric at time t can be derived by considering the equivalence $S_t(\mathbf{p}(t)) = 0 \Leftrightarrow S_0(\tilde{\mathbf{p}}) = 0$ which leads to

$$\begin{aligned}
S_t(\mathbf{x}) &= \mathbf{x}^T\mathbf{A}(t)\mathbf{x} + 2\mathbf{x}^T\mathbf{a}(t) + a_0(t) \quad \text{with} \\
\mathbf{A}(t) &= \left(1 + (wt)^2\right)^2 \cdot \mathbf{R}(t)\mathbf{A}\mathbf{R}(t)^T, \\
\mathbf{a}(t) &= \left(1 + (wt)^2\right)^2 \cdot \mathbf{R}(t)(\mathbf{A}\mathbf{o}(t) + \mathbf{a}), \\
a_0(t) &= \left(1 + (wt)^2\right)^2 \cdot S_0(\mathbf{o}(t)), \\
\mathbf{o}(t) &= \tilde{\mathbf{c}} - \mathbf{R}(t)^T(\tilde{\mathbf{c}} + \mathbf{s}(t)).
\end{aligned} \tag{3.19}$$

The degree of $S_t(\mathbf{x})$ in the parameter t is two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition of a translation and a rotation. Similarly, if a plane associated with $\mathcal{O}_2(0)$ is given by $\mathbf{n}^T\mathbf{x} = n_0$, then at time t this equation has the form

$$\begin{aligned}
\mathbf{n}(t)^T\mathbf{x} &= n_0(t) \quad \text{with} \\
\mathbf{n}(t) &= \left(1 + (wt)^2\right) \cdot \mathbf{R}(t)\mathbf{n}, \\
n_0(t) &= \left(1 + (wt)^2\right) \cdot \left(n_0 - \mathbf{n}^T\mathbf{o}(t)\right),
\end{aligned} \tag{3.20}$$

where $\mathbf{o}(t)$ is defined as in the case of a quadric. The degree of this implicit form in t is one in the case of a translational motion, two in the case of a rotational motion and three in the case of a superposition. Table 3.1 shows the degrees of the coefficients of the implicit forms for the three types of motion.

3.3.2 Specialization for Quadratic Complexes

We show that in the case of quadratic complexes the task of computing the potential collision times can be reduced to polynomial equations. We analyze the degrees of these equations for the case of pure translational motions, pure rotational motions and

[1] If the time interval $[0, 1/20]$ is normalized to $[0,1]$, then the velocities are multiplied by 20.

	n	n_0	$\mathbf{A}$	a	a_0
T	0	1	0	1	2
R	2	2	4	4	4
T+R	2	3	4	5	6

Table 3.1: The degrees of the coefficients of the implicit forms (3.19) and (3.20). The abbreviations T, R and T+R denote a pure translation, a pure rotation and a superposition of both, respectively.

superpositions of translations and rotations. The point-on-edge tests and point-in-face tests that occur in the generic algorithm are performed in the same way as shown in 3.2.2. At the end, we show how the penetration test can be performed in non-degenerate situations. However, we are not able to perform this test efficiently in all cases. We show that the penetration test can be formulated as a system of polynomial inequalities in four variables for which the solvability in the neighbourhood of the origin has to be decided. It is an open problem how this decision can be made efficiently in all cases.

In the following paragraphs we derive polynomial equations for the potential collision times in the vertex-face test, the edge-edge test, the edge-face test and the face-face test. The maximum degrees of these equations are summarized in table 3.2. In most cases we are not able to compute these polynomials symbolically, not even with the help of a computer algebra system on a machine with 256 MB ram. This is because they are defined as determinants of matrices whose entries are polynomials with very complicated coefficients. Thus, we propose to use the methods presented in 2.4.5 to find the roots of the polynomials in matrix representation. But nevertheless, we will determine the degrees of the polynomials and, in some cases, prove that they can be factorized. Besides the fact that these degree bounds are of theoretical interest, they can be very useful if one wants to determine the coefficients of the polynomials using interpolation techniques. Hence, it is our aim to keep these degrees as low as possible and therefore, we will make case distinctions on the specific curve and surface types as can be seen in table 3.2. We try to discuss these cases as thoroughly as possible.

Whereas the vertex-face test is straightforward, the remaining tests are far more complex. In the discussion of the edge-edge test we spend a lot of effort on showing that the resulting polynomial can be factored. In the edge-face test and the face-face test we make use of our results presented in 2.4.3

Vertex-Face Test

We describe how to compute the potential collision times between a vertex $\mathbf{p}$ of $\mathcal{O}_1$ and a face $\mathcal{F}$ of $\mathcal{O}_2$ and vice versa. First, let $\mathbf{p}$ be a vertex of $\mathcal{O}_1$. Let $\mathcal{F}$ be embedded in the quadric $\mathcal{Q}$. We apply a translation to the scene in such a way that $\mathbf{p} = 0$. Let the quadratic form defining $\mathcal{Q}$ after this transformation be given in the form (2.1). A necessary condition for $\mathbf{p} \in \mathcal{F}$ is obviously $\mathbf{p} \in \mathcal{Q}$ which in the case of $\mathbf{p}$ lying in the

	Translation	Rotation	Superposition
Vertex-Face Test			
vertex vs. plane	1	2	3
vertex vs. general quadric	2	4	6
Edge-Edge Test			
straight line vs. straight line	1	4	5
straight line vs. general conic	2	4	6
general conic vs. general conic	4	8	12
Edge-Face Test			
parabola vs. plane	1	4	5
ellipse/hyperbola vs. plane	2	4	6
straight line vs. general quadric	2	4	6
general conic vs. general quadric	8	16	24
Face-Face Test			
plane vs. central quadric	2	4	6
plane vs. non-central quadric	1	4	5
general quadric vs. general quadric	12	24	36

Table 3.2: The maximum degrees of the polynomial equations for the potential collision times.

origin is equivalent to $a_0 = 0$. We use the equations (3.19) to obtain an equation in the parameter t for the points in time when the vertex lies on $\mathcal{Q}$. If we have a pure translation, i.e. $\boldsymbol{\omega} = 0$, then $a_0(t) = S_0(-\mathbf{v} \cdot t)$, which is a polynomial of degree two in t. In the case of a pure rotation, i.e. $\mathbf{v} = 0$, we have $a_0(t) = (1 + (wt)^2)^2 \cdot S_0(\tilde{\mathbf{c}} - \mathbf{R}(t)^T \tilde{\mathbf{c}})$, which is a polynomial of degree four. If the motion consists of a superposition of a translation and a rotation, then the resulting degree is six.

Now we consider $\mathbf{p}$ to be a vertex of $\mathcal{O}_2$ and $\mathcal{F}$ a face of $\mathcal{O}_1$. Again, we denote the quadric containing $\mathcal{F}$ as $\mathcal{Q}$. The position of the vertex at time t can be expressed as

$$\mathbf{p}(t) = \mathbf{R}(t)(\tilde{\mathbf{p}} - \tilde{\mathbf{c}}) + \tilde{\mathbf{c}} + \mathbf{s}(t).$$

We insert this into the quadratic form of $\mathcal{Q}$. In the case $\boldsymbol{\omega} = 0$ this leads to a quadratic equation. Otherwise, we multiply with the degree four denominator to obtain a polynomial of degree four in the case of a pure rotation and degree six in the case of a superposition.

If we assume that $\mathcal{F}$ is embedded on a plane rather than on a quadric, then we proceed analogously. In that case we obtain a linear equation in the case of a pure translation, a quadratic equation in the case of a pure rotation and a cubic one in the case of a superposition. To summarize all this we state the following theorem.

Theorem 3.11. *Finding the potential collision times between a vertex and a quadric can be reduced to finding the roots of polynomials of degree at most two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition. In the case of a vertex and a plane, the degrees are one, two and three, respectively.*

Edge-Edge Test

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be edges of $\mathcal{O}_1$ and $\mathcal{O}_2$, respectively. With $\mathcal{C}_1$ and $\mathcal{C}_2$ we denote the two conics containing these edges. First, we compute a superset $\mathcal{T}$ of the potential collision times. Then, we check for each time $t_c \in \mathcal{T}$ whether it is a potential collision time, i.e. whether $\mathcal{C}_1$ and $\mathcal{C}_2(t_c)$ intersect. If this is the case we compute the respective intersection points. We start with the computation of the superset of the potential collision times. We make a case distinction whether the curves are straight lines or not.

Case 1: Both $\mathcal{C}_1$ and $\mathcal{C}_2$ are straight lines. Let $\mathcal{C}_i$ be given by a vertex $\mathbf{a}_i$ and a direction $\mathbf{u}_i$. Obviously, the two lines can only intersect if the point $\mathbf{a}_2$ lies in the plane containing $\mathbf{a}_1$ and spanned by the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$. This means that $\mathbf{u}_1, \mathbf{u}_2$ and $\mathbf{a}_2 - \mathbf{a}_1$ are linearly dependent, i.e.

$$\det[\mathbf{u}_1, \mathbf{u}_2(t), \mathbf{a}_2(t) - \mathbf{a}_1] = 0. \tag{3.21}$$

The direction vector $\mathbf{u}_2(t)$ and the vertex $\mathbf{a}_2(t)$ are given by

$$\begin{aligned} \mathbf{u}_2(t) &= \mathbf{R}(t)\tilde{\mathbf{u}}_2 \quad \text{and} \\ \mathbf{a}_2(t) &= \mathbf{R}(t)(\tilde{\mathbf{a}}_2 - \tilde{\mathbf{c}}) + \tilde{\mathbf{c}} + \mathbf{v}t. \end{aligned}$$

In the case of $\boldsymbol{\omega} = \mathbf{0}$ equation (3.21) is linear in t. If $\boldsymbol{\omega} \neq \mathbf{0}$, then we multiply the second and third row on the left hand side of (3.21) by $1 + (wt)^2$ and obtain a degree four polynomial in the case of $\mathbf{v} = \mathbf{0}$ and a degree five polynomial in the case of a superposition.

Case 2: Neither $\mathcal{C}_1$ nor $\mathcal{C}_2$ is a straight line. We assume that both curves are given by the intersection between a quadric and a plane. Let the quadric $\mathcal{Q}_1$ containing $\mathcal{C}_1$ be given by the matrix $\mathbf{A}_H$ and let the plane containing that curve have the equation $\mathbf{m}^T\mathbf{x} = m_0$. With $\mathcal{Q}_2, \mathbf{B}_H, \mathbf{n}$ and n_0 we denote the respective quantities for $\mathcal{C}_2$. In order to keep the equations easier we additionally assume the following.

- $\mathbf{m} = [0, 0, 1]^T$ and $m_0 = 0$, i.e. $\mathcal{C}_1$ lies in the (x_1, x_2)-plane.

- $\mathbf{A}$ has diagonal form, $a_{11} \neq 0$ and $a_1 = 0$.

This can always be achieved by applying an appropriate translation and a rotation. If there is an intersection point $\mathbf{x}$ between $\mathcal{C}_1$ and $\mathcal{C}_2$, then this point is a solution of the following system:

$$\begin{aligned} \mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{x}^T\mathbf{a} + a_0 &= 0, \\ \mathbf{x}^T\mathbf{B}\mathbf{x} + 2\mathbf{x}^T\mathbf{b} + b_0 &= 0, \\ \mathbf{x}^T\mathbf{m} - m_0 &= 0, \\ \mathbf{x}^T\mathbf{n} - n_0 &= 0. \end{aligned} \tag{3.22}$$

As $\mathcal{C}_1$ lies in the (x_1, x_2)-plane, the third of these equations simplifies to $x_3 = 0$. Substituting this into the remaining equations and using the above assumptions we obtain three equations in two variables, namely

$$\begin{array}{rclcl} f_1(x_1,x_2) & = & a_{11}x_1^2 + a_{22}x_2^2 + 2a_2x_2 + a_0 & = & 0, \\ f_2(x_1,x_2) & = & b_{11}x_1^2 + b_{22}x_2^2 + 2b_{12}x_1x_2 + 2b_1x_1 + 2b_2x_2 + b_0 & = & 0, \\ f_3(x_1,x_2) & = & n_1x_1 + n_2x_2 - n_0 & = & 0. \end{array}$$

We eliminate the variable x_2 and reduce this system to two equations in one variable by setting

$$\begin{array}{rcll} g_1(x_1) & = & \mathrm{res}_{x_2}(f_1, f_3)(x_1) & \text{and} \\ g_2(x_1) & = & \mathrm{res}_{x_2}(f_2, f_3)(x_1). & \end{array}$$

Both these polynomials have degree two in x_1. We eliminate x_1 from this system by defining the function $D(\mathcal{C}_1, \mathcal{C}_2) = \mathrm{res}_{x_1}(g_1, g_2)$. This function has the value zero whenever the two curves have a common point. In order to get a polynomial equation in the time parameter we replace n_0 and the entries of $\mathbf{n}$ and $\mathbf{B}_H$ using the relationships (3.19) and (3.20). Expanding the algebraic expression for the resultant $D(\mathcal{C}_1, \mathcal{C}_2)$, we get a sum of 106 terms each of which is a product of six time dependent quantities. Replacing these quantities leads to a huge polynomial. In the case of $\boldsymbol{\omega} \neq \mathbf{0}$ it was not possible to compute this polynomial symbolically with the computer algebra system Maple® on a machine with 256 MB ram. In the case $\boldsymbol{\omega} = \mathbf{0}$ we immediately verify that the total degree of f_1 and f_2 in the variables x_1, x_2 and t is two and the total degree of f_3 is these variables is one. Thus, g_1 and g_2 define conic sections in the (x_1, t)-plane and therefore the degree of their resultant has degree four in t. If a rotation is involved, we replace the time dependent quantities in the 4×4 Sylvester matrix of g_1 and g_2 rather than in the resultant. This is the same as replacing them in the polynomials g_1 and g_2. The polynomial f_1 is always constant in t. In the case of $\mathbf{v} = \mathbf{0}$ the degrees of the polynomials f_2 and f_3 in t are four and two, respectively. The matrices of the degrees in t of the entries of $\mathbf{Syl}_{x_2}(f_1, f_3)$ and $\mathbf{Syl}_{x_2}(f_2, f_3)$ are

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & -\infty \\ -\infty & 2 & 2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 4 & 4 & 4 \\ 2 & 2 & -\infty \\ -\infty & 2 & 2 \end{bmatrix},$$

respectively. Thus, the degrees of g_1 and g_2 in t are four and eight, respectively. Using Maple® we find that $(1 + (wt)^2)^2$ is a factor of each coefficient of g_2 as a polynomial in x_1. Therefore, we can divide the last two rows of $\mathbf{Syl}_{x_1}(g_1, g_2)$ by this factor and obtain a matrix polynomial $\mathbf{M}(t)$ of degree four in t. The potential collision times are roots of the determinant $\det(\mathbf{M}(t))$. We compute all roots using the techniques introduced in 2.4.5. The reason why these roots form a proper superset of the potential collision times in general is that some of them might only be extended to a solution of 3.22 by points $\mathbf{x}$ with complex coordinates.

In the case of a superposition of a rotation and a translation the degrees of the polynomials f_2 and f_3 in t are six and three, respectively. The degree matrices are

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 3 & -\infty \\ -\infty & 2 & 3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 4 & 5 & 6 \\ 2 & 3 & -\infty \\ -\infty & 2 & 3 \end{bmatrix}.$$

Hence, the degrees of g_1 and g_2 in t are six and ten, respectively. Again we find that each coefficient of g_2 as a polynomial in x_1 is divisible by $(1 + (wt)^2)^2$. Thus, we can again divide the last two rows of $\mathbf{Syl}_{x_1}(g_1, g_2)$ by this factor and obtain a matrix polynomial $\mathbf{M}(t)$ of degree six. Again, we use the techniques described in 2.4.5 to find the roots of $\det(\mathbf{M}(t))$, which form a superset of the potential collision times.

With the help of Maple® we find that the resultant of g_1 and g_2 can be written in the form $D(\mathcal{C}_1, \mathcal{C}_2) = n_2^4 \cdot h$. This means that in the case that n_2 is constantly zero this resultant is constantly zero, as well. In this case we reverse the order of eliminating the variables in the above approach. We first eliminate x_1 from f_1, f_2 and f_3. Then we eliminate x_2 from the resulting polynomials $\tilde{g}_1$ and $\tilde{g}_2$. Let us denote the resultant of these two polynomials w.r.t. x_2 by $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2)$. Again, with the help of Maple® we observe that this polynomial can be written in the form $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2)(t) = n_1^4 \cdot h$ with the same factor h. If n_1 is also constantly zero, then $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2)$ is constantly zero, as well. But this means that $\mathbf{n} \equiv [0, 0, \pm 1]^T$, which means that either $\boldsymbol{\omega} = \mathbf{0}$ or $\boldsymbol{\omega}$ is parallel to $\mathbf{n}(0)$ which is again parallel to the x_3-axis. In this case we have to look for the roots of $n_0(t)$, which is linear in the case of a translation, quadratic in the case of a rotation and cubic in the case of a superposition. If n_0 is also constantly zero, then $\mathbf{v}$ lies in the (x_1, x_2)-plane and we have to find the collision times of two conics in the plane. This problem also arises as a subproblem in the edge-face test and therefore we will not discuss it here.

Now we assume that $\mathbf{n}$ is not constantly parallel to the x_3-axis. Then the candidates for the potential collision times are either roots of $n_1(t), n_2(t)$ or $h(t)$. Therefore, we want to analyze the degree of the polynomial h. Since we cannot factor n_2 out of $\mathbf{Syl}_{x_1}(g_1, g_2)$ and we are not able to compute $D(\mathcal{C}_1, \mathcal{C}_2)(t)$ symbolically, we cannot compute $h(t)$ symbolically, as well. But if we know its degree, then we can use interpolation methods as described in 2.4.5 to find its roots. If we compute the degree matrix of the Sylvester matrix of g_1 and g_2 we get

$$\begin{bmatrix} 4 & 5 & 6 & -\infty \\ -\infty & 4 & 5 & 6 \\ 8 & 9 & 10 & -\infty \\ -\infty & 8 & 9 & 10 \end{bmatrix}.$$

From this we can conclude that the degree of $D(\mathcal{C}_1, \mathcal{C}_2)(t)$ is 28 in the case of a superposition. Similarly, we get degree 24 in the case of a pure rotation. Since $\mathbf{n}$ has degree two in t and $D(\mathcal{C}_1, \mathcal{C}_2) = n_2^4 \cdot h$, we conclude that h has degree 16 in the case of a rotation and 20 in the case of a superposition. We saw that we can divide the last two rows of the Sylvester matrix by $(1 + w^2t^2)^2$ and obtained the matrix polynomial $\mathbf{M}(t)$. Thus, the degree of $\det(\mathbf{M}(t))$ is 16 in the rotational case and 20 in the case of a superposition. The question is, whether the determinant of $\mathbf{M}(t)$ still can be divided by $n_2(t)^4$ or – putting it the other way round – whether h can be written as $h(t) = (1 + (w^2t^2)^4 \cdot r(t)$. The answer to this question is yes, which we prove in the following.

Lemma 3.12. *If* $\boldsymbol{\omega} \times \mathbf{n}(0) \neq 0$*, then at most one component of* $\mathbf{n}(t)$ *is divisible by* $1 + (wt)^2$*.*

Proof. Suppose that $1 + (wt)^2$ divides two components of $\mathbf{n}(t)$. Because of the relationship (3.20) we have $\mathbf{n}(i/w) = 2(\boldsymbol{w} \times (\tilde{\mathbf{n}} \times \boldsymbol{w}) + i(\boldsymbol{w} \times \tilde{\mathbf{n}}))$, where we wrote $\tilde{\mathbf{n}}$ for $\mathbf{n}(0)$. Since two components of this vector must be zero, it follows that the real part $\boldsymbol{w} \times (\tilde{\mathbf{n}} \times \boldsymbol{w})$ as well as the imaginary part $\boldsymbol{w} \times \tilde{\mathbf{n}}$ must be parallel to one of the coordinate axes. In particular, these two vectors must be parallel to each other, i.e.

$$\begin{aligned} (\boldsymbol{w} \times \tilde{\mathbf{n}}) \times \big(\boldsymbol{w} \times (\tilde{\mathbf{n}} \times \boldsymbol{w})\big) &= 0 \quad \Leftrightarrow \\ (\tilde{\mathbf{n}} \times \boldsymbol{w})^2 \boldsymbol{w} &= 0. \end{aligned}$$

This means that $\boldsymbol{w} \times \tilde{\mathbf{n}} = 0$ which is a contradiction. □

Therefore, not both $n_1(t)$ and $n_2(t)$ are divisible by $1 + (wt)^2$. Since $(1 + (wt)^2)^4$ is a factor of both $D(\mathcal{C}_1, \mathcal{C}_2)(t) = n_2(t)^4 \cdot h(t)$ and $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2)(t) = n_1(t)^4 \cdot h(t)$, this lemma implies that there is a polynomial r such that $h(t) = (1 + (wt)^2)^4 \cdot r(t)$ whenever $\boldsymbol{\omega} \times \mathbf{n}(0) \neq 0$. In order to show that $h(t)$ can also be factorized in this way if $\boldsymbol{\omega}$ and $\mathbf{n}(0)$ are parallel, we need the following lemma.

Lemma 3.13. *Let* $f, g \in \mathbb{C}[x_1, \ldots, x_n]$ *be polynomials and let* $g \neq 0$*. If* $f(\xi_1, \ldots, \xi_n) = 0$ *for all* $(\xi_1, \ldots, \xi_n) \in \mathbb{C}^n$ *satisfying* $g(\xi_1, \ldots, \xi_n) \neq 0$*, then* $f = 0$*.*

Proof. For a polynomial p and an ideal I we write $V(p)$ and $V(I)$ for the varieties defined by p and I, respectively. Further, we write $\langle p \rangle$ for the ideal defined by p. If $\mathcal{P}$ is a pointset, then we write $\overline{\mathcal{P}}$ for the Zariski closure of $\mathcal{P}$. First, we notice, that

$\overline{\mathbb{C}^n \setminus V(g)} \subset V(f)$. Next, by theorem 7 from [CLO97] we have

$$\begin{aligned}
\overline{\mathbb{C}^n \setminus V(g)} &= \overline{V(\langle 0\rangle) \setminus V(\langle g\rangle)} \\
&= V\big(\{h \in \mathbb{C}[x_1, \ldots, x_n] \,|\, hq \in \langle 0\rangle \text{ for all } q \in \langle g\rangle\}\big) \\
&= V(\langle 0\rangle) \\
&= \mathbb{C}^n.
\end{aligned}$$

Hence, $V(f) = \mathbb{C}^n$ and therefore $f = 0$. □

This statement is intuitively obvious. If f takes for all but a zero-set of points the value zero, then it should be the zero polynomial. Now we can show that we can factorize the polynomial h. The following corollary states the more general result that the resultant of g_1 and g_2 can be factorized.

Corollary 3.14. $D(\mathcal{C}_1, \mathcal{C}_2)(t) = n_2(t)^4 \cdot (1 + (wt)^2)^4 \cdot r(t)$ *for all conics* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *and all velocities* $\mathbf{v}$ *and angular velocities* $\boldsymbol{\omega}$.

Proof. We already know that $D(\mathcal{C}_1, \mathcal{C}_2)(t) = n_2(t)^4 \cdot h(t)$. For situations where $\mathbf{n}(t)$ is constantly parallel to the x_3-axis we know that $n_2(t)$ is constantly zero and thus, the claim is trivial. Otherwise, define the polynomial $g(\boldsymbol{\xi}) = (\boldsymbol{\omega} \times \mathbf{n}(0))^2$, where $\boldsymbol{\xi}$ is the vector of all generic coefficients defining $\mathcal{C}_1, \mathcal{C}_2$, $\boldsymbol{\omega}$ and $\mathbf{v}$ including t. Obviously, g is not the zero polynomial. With lemma 3.12 we have seen that $f(\boldsymbol{\xi}) = h(t) - (1+(wt)^2)^4 \cdot r(t) = 0$ for all $\boldsymbol{\xi}$ satisfying $g(\boldsymbol{\xi}) \neq 0$. Hence, lemma 3.13 implies that f is the zero polynomial which completes the proof of the corollary. □

Together with our considerations concerning the degree of $D(\mathcal{C}_1, \mathcal{C}_2)(t)$ this corollary shows that we can find a superset of the potential collision times between two conics by finding the roots of polynomials of degree at most four in the case of a pure translation, eight in the case of a pure rotation and twelve in the case of a superposition.

Case 3: $\mathcal{C}_1$ is a straight line and $\mathcal{C}_2$ is a conic. We assume that $\mathcal{C}_1$ is the x_1-axis. This can be achieved by applying an appropriate translation and a rotation. As in the case of two conics we start with the equations (3.22). Since $\mathcal{C}_1$ is the x_1-axis, we replace the first of these equations by $x_2 = 0$ and – as before – the third by $x_3 = 0$. Substituting this in the remaining two equations, we get the system

$$\begin{aligned}
g_1(x_1) &= b_{11}x_1^2 + 2b_1x_1 + b_0 \quad \text{and} \\
g_2(x_1) &= n_1x_1 - n_0.
\end{aligned}$$

As before, we define $D(\mathcal{C}_1, \mathcal{C}_2) = \mathrm{res}_{x_1}(g_1, g_2)$ and obtain

$$D(\mathcal{C}_1, \mathcal{C}_2) = n_0^2b_{11} + n_1^2b_0 + 2n_0n_1b_1.$$

By table 3.1 we find that the degree of $D(\mathcal{C}_1, \mathcal{C}_2)(t)$ is two in the case of a translation, eight in the case of a rotation and ten in the case of a superposition. If we evaluate this polynomial with randomly chosen coefficients using Maple®, then we see that it factorizes as $(1 + (wt)^2)^2 \cdot r(t)$. We want to prove that this is always possible.

Lemma 3.15. $D(\mathcal{C}_1, \mathcal{C}_2)(t) = (1 + (wt)^2)^2 \cdot r(t)$ *for all straight lines* $\mathcal{C}_1$ *and conics* $\mathcal{C}_2$ *and all velocities* $\mathbf{v}$ *and angular velocities* $\boldsymbol{\omega}$.

Proof. We define two polynomials $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2)$ and $\overline{D}(\mathcal{C}_1, \mathcal{C}_2)$ as follows. Let $\tilde{f}_1(x_1, x_2) = x_2$ be the first one of the four polynomials in (3.22). Then we substitute $x_3 = 0$ into the second and the fourth of these polynomials which yields $\tilde{f}_2(x_1, x_2)$ and $\tilde{f}_3(x_1, x_2)$, respectively. We have the system of equations

$$\begin{array}{rclcl}
\tilde{f}_1(x_1, x_2) & = & x_2 & = & 0, \\
\tilde{f}_2(x_1, x_2) & = & b_{11}x_1^2 + b_{22}x_2^2 + 2b_{12}x_1x_2 + 2b_1x_1 + 2b_2x_2 + b_0 & = & 0, \\
\tilde{f}_3(x_1, x_2) & = & n_1x_1 + n_2x_2 - n_0 & = & 0.
\end{array}$$

For $i = 1, 2$ we define $\tilde{g}_i(x_1) = \operatorname{res}_{x_2}(\tilde{f}_i, \tilde{f}_3)(x_1)$. This means, $\tilde{g}_1(x_1) = n_1x_1 - n_0$ and $\tilde{g}_2(x_1)$ is the same polynomial as $g_2(x_1)$ in the case of two conics. Then we set $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2) = \operatorname{res}_{x_1}(\tilde{g}_1, \tilde{g}_2)$. If we evaluate this resultant, we realize that $\tilde{D}(\mathcal{C}_1, \mathcal{C}_2) = n_2^2 \cdot D(\mathcal{C}_1, \mathcal{C}_2)$.

$\overline{D}(\mathcal{C}_1, \mathcal{C}_2)$ is defined similarly. Let $\overline{f}_1(x_1, x_3)$ be the third one of the polynomials in (3.22). Then we substitute $x_2 = 0$ into the remaining two equations and obtain the system

$$\begin{array}{rclcl}
\overline{f}_1(x_1, x_3) & = & x_3 & = & 0, \\
\overline{f}_2(x_1, x_3) & = & b_{11}x_1^2 + b_{33}x_3^2 + 2b_{13}x_1x_3 + 2b_1x_1 + 2b_3x_3 + b_0 & = & 0, \\
\overline{f}_3(x_1, x_3) & = & n_1x_1 + n_3x_3 - n0 & = & 0.
\end{array}$$

For $i = 1, 2$ we define $\overline{g}_i(x_1) = \operatorname{res}_{x_3}(\overline{f}_i, \overline{f}_3)(x_1)$. In particular, $\overline{g}_1(x_1) = \tilde{g}_1(x_1)$. Now, we set $\overline{D}(\mathcal{C}_1, \mathcal{C}_2) = \operatorname{res}_{x_1}(\overline{g}_1, \overline{g}_2)$. Evaluating this resultant, we observe that $\overline{D}(\mathcal{C}_1, \mathcal{C}_2) = n_3^2 \cdot D(\mathcal{C}_1, \mathcal{C}_2)$.

Since $\tilde{g}_2$ is equal to the polynomial g_2 in the case of two general conics, we already know that this polynomial is divisible by $(1 + (wt)^2)^2$. Using Maple® we find that this is also true for $\overline{g}_2$. The polynomials $\tilde{g}_1$ and $\overline{g}_1$ are linear in x_1 and hence, $(1 + (wt)^2)^2$ is a factor of one row of both $\mathbf{Syl}_{x_1}(\tilde{g}_1, \tilde{g}_2)$ and $\mathbf{Syl}_{x_1}(\overline{g}_1, \overline{g}_2)$. Since $\tilde{D} = n_2^2 \cdot D$ and $\overline{D} = n_3^2 \cdot D$, it follows from lemma 3.12 that $D(\mathcal{C}_1, \mathcal{C}_2)(t) = (1 + (wt)^2)^2 \cdot r(t)$ whenever $\boldsymbol{\omega} \times \mathbf{n}(0) \neq \mathbf{0}$. As in the proof of corollary 3.14, lemma 3.13 implies that this is true in all cases. □

This lemma together with our considerations concerning the degree of $D(\mathcal{C}_1, \mathcal{C}_2)$ implies that the degree of the polynomial $r(t)$ is two in the case of a translation, four in the case of a rotation and six in the case of a superposition.

Case 4: $\mathcal{C}_1$ is a conic and $\mathcal{C}_2$ is a straight line. As in case 2, we start with the equations (3.22) and assume again that $\mathbf{m} = [0, 0, 1]^T$ and that $\mathbf{A}$ is a diagonal matrix and $a_{11} \neq 0$ and $a_1 = 0$. Since $\mathcal{C}_2$ is a straight line we replace the second equation in (3.22) by

$$\mathbf{x}^T\mathbf{b} - b_0 = 0.$$

After inserting $x_3 = 0$ into the first, second and fourth equation we get the system

$$\begin{array}{rcl} f_1(x_1,x_2) & = & a_{11}x_1^2 + a_{22}x_2^2 + 2a_2x_2 + a_0 & = & 0, \\ f_2(x_1,x_2) & = & b_1x_1 + b_2x_2 - b_0 & = & 0, \\ f_3(x_1,x_2) & = & n_1x_1 + n_2x_2 - n_0 & = & 0. \end{array}$$

We define $g_i(x_1) = \mathrm{res}_{x_2}(f_i, f_3)(x_1)$ and then $D(\mathcal{C}_1,\mathcal{C}_2) = \mathrm{res}_{x_1}(g_1, g_2)$. In the case $\omega = 0$ we observe that the total degree of f_1 in x_1, x_2 and t is two and the total degrees of f_2 and f_3 in these variables are one. Hence, g_1 defines a conic section and g_2 a straight line in the (x_1,t)-plane. Thus, the degree of $D(\mathcal{C}_1,\mathcal{C}_2)$ in t is two in that case. If the motion is a pure rotation, then the matrices of the degrees in t of the entries of $\mathbf{Syl}_{x_2}(f_i, f_3)$ are

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 2 & -\infty \\ -\infty & 2 & 2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix},$$

respectively. Thus, the degrees of both g_1 and g_2 in t are four. With the help of Maple® we see that each coefficient of g_2 as polynomial in x_1 can be divided by $1+(wt)^2$. Hence, we can divide the last two rows of $\mathbf{Syl}_{x_1}(g_1, g_2)$ by this factor and obtain a (3×3)-matrix polynomial of degree four in t, that has the degree matrix

$$\begin{bmatrix} 4 & 4 & 4 \\ 2 & 2 & -\infty \\ -\infty & 2 & 2 \end{bmatrix}.$$

This implies that $D(\mathcal{C}_1,\mathcal{C}_2)(t)$ factorizes as $(1+(wt)^2)^2 \cdot h(t)$ where $h(t)$ has degree eight in t. In the case of a superposition of a translation and a rotation the degree matrices are

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 3 & -\infty \\ -\infty & 2 & 3 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 2 & 3 \\ 2 & 3 \end{bmatrix}$$

from where we conclude that the degrees of g_1 and g_2 in t are six and five, respectively. Again, we can divide the last two rows of $\mathbf{Syl}_{x_1}(g_1, g_2)$ by $1 + (wt)^2$ and we obtain a (3×3)-matrix polynomial of degree six in t that has the degree matrix

$$\begin{bmatrix} 4 & 5 & 6 \\ 2 & 3 & -\infty \\ -\infty & 2 & 3 \end{bmatrix}.$$

Thus, we conclude that $D(\mathcal{C}_1,\mathcal{C}_2)(t)$ can be factorized in the form $(1 + (wt)^2)^2 \cdot h(t)$ where $h(t)$ has degree ten in t.

If we evaluate $D(\mathcal{C}_1,\mathcal{C}_2)$ symbolically before we replace the coefficients of $\mathcal{C}_2$ by time dependent functions we observe that we can also factorize this determinant in the form $D(\mathcal{C}_1,\mathcal{C}_2) = n_2^2 \cdot r(t)$. Similar to case 2 we can define a function $\tilde{D}(\mathcal{C}_1,\mathcal{C}_2)$ by changing the order of variable elimination. We find that this function can be written in the form $n_1^2 \cdot r(t)$ with the same polynomial r. With the same argument as in the case of two conics we can prove that there is a polynomial $p(t)$ such that we can write $D(\mathcal{C}_1, \mathcal{C}_2)(t) = n_2(t)^2 \cdot (1 + (wt)^2)^2 \cdot p(t)$, where $p(t)$ has degree four in the case of a pure rotation and six in the case of a superposition.

Now we describe how to check whether a given point t_c in time is a potential collision time. This means we check whether $\mathcal{C}_1$ and $\mathcal{C}_2(t_c)$ intersect and if this is the case we compute the intersection points. For the sake of simplicity we omit the time parameter now and write $\mathcal{C}_2$ instead of $\mathcal{C}_2(t_c)$. Again we make a case distinction whether the curves are straight lines or not.

First, let both $\mathcal{C}_1$ and $\mathcal{C}_2$ be straight lines. In this case we know that equation (3.21) holds for $t = t_c$. Hence, the lines intersect if and only if the equation

$$\mathbf{a}_1 + \lambda \mathbf{u}_1 = \mathbf{a}_2 + \mu \mathbf{u}_2$$

has a real solution (λ, μ). If $\mathbf{u}_1 \times \mathbf{u}_2 = \mathbf{0}$, then the lines are either parallel or identical. If additionally $\mathbf{u}_1 \times (\mathbf{a}_2 - \mathbf{a}_1) = \mathbf{0}$ they are identical and we may choose any point on them as an intersection point. Otherwise, the lines are parallel. So let $\mathbf{u}_1 \times \mathbf{u}_2 \neq \mathbf{0}$. Then, by Cramer's rule we have an intersection with

$$\begin{aligned} \lambda &= \frac{\det[\mathbf{a}_2 - \mathbf{a}_1, \mathbf{u}_2, \mathbf{u}_1 \times \mathbf{u}_2]}{(\mathbf{u}_1 \times \mathbf{u}_2)^2} \quad \text{and} \\ \mu &= \frac{\det[\mathbf{a}_2 - \mathbf{a}_1, \mathbf{u}_1, \mathbf{u}_1 \times \mathbf{u}_2]}{(\mathbf{u}_1 \times \mathbf{u}_2)^2}. \end{aligned}$$

Thus, we have solved only linear equations in order to perform this intersection test.

Now, we consider the case that not both curves are straight lines. W.l.o.g. we assume that $\mathcal{C}_2$ is not straight. In this case we first determine the intersection points between $\mathcal{C}_1$ and the plane $\mathcal{P}$ that contains $\mathcal{C}_2$ by inserting the parameterization of $\mathcal{C}_1$ into the implicit form of $\mathcal{P}$ and solving the resulting equation. This equation is linear in the case that $\mathcal{C}_1$ is a straight line and quadratic otherwise. If such an intersection point does not exist, i.e. the equation has no real solution, we know that t_c is not a potential intersection point. Otherwise, there are either finitely or infinitely many intersection points, depending on whether $\mathcal{C}_1$ lies in $\mathcal{P}$ or not. If $\mathcal{C}_1$ intersects $\mathcal{P}$ in finitely many points, let $\mathbf{p}$ be such an intersection point. We must check whether $\mathbf{p}$ lies on the curve $\mathcal{C}_2$. As stated in section 2.3.3 we know a point $\mathbf{c}$ on the plane $\mathcal{P}$ as well as two orthogonal unit vectors $\mathbf{u}$ and $\mathbf{v}$ forming the coordinate system of that plane such that $\mathcal{C}_2$ is given by the implicit form (2.5). The coordinates of the point $\mathbf{p}$ in that coordinate system are $[\lambda, \mu] = [(\mathbf{p} - \mathbf{c})^T \mathbf{u}, (\mathbf{p} - \mathbf{c})^T \mathbf{v}]$. The curve $\mathcal{C}_2$ contains $\mathbf{p}$ if and only if the coordinates $[\lambda, \mu]$ satisfy the form (2.5).

In the case of $\mathcal{C}_1$ lying in the plane $\mathcal{P}$ we express the parameterization $\mathbf{x}(s)$ of $\mathcal{C}_1$ in the coordinate system of $\mathcal{P}$ formed by the vectors $\mathbf{u}$ and $\mathbf{v}$ and the origin $\mathbf{c}$. The coordinates $[\lambda(s), \mu(s)]$ of $\mathbf{x}(s)$ in this system are $[(\mathbf{x}(s) - \mathbf{c}))^T \mathbf{u}, (\mathbf{x}(s) - \mathbf{c})^T \mathbf{v}]$. We insert this into the implicit form (2.5). In the case of $\mathcal{C}_1$ being a straight line we get a quadratic equation, and if $\mathcal{C}_1$ is a conic we multiply with the denominators to get a polynomial equation of degree four. The solutions of this equation in s give us the desired intersection points

$\mathbf{x}(s)$. Thus, the degree of the polynomial equations is at most two if exactly one of the curves is a straight line and at most four if both curves are conic sections.

We summarize the results in the following theorem.

Theorem 3.16. *Finding the potential collision times between two conics $\mathcal{C}_1$ and $\mathcal{C}_2$ can be reduced to finding the roots of polynomials whose degrees are upper-bounded by the following table.*

	translation	*rotation*	*superposition*
Both $\mathcal{C}_1$ and $\mathcal{C}_2$ are straight lines	1	4	5
$\mathcal{C}_1$ or $\mathcal{C}_2$ is a straight line	2	4	6
Neither $\mathcal{C}_1$ nor $\mathcal{C}_2$ is a straight line	4	8	12

Edge-Face Test

Now we describe how to compute the potential collision times between an edge $\mathcal{E}$ of $\mathcal{O}_1$ and a face $\mathcal{F}$ of $\mathcal{O}_2$ and vice versa. As usual, we denote the curve containing $\mathcal{E}$ by $\mathcal{C}$ and the surface containing $\mathcal{F}$ by $\mathcal{S}$.

Case 1: Let us first consider the case that $\mathcal{E}$ belongs to $\mathcal{O}_1$. We distinguish the cases whether $\mathcal{C}$ is a straight line or not and whether $\mathcal{S}$ is a plane or not. We do not have to consider the line-plane case because whenever a straight line segment collides with a planar face there must be a collision between this segment and the boundary of that face or between the endpoints of the segment and the face. Thus, this case is covered by the edge-edge collision detection and the vertex-face collision detection.

Let $\mathcal{C}$ be a straight line and let $\mathcal{S}$ not be a plane. By applying an appropriate rotation and translation to the scene we can assume that $\mathcal{C}$ is the x_1-axis. The quadric $\mathcal{S}$ is defined by the (4×4)-matrix $\mathbf{A}_H$. Inserting the parameterization $\mathbf{x}(\lambda) = [\lambda, 0, 0, 1]^T$ of the x_1-axis into the quadratic form of $\mathcal{S}$ we get a quadratic polynomial in λ whose roots correspond to the intersection points between $\mathcal{C}$ and $\mathcal{S}$. If there is a tangential intersection point then the discriminant of this polynomial is zero. This discriminant is

$$\Delta(\mathcal{C}, \mathcal{S}) = a_1^2 - a_{11}a_0.$$

We replace a_{11}, a_1 and a_0 in this function by the respective time dependent functions which are given by the relationships (3.19). With the help of Maple® we find that after this substitution the resulting polynomial is divisible by $(1 + (wt)^2)^2$. The resulting degree is six in the case of a superposition of a translation and a rotation, four in the case of a pure rotation and two in the case of a pure translation.

Now, let $\mathcal{C}$ be not a straight line and let $\mathcal{S}$ be a plane. By applying an appropriate rotation and translation to the scene we can assume that $\mathcal{C}$ lies in the (x_1, x_2)-plane and its quadratic form in the variables x_1 and x_2 is given by the matrix

$$\mathbf{A}_H = \begin{bmatrix} a_{11} & 0 & a_1 \\ 0 & a_{22} & a_2 \\ a_1 & a_2 & a_0 \end{bmatrix}.$$

The plane $\mathcal{S}$ is defined by the implicit form $\mathbf{x}^T\mathbf{n} - n_0 = 0$.

Lemma 3.17. *Let $\mathcal{C}$ be a curve in the plane $\mathcal{P}$ and let $\mathcal{S}$ be a surface. Then, $\mathbf{p}$ is a tangential intersection point of $\mathcal{C}$ and $\mathcal{S}$ if and only if $\mathbf{p}$ is a tangential intersection point of $\mathcal{C}$ and the intersection curve between $\mathcal{S}$ and $\mathcal{P}$.*

Proof. Let $\mathbf{c}(t)$ be a parameterization of $\mathcal{C}$ and let $\mathbf{c}(t_0) = \mathbf{p}$. Let the plane $\mathcal{P}$ be given by the equation $\mathbf{x}^T\mathbf{n} = n_0$. As $\mathcal{C}$ lies in $\mathcal{P}$, we have $\mathbf{c}(t)^T\mathbf{n} = n_0$ for all t which implies that $\dot{\mathbf{c}}(t)^T\mathbf{n} = 0$ for all t. Let $\mathbf{m}$ denote the normal of $\mathcal{S}$ in $\mathbf{p}$. The tangent of the intersection curve between $\mathcal{S}$ and $\mathcal{P}$ in the point $\mathbf{p}$ is parallel to the vector $\mathbf{u} = \mathbf{m} \times \mathbf{n}$. To prove the lemma, we must show that $\dot{\mathbf{c}}(t_0)^T\mathbf{m} = 0$ if and only if $\dot{\mathbf{c}}(t_0)$ is parallel to $\mathbf{u}$, i.e. $\dot{\mathbf{c}}(t_0) \times \mathbf{u} = \mathbf{0}$. It holds

$$\begin{aligned} \dot{\mathbf{c}}(t_0) \times \mathbf{u} &= \mathbf{m} \cdot \dot{\mathbf{c}}(t_0)^T\mathbf{n} - \mathbf{n} \cdot \dot{\mathbf{c}}(t_0)^T\mathbf{m} \\ &= -\mathbf{n} \cdot \dot{\mathbf{c}}(t_0)^T\mathbf{m}. \end{aligned}$$

Since $\mathbf{n} \neq \mathbf{0}$, this proves the lemma. □

By this lemma, the tangential intersection points between $\mathcal{C}$ and $\mathcal{S}$ are exactly the tangential intersection points between $\mathcal{C}$ and the intersection line $\mathcal{D}$ between $\mathcal{S}$ and the (x_1, x_2)-plane. The "quadratic" form defined by the matrix

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & n_1 \\ 0 & 0 & n_2 \\ n_1 & n_2 & -n_0 \end{bmatrix}$$

defines the line $\mathcal{D}$. By theorem 2.30, a necessary condition for $\mathcal{C}$ and $\mathcal{D}$ to have a non-singular tangential intersection is that $\chi_{\mathbf{A},\mathbf{B}}$ has a multiple root. Thus, we define $\Delta(\mathcal{C}, \mathcal{S})$ to be the discriminant of $\chi_{\mathbf{A},\mathbf{B}}$. Using Maple® we see that $\Delta(\mathcal{C}, \mathcal{S}) = a_{11}a_{22}\Delta'(\mathcal{C}, \mathcal{S})$ for a polynomial Δ' in the coefficients of $\mathcal{C}$ and $\mathcal{S}$. Hence, $\Delta(\mathcal{C}, \mathcal{S})$ is zero if $\mathcal{C}$ is a parabola. Therefore, we first assume that $\mathcal{C}$ is not a parabola, i.e. it is an ellipse or a hyperbola. If we replace the components of $\mathbf{n}$ as well as n_0 in $\Delta'(\mathcal{C}, \mathcal{S})$ by the corresponding time dependent functions (3.20) we obtain a polynomial in t whose roots are the candidates for the potential collision times. The degree of this polynomial is two in the case of a translation, four in the case of a rotation and six in the case of a superposition. If $\mathcal{C}$ is a parabola, then we can assume that it can be parameterized in the form $x(\lambda) = [\lambda, a\lambda^2, 0, 1]$. Inserting this into the implicit form of $\mathcal{S}$ we obtain a quadratic polynomial

$$f(\lambda) = an_2\lambda^2 + an_1\lambda - n_0$$

whose roots correspond to the intersection points between $\mathcal{C}$ and $\mathcal{S}$. If there is a tangential intersection point then the discriminant $n_1^2 + 4an_2n_0$ is zero. Inserting the time dependent functions (3.20) into this discriminant, we get a polynomial whose roots are the candidates for the potential collision times. The degree of this polynomial is one in the case of a translation, four in the case of a rotation and five in the case of a superposition.

Finally, we consider the case that $\mathcal{C}$ not a straight line and $\mathcal{S}$ is not a plane. By applying an appropriate transformation to the coordinate system we can assume that $\mathcal{C}$ lies in the (x_1, x_2)-plane and that its quadratic form in the variables x_1 and x_2 is given by the matrix

$$\mathbf{A}_H = \begin{bmatrix} a_{11} & 0 & a_1 \\ 0 & a_{22} & a_2 \\ a_1 & a_2 & a_0 \end{bmatrix}.$$

The quadric $\mathcal{S}$ is defined by the (4×4)-matrix $\mathbf{B}_H$. By lemma 3.17, the tangential intersection points between $\mathcal{C}$ and $\mathcal{S}$ are exactly the tangential intersection points between $\mathcal{C}$ and the intersection curve $\mathcal{D}$ between $\mathcal{S}$ and the (x_1, x_2)-plane. We obtain a quadratic form for $\mathcal{D}$ by letting $x_3 = 0$ in the quadratic form of $\mathcal{S}$. This form is given by the matrix $\mathbf{C}_H$ which is obtained from $\mathbf{B}_H$ by removing the third row and the third column. Because of theorem 2.30, a necessary condition for $\mathcal{C}$ and $\mathcal{D}$ to have a non-singular tangential intersection point $\mathbf{p}$ is that $\chi_{A,C}$ either has a multiple real root or is identically zero. A tangential intersection point of $\mathcal{C}$ and $\mathcal{D}$ might be a singular point of $\mathcal{D}$ even though it is not a singular point of $\mathcal{S}$. Thus, in order to have a necessary condition for $\mathcal{C}$ and $\mathcal{S}$ to have a non-singular tangential intersection point we need the following lemma in addition to theorem 2.30.

Lemma 3.18. *Let two conics be given by the (3×3)-matrices $\mathbf{A}_H$ and $\mathbf{B}_H$. If there is a point $\mathbf{p}_H$ such that $\mathbf{p}_H^T\mathbf{A}_H\mathbf{p}_H = 0$ and $\mathbf{B}_H\mathbf{p}_H = \mathbf{0}$, then the degree of $\chi_{A,B}$ is at most one.*

Proof. By lemma 2.13, the coefficients of λ^3 and λ^2 in $\chi_{A,B}(\lambda)$ are $\det \mathbf{B}_H$ and $\operatorname{tr}(\mathbf{A}_H \cdot \operatorname{adj} \mathbf{B}_H)$, respectively. From $\mathbf{B}_H\mathbf{p}_H = \mathbf{0}$ it follows that $\det \mathbf{B}_H = 0$. Hence, it remains to show that $\operatorname{tr}(\mathbf{A}_H \cdot \operatorname{adj} \mathbf{B}_H) = 0$, as well. By applying an appropriate transformation we can assume that $\mathbf{p}_H = [0, 0, 1]^T$. Then, the matrix $\mathbf{B}_H$ must have the form $\mathbf{B}_H = \operatorname{diag}(\mathbf{B}, 0)$. It follows that $\operatorname{adj} \mathbf{B}_H = \operatorname{diag}(0, 0, \det \mathbf{B})$ and therefore $\mathbf{A}_H \cdot \operatorname{adj} \mathbf{B}_H = \operatorname{diag}(0, 0, a_0 \cdot \det \mathbf{B})$. Since $\mathbf{p}_H^T\mathbf{A}_H\mathbf{p}_H = 0$ we must have $a_0 = 0$ and we are done. □

We define the function $\Delta(\mathcal{C},\mathcal{S})$ as the discriminant of $\chi_{\mathbf{A},\mathbf{C}}$, which can be written in the following way:

$$\begin{aligned}\Delta(\mathcal{C},\mathcal{S}) &= \frac{1}{|\mathbf{C}_\mathrm{H}|}\mathrm{res}_\lambda\left(\chi_{\mathbf{A},\mathbf{C}}, \frac{d\chi_{\mathbf{A},\mathbf{C}}}{d\lambda}\right)\\ &= \frac{1}{|\mathbf{C}_\mathrm{H}|}\left|\mathbf{Syl}_\lambda\left(\chi_{\mathbf{A},\mathbf{C}}, \frac{d\chi_{\mathbf{A},\mathbf{C}}}{d\lambda}\right)\right|\\ &= \begin{vmatrix} |\mathbf{C}_\mathrm{H}| & \mathrm{tr}\left(\mathbf{A}_\mathrm{H}\overline{\mathbf{C}}_\mathrm{H}\right) & \mathrm{tr}\left(\mathbf{C}_\mathrm{H}\overline{\mathbf{A}}_\mathrm{H}\right) & |\mathbf{A}_\mathrm{H}| \\ -\mathrm{tr}\left(\mathbf{A}_\mathrm{H}\overline{\mathbf{C}}_\mathrm{H}\right) & -2\mathrm{tr}\left(\mathbf{C}_\mathrm{H}\overline{\mathbf{A}}_\mathrm{H}\right) & -3|\mathbf{A}_\mathrm{H}| & 0 \\ 3|\mathbf{C}_\mathrm{H}| & 2\mathrm{tr}\left(\mathbf{A}_\mathrm{H}\overline{\mathbf{C}}_\mathrm{H}\right) & \mathrm{tr}\left(\mathbf{C}_\mathrm{H}\overline{\mathbf{A}}_\mathrm{H}\right) & 0 \\ 0 & 3|\mathbf{C}_\mathrm{H}| & 2\mathrm{tr}\left(\mathbf{A}_\mathrm{H}\overline{\mathbf{C}}_\mathrm{H}\right) & \mathrm{tr}\left(\mathbf{C}_\mathrm{H}\overline{\mathbf{A}}_\mathrm{H}\right)\end{vmatrix}, \end{aligned} \tag{3.23}$$

where we used the notation $\overline{\mathbf{A}} = \mathrm{adj}\,\mathbf{A}$ and $|\mathbf{A}| = \det\mathbf{A}$. We denote the matrix (3.23) by $\mathbf{M}$.

Lemma 3.19. *If $\mathcal{C}$ and $\mathcal{S}$ have a non-singular tangential intersection point, then $\Delta(\mathcal{C},\mathcal{S}) = 0$.*

Proof. By lemma 3.17, a point $\mathbf{p}$ is a tangential intersection point of $\mathcal{C}$ and $\mathcal{S}$ if and only if $\mathbf{p}$ is a tangential intersection point of $\mathcal{C}$ and $\mathcal{D}$. If $\mathbf{p}$ is a non-singular point of $\mathcal{D}$, then by theorem 2.30 the polynomial $\chi_{\mathbf{A},\mathbf{C}}$ has a multiple root or vanishes identically. Hence, the discriminant $\Delta(\mathcal{C},\mathcal{S}) = 0$. If $\mathbf{p}$ is a singular point of $\mathcal{D}$, then by lemma 3.18 the coefficients $|\mathbf{C}_\mathrm{H}|$ and $\mathrm{tr}\left(\mathbf{A}_\mathrm{H}\overline{\mathbf{C}}_\mathrm{H}\right)$ of λ^3 and λ^2 in $\chi_{\mathbf{A},\mathbf{C}}(\lambda)$ are zero. But then, the first column of the matrix (3.23) is zero. □

Thus, we have a necessary polynomial condition for $\mathcal{C}$ and $\mathcal{S}$ to have a tangential intersection. If we replace the entries of $\mathbf{B}_\mathrm{H}$ by the respective time dependent functions and multiply with the denominator, we get a polynomial whose roots are the candidates for the potential collision times. Similarly to the edge-edge case, we do not make this substitution in the coefficients of the polynomial $\Delta(\mathcal{C},\mathcal{S})$ but in the matrix $\mathbf{M}$ and obtain a matrix polynomial $\mathbf{M}(t)$. The time dependent functions that we use are of a similar form as those listed in (3.19) but without the factor $(1+(wt)^2)^2$. Using Maple® we find that the greatest common denominator of the entries of the resulting matrix is $(1+(wt)^2)^2$. We multiply with this denominator and obtain $\mathbf{M}(t)$. The degrees of the entries of this matrix polynomial in the case of a superposition of a translation and a rotation are given by the matrix

$$\begin{bmatrix} 6 & 6 & 6 & 4 \\ 6 & 6 & 4 & -\infty \\ 6 & 6 & 6 & -\infty \\ -\infty & 6 & 6 & 6 \end{bmatrix}. \tag{3.24}$$

Hence, the degree of $\det(\mathbf{M}(t))$ is 24. In the case of a pure rotation the degrees of the non-zero entries of $\mathbf{M}(t)$ are four. Thus, in this case the degree of the determinant is

16. If we have a pure translation, the degrees are given by the matrix

$$\begin{bmatrix} 2 & 2 & 2 & 0 \\ 2 & 2 & 0 & -\infty \\ 2 & 2 & 2 & -\infty \\ -\infty & 2 & 2 & 2 \end{bmatrix}, \tag{3.25}$$

and hence the degree of the determinant is eight.

Case 2: Now, we show how the candidates for the potential collision times can be computed if the surface $\mathcal{S}$ belongs to the stationary object $\mathcal{O}_1$ whereas $\mathcal{C}$ belongs to $\mathcal{O}_2$. Again, we distinguish the cases whether $\mathcal{C}$ is a straight line or not and whether $\mathcal{S}$ is a plane or not.

First, let $\mathcal{C}$ be a straight line and let $\mathcal{S}$ be not a plane. We apply a rotation and a translation to the scene such that $\mathcal{S}$ is given by the quadratic form defined by

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & 0 & a_1 \\ 0 & a_{22} & 0 & a_2 \\ 0 & 0 & a_{33} & a_3 \\ a_1 & a_2 & a_3 & a_0 \end{bmatrix}.$$

Let the parameterization of $\mathcal{C}$ be $\mathbf{x}(\lambda) = \mathbf{p}+\lambda\mathbf{r}$. We insert this parameterization into the quadratic form defining $\mathcal{S}$ and obtain a quadratic equation in λ whose roots correspond to the intersection points between $\mathcal{C}$ and $\mathcal{S}$. If there is a tangential intersection point then the discriminant $\Delta(\mathcal{S},\mathcal{C})$ of this polynomial is zero. In order to obtain a polynomial in the time parameter we replace $\mathbf{p}$ and $\mathbf{r}$ by the respective time dependent functions

$$\begin{aligned} \mathbf{p}(t) &= \mathbf{R}(t)(\tilde{\mathbf{p}} - \mathbf{c}) + \mathbf{c} + \mathbf{s}(t), \\ \mathbf{r}(t) &= \mathbf{R}(t)\tilde{\mathbf{r}} \end{aligned}$$

and multiply with the denominator $(1 + (wt)^2)^2$. With the help of Maple® we see that the degree of the resulting polynomial is two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition.

Now, let $\mathcal{C}$ be not a straight line and let $\mathcal{S}$ be a plane. By applying an appropriate rotation and translation to the scene we can assume that $\mathcal{S}$ is the (x_1, x_2)-plane. Let $\mathcal{C}$ lie in the plane $\mathcal{P}$ with the parameterization $\mathbf{x}(\alpha, \beta) = \mathbf{r} + \alpha\mathbf{p} + \beta\mathbf{q}$. The implicit form of $\mathcal{C}$ in (α, β)-coordinates is given by

$$\mathbf{A}_H = \begin{bmatrix} a_{11} & 0 & a_1 \\ 0 & a_{22} & a_2 \\ a_1 & a_2 & a_0 \end{bmatrix}.$$

By lemma 3.17, the tangential intersection points between $\mathcal{C}$ and $\mathcal{S}$ are exactly the tangential intersection points between $\mathcal{C}$ and the intersection line $\mathcal{D}$ between $\mathcal{S}$ and $\mathcal{P}$.

This line is implicitly given by the equation $r_3 + \alpha p_3 + \beta q_3 = 0$. We write this as a "quadratic" form defined by the matrix

$$\mathbf{B}_H = \begin{bmatrix} 0 & 0 & p_3 \\ 0 & 0 & q_3 \\ p_3 & q_3 & 2r_3 \end{bmatrix}.$$

By theorem 2.30, a necessary condition for $\mathcal{C}$ and $\mathcal{D}$ to have a non-singular intersection is that $\chi_{\mathbf{A},\mathbf{B}}$ has a multiple root. Thus, we define $\Delta(\mathcal{S},\mathcal{C})$ as the discriminant of $\chi_{\mathbf{A},\mathbf{B}}$. As in the case of a stationary conic and a moving plane, we see with the help of Maple® that $\Delta(\mathcal{S},\mathcal{C}) = a_{11}a_{22}\Delta'(\mathcal{S},\mathcal{C})$ for a polynomial Δ' in the coefficients of $\mathcal{S}$ and $\mathcal{C}$. Hence, the discriminant is zero if $\mathcal{C}$ is a parabola. So, let us first assume that $\mathcal{C}$ is not a parabola. We replace all occurrences of $\mathbf{r}, \mathbf{p}$ and $\mathbf{q}$ in $\Delta'(\mathcal{S},\mathcal{C})$ by the corresponding time dependent functions

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{R}(t)(\tilde{\mathbf{r}} - \mathbf{c}) + \mathbf{c} + \mathbf{s}(t), \\ \mathbf{p}(t) &= \mathbf{R}(t)\tilde{\mathbf{p}}, \\ \mathbf{q}(t) &= \mathbf{R}(t)\tilde{\mathbf{q}}, \end{aligned} \tag{3.26}$$

where $\mathbf{R}(t)$ and $\mathbf{s}(t)$ are the rotation matrix and the translation vector as defined before and $\mathbf{c}$ is the center of rotation. By multiplying with the denominator of the result we get a polynomial whose roots are the candidates for the potential collision times. With Maple® we find out that the degree of this polynomial is two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition. If $\mathcal{C}$ is a parabola then we can assume that it is given by the parameterization $x(\lambda) = \mathbf{r} + \lambda\mathbf{p} + a\lambda^2\mathbf{q}$. The parabola intersects the (x_1, x_2)-plane if and only if there is a λ such that the third component $x_3(\lambda) = r_3 + \lambda p_3 + a\lambda^2 q_3$ is zero. If there is a tangential intersection point, then the discriminant $\Delta(\mathcal{S},\mathcal{C}) = p_3^2 - 4ar_3q_3$ of this polynomial is zero. Again, we use (3.26) and multiply with the denominator to get a polynomial in t whose roots are the candidates for the potential collision times. This polynomial has degree one in the case of a pure translation, four in the case of a pure rotation and five in the case of a superposition.

Finally, we consider the case that $\mathcal{C}$ is not a straight line and $\mathcal{S}$ is not a plane. Let $\mathcal{S}$ be given by the (4×4)-matrix $\mathbf{A}_H$ and let $\mathcal{C}$ lie in the plane $\mathcal{P}$ parameterized as $x(\alpha, \beta) = \mathbf{r} + \alpha\mathbf{p} + \beta\mathbf{q}$. Let the implicit form of $\mathcal{C}$ in (α, β)-coordinates be given by the (3×3)-matrix

$$\begin{bmatrix} b_{11} & 0 & b_1 \\ 0 & b_{22} & b_2 \\ b_1 & b_2 & b_0 \end{bmatrix}.$$

Inserting the parameterization of $\mathcal{P}$ into the quadratic form of $\mathcal{S}$ we obtain an implicit equation for the intersection curve $\mathcal{D}$ between $\mathcal{P}$ and $\mathcal{S}$ in (α, β)-coordinates. Let $\mathbf{C}_H$ be the (3×3)-matrix corresponding to this quadratic form. We define $\Delta(\mathcal{S},\mathcal{C})$ as the discriminant of $\chi_{\mathbf{B},\mathbf{C}}$. If we replace all occurrences of $\mathbf{A}_H$ in (3.23) by $\mathbf{B}_H$ and call the resulting matrix $\mathbf{M}$, then $\Delta(\mathcal{S},\mathcal{C}) = \det \mathbf{M}$. By lemma 3.19, a necessary polynomial condition for $\mathcal{S}$ and $\mathcal{C}$ to have a tangential intersection is $\Delta(\mathcal{S},\mathcal{C}) = 0$. In order to get a matrix polynomial in the time parameter we must consider the plane $\mathcal{P}$ to be time dependent. The point $\mathbf{r}$ and the vectors $\mathbf{p}$ and $\mathbf{q}$ at time t are given by the equations (3.26). We replace all occurrences of the components of $\mathbf{r}, \mathbf{p}$ and $\mathbf{q}$ in $\mathbf{M}$ using these relationships. The entries of the resulting matrix are too complicated for Maple® to express them symbolically. Thus, we assume that at the beginning of the time interval the plane $\mathcal{P}$ is the (x_1, x_2)-plane. This can be achieved by applying an appropriate translation and rotation to the scene in advance. This is the reason why we did not assume the surface $\mathcal{S}$ to be in standard form. With this assumption we have $\tilde{\mathbf{r}} = \mathbf{0}, \tilde{\mathbf{p}} = [1, 0, 0]^T$ and $\tilde{\mathbf{q}} = [0, 1, 0]^T$. Now we find with the help of Maple® that the greatest common denominator of the entries of the resulting matrix after the substitution is again $(1 + (wt)^2)^2$. We multiply with this denominator and denote the result by $\mathbf{M}(t)$. In the case of a superposition of a rotation and a translation the degrees of the entries of $\mathbf{M}(t)$ are given by the matrix (3.24). Thus, the degree of $\det(\mathbf{M}(t))$ is again 24 in this case. In the case of a pure rotation the degrees are given by the matrix (3.25) and hence, the degree of the determinant is eight. If we have a pure rotation, then all non-zero entries of $\mathbf{M}(t)$ have degree four. Thus, the degree of the determinant is 16 in this case.

Now, that we have shown how to formulate the problem of finding candidates for the potential collision times as a polynomial equation in t, we must check for each such candidate t_0 whether there actually are tangential intersection points at that time. If this is the case we have to compute these points. Thus, let $\mathcal{C} = \mathcal{C}(t_0)$ and $\mathcal{S} = \mathcal{S}(t_0)$ for such a candidate, and let $\mathbf{x}(\lambda)$ be a parameterization of $\mathcal{C}$. Inserting $\mathbf{x}(\lambda)$ into the implicit form defining $\mathcal{S}$, we get a polynomial $f(\lambda)$ whose roots correspond to the intersection points between $\mathcal{C}$ and $\mathcal{S}$. Since we are interested in tangential intersection points, we have to look for multiple roots of f. Thus, we determine the roots λ_0 of the derivative f' of f with respect to λ and check whether $f(\lambda_0) = 0$. In this way we obtain all tangential intersection points. The degree of the polynomial f' is one in the case of $\mathcal{C}$ being a straight line or $\mathcal{S}$ being a plane and three in the case of $\mathcal{C}$ and $\mathcal{S}$ being both non-linear. Hence, the maximal degree of the polynomial equations that have to be solved to find the potential collision times and the witness points is always dominated by the degree of the polynomial whose roots are the candidates for the potential collision times. We summarize the results of this section in the following theorem.

Theorem 3.20. *Finding the potential collision times between a conic $\mathcal{C}$ and a quadric $\mathcal{S}$ can be reduced to finding the roots of polynomials whose degrees are upper-bounded by the following table.*

	translation	*rotation*	*superposition*
$\mathcal{C}$ is a parabola and $\mathcal{S}$ is a plane	1	4	5
$\mathcal{C}$ is an ellipse or a hyperbola and $\mathcal{S}$ is a plane	2	4	6
$\mathcal{C}$ is a straight line and $\mathcal{S}$ is a general quadric	2	4	6
$\mathcal{C}$ is not a straight line and $\mathcal{S}$ is a general quadric	8	16	24

Face-Face Test

Now, we show how to compute the potential collision times for two faces $\mathcal{F}_1$ and $\mathcal{F}_2$ of $\mathcal{O}_1$ and $\mathcal{O}_2$. We denote the surfaces containing these faces by $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. We distinguish the cases whether one of the surfaces is a plane or not.

Case 1: $\mathcal{S}_1$ or $\mathcal{S}_2$ is a plane. Let this plane be given by the implicit form $\mathbf{x}^T\mathbf{n} = n_0$. We write this as a quadratic form defined by the matrix

$$\mathbf{N}_H = \begin{bmatrix} 0 & 0 & 0 & n_1 \\ 0 & 0 & 0 & n_2 \\ 0 & 0 & 0 & n_3 \\ n_1 & n_2 & n_3 & -2n_0 \end{bmatrix}.$$

Let the quadric be given by the quadratic form defined by the matrix $\mathbf{A}_H$. We can assume that this quadric is not ruled. The reason for this is the following. If a plane touches a ruled quadric in a point $\mathbf{p}$, then it contains the whole generating straight line $\mathcal{L}$ through that point. If $\mathcal{L}$ is a tangential intersection line and there is a collision between $\mathcal{F}_1$ and $\mathcal{F}_2$ in $\mathbf{p}$, then also the boundaries of these faces are involved in the collision. Hence, this case is covered by the before mentioned collision tests. If $\mathcal{L}$ is not a tangential intersection line and the point $\mathbf{p}$ lies in both $\mathcal{F}_1$ and $\mathcal{F}_2$, then there must be a penetration between these faces. Thus, the interiors of the objects intersect and we know that there must have been a collision at some earlier point in time.

If there is a tangential intersection point between the plane and the quadric, then by theorem 2.24, the polynomial $\chi_{A,N}$ has either a multiple root or is identically zero. By expanding $\chi_{A,N}(\lambda) = \det(\mathbf{A}_H + \lambda\mathbf{N}_H)$ by cofactors of the last column, we see that the degree of this polynomial in λ is two. Since we are looking for the earliest point in time when the objects get in touch, we are only interested in points, where the surfaces touch externally. By corollary 2.29, a necessary condition for this is that $\chi_{A,N}$ has either a *positive* multiple root or is identically zero. We define the function $\Delta(\mathcal{S}_1, \mathcal{S}_2)$ as the discriminant of $\chi_{A,N}$. Using Maple®, we find that it holds $\Delta(\mathcal{S}_1, \mathcal{S}_2) = \det(\mathbf{A}) \cdot \Delta'(\mathcal{S}_1, \mathcal{S}_2)$ for a polynomial Δ' in the coefficients defining the surfaces. Hence, the discriminant vanishes if $\mathbf{A}_H$ defines a non-central surface. Let us first assume that $\mathbf{A}_H$ defines a central surface. Since we did not make any assumptions on the position and orientation of the surfaces, we may assume that the coordinate system has been transformed in such a way that the angular velocity $\boldsymbol{\omega}$ is parallel to the x_1-axis and the center of mass $\tilde{\mathbf{c}}$ of the moving object equals the origin. We replace all occurrences of the coefficients defining the moving surface in Δ' by the respective time depending functions using the relationships (3.19) or (3.20), respectively, but without the multiplication with the denominator. Instead, we multiply with the denominator $(1+(wt)^2)^2$ after the substitution and obtain a polynomial whose roots are the candidates for the potential collision times. The degree of this polynomial is two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition.

Now, let $\mathbf{A}_H$ define a non-central surface. Since we do not have to consider ruled surfaces, we can assume that $\mathbf{A}_H$ defines an elliptic paraboloid. We first look at the case that $\mathcal{S}_2$ is the plane, i.e. the plane is the moving surface. We rotate and translate the coordinate system in such a way that $\mathbf{A}_H$ has the form shown in table 2.1. We replace the symbols a, b and c in the implicit form given in that table by $1/a^2, 1/b^2$ and c^2, respectively. Then, the paraboloid can be parameterized as

$$x(\lambda,\mu) = \begin{bmatrix} \lambda ac(1-\mu^2) \\ 2\lambda bc\mu \\ (1+\mu^2)\lambda^2/2 \\ 1+\mu^2 \end{bmatrix}.$$

We insert this into the implicit form of the plane and obtain a quadratic polynomial $f(\lambda,\mu)$ whose roots correspond to the intersection points between $\mathcal{S}_1$ and $\mathcal{S}_2$. If there is a tangential intersection point, then there are values λ_0 and μ_0 for which the partial derivatives $\partial f/\partial\lambda$ and $\partial f/\partial\mu$ vanish, as well. Hence, we need a necessary condition for these three polynomials to vanish simultaneously. We do this by computing a Groebner basis using graded lexicographic order with the computer algebra system Singular®. This basis consists of the only element

$$\begin{aligned} \Delta(\mathcal{S}_1,\mathcal{S}_2) &= cn_0(a^2n_1^2+b^2n_2^2)\cdot\Delta'(\mathcal{S}_1,\mathcal{S}_2) \quad \text{with} \\ \Delta'(\mathcal{S}_1,\mathcal{S}_2) &= a^2c^2n_1^2-2n_0n_3+b^2c^2n_2^2. \end{aligned}$$

If there is a tangential intersection point between the plane and the paraboloid and $n_0 = 0$, then this point must be the apex of the paraboloid. The normal in this apex is parallel to the x_3-axis and hence $n_1 = n_2 = 0$. But then, $\Delta'(\mathcal{S}_1,\mathcal{S}_2) = 0$. Similarly, if there is a tangential intersection point and $a^2n_1^2+b^2n_2^2 = 0$, then the normal in this point must be parallel to the x_3-axis. The only point on the paraboloid for which this is the case is the apex, and hence $n_0 = 0$, which again implies $\Delta'(\mathcal{S}_1,\mathcal{S}_2) = 0$. This means, whenever there is a tangential intersection between the two surfaces, then the polynomial Δ' vanishes. In order to get a polynomial whose roots are the candidates for the potential collision times, we replace $\mathbf{n}$ and n_0 in $\Delta'(\mathcal{S}_1,\mathcal{S}_2)$ by the corresponding time dependent functions and multiply with the denominator $(1+(wt)^2)^2$. This polynomial has degree one in the case of a pure translation, four in the case of a pure rotation and five in the case of a superposition.

Let us now assume that the plane belongs to object $\mathcal{O}_1$ and is therefore stationary. Then we apply a rotation and a translation to the scene such that $\mathbf{n} = [0,0,1]^T$ and $n_0 = 0$, i.e. $\mathcal{S}_1$ is the (x_1, x_2)-plane. The elliptic paraboloid can be parameterized as

$$\mathbf{x}(\lambda, \mu) = \mathbf{r} + \lambda ac\frac{1-\mu^2}{1+\mu^2}\mathbf{p} + \lambda bc\frac{2\mu}{1+\mu^2}\mathbf{q} - \frac{\lambda^2}{2}\mathbf{p} \times \mathbf{q}$$

for a point $\mathbf{r}$ and unit vectors $\mathbf{p}$ and $\mathbf{q}$ with $\mathbf{p}^T\mathbf{q} = 0$. The point $\mathbf{r}$ is the apex of the paraboloid and $\mathbf{p} \times \mathbf{q}$ is parallel to the normal in $\mathbf{r}$. We insert this parameterization into the implicit form of the (x_1, x_2)-plane which yields $x_3(\lambda, \mu) = 0$. Multiplying this with $1 + \mu^2$ we obtain a quadratic polynomial in λ and μ whose roots correspond to the intersection points between $\mathcal{S}_1$ and $\mathcal{S}_2$. Again, we need to find a criterion for this polynomial and its partial derivatives to have a common root. As before, we use Singular® to compute a Groebner basis for these three polynomials. The only element of this basis is

$$\begin{aligned}\Delta(\mathcal{S}_1, \mathcal{S}_2) &= cr_3(a^2p_3^2 + b^2q_3^2) \cdot \Delta'(\mathcal{S}_1, \mathcal{S}_2) \quad \text{with} \\ \Delta'(\mathcal{S}_1, \mathcal{S}_2) &= a^2c^2p_3^2 + 2r_3(p_1q_2 - p_2q_1) + b^2c^2q_3^2.\end{aligned}$$

If there is a tangential intersection point between $\mathcal{S}_1$ and $\mathcal{S}_2$ and $r_3 = 0$, then this intersection point must be the apex. In this case, the normal $\mathbf{p} \times \mathbf{q}$ must be parallel to $[0,0,1]^T$. Hence, $p_3 = q_3 = 0$, which implies $\Delta'(\mathcal{S}_1, \mathcal{S}_2) = 0$. Similarly, if there is a tangential intersection point and $a^2p_3^2 + b^2q_3^2 = 0$, then $p_3 = q_3 = 0$ and hence, $\mathbf{p} \times \mathbf{q}$ is parallel to $[0,0,1]^T$. The only point of the paraboloid whose normal points in this direction is the apex. Thus, $\mathbf{r}$ must be the tangential intersection point which implies $r_3 = 0$. But then, $\Delta'(\mathcal{S}_1, \mathcal{S}_2) = 0$. Thus, a necessary condition for $\mathcal{S}_1$ and $\mathcal{S}_2$ to intersect tangentially is $\Delta'(\mathcal{S}_1, \mathcal{S}_2) = 0$. In order to obtain a polynomial whose roots are the candidates for the potential intersection points we replace $\mathbf{r}, \mathbf{p}$ and $\mathbf{q}$ by the corresponding time dependent function which have the form (3.26). After multiplying with the denominator $(1 + (wt)^2)^2$ we get a polynomial of degree one in the case of a pure translation, four in the case of a pure rotation and five in the case of a superposition.

Case 2: Neither $\mathcal{S}_1$ nor $\mathcal{S}_2$ is a plane. Let the quadratic forms of these surfaces be defined by the matrices $\mathbf{A}_H$ and $\mathbf{B}_H$, respectively. Again, we are interested in the points where $\mathcal{S}_1$ and $\mathcal{S}_2$ touch externally. Because of corollary 2.29, a necessary condition for this is that $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ has either a positive multiple root or is identically zero. This polynomial has degree four in λ. We define $\Delta(\mathcal{S}_1, \mathcal{S}_2)$ as the discriminant of $\chi_{\mathbf{A},\mathbf{B}}$. The discriminant of a polynomial $f(\lambda)$ of degree k with leading coefficient L is defined as

$$\frac{1}{L}\text{res}_\lambda\left(f, \frac{df}{d\lambda}\right) = \frac{1}{L}\left|\mathbf{Syl}_\lambda\left(f, \frac{df}{d\lambda}\right)\right|.$$

This Sylvester matrix has dimension $2k - 1$. Its first column has L as first entry, kL as kth entry and has zeros in all remaining places. Hence, we can eliminate the kL in the first column with a row operation. If we define $\mathbf{M}$ as the $(1,1)$-cofactor of the resulting matrix, then the discriminant of $f(\lambda)$ is given by the determinant of $\mathbf{M}$. The dimension of $\mathbf{M}$ is $2(k-1)$. Thus, the discriminant $\Delta(\mathcal{S}_1,\mathcal{S}_2)$ can be computed as the determinant of a (6×6)-matrix. We call this matrix $\mathbf{M}(\mathcal{S}_1,\mathcal{S}_2)$.

We did not make any assumption on the position and orientation of the surfaces. Hence, we may assume that the coordinate system has been transformed in such a way that the angular velocity $\boldsymbol{\omega}$ is parallel to the x_1-axis and the center of mass $\tilde{\mathbf{c}}$ of $\mathcal{O}_2$ equals the origin. In order to get a polynomial whose roots are the candidates for the potential collision times, we replace all occurrences of the coefficients defining the moving surface $\mathcal{S}_2$ in the matrix $\mathbf{M}(\mathcal{S}_1,\mathcal{S}_2)$ by the corresponding time dependent functions. For this substitution we use the relationships (3.19) but without the multiplication with the denominator. Instead, we multiply each column of the matrix after the substitution with the least common denominator of its entries. Due to our assumption on the angular velocity and the center of mass, this computation can be done symbolically using Maple®. If we have a superposition of a translation and a rotation, the degrees of the entries of the resulting matrix are given by

$$\begin{bmatrix} 4 & 6 & 6 & 6 & 4 & -\infty \\ -\infty & 4 & 6 & 6 & 6 & 4 \\ 6 & 6 & 6 & 4 & -\infty & -\infty \\ 4 & 6 & 6 & 6 & -\infty & -\infty \\ -\infty & 4 & 6 & 6 & 6 & -\infty \\ -\infty & -\infty & 4 & 6 & 6 & 6 \end{bmatrix}, \tag{3.27}$$

Thus, the degree of the polynomial whose roots are the candidates for the potential collision times is at most 36. Similarly, we find that in the case of a pure rotation this degree is 24 and for a pure translation we get a degree of at most twelve.

Now we must decide for each candidate t_0 for a potential collision time whether there are tangential intersection points at that time and if so, we must compute these points. Thus Let $\mathcal{S}_1=\mathcal{S}_1(t_0)$ and $\mathcal{S}_2=\mathcal{S}_2(t_0)$. We distinguish the cases that one of the surfaces is a plane or not.

Case 1: Let one of the surfaces be a plane. W.l.o.g. we assume that this is $\mathcal{S}_1$. Let $\mathbf{x}(\lambda,\mu)=\mathbf{r}+\lambda\mathbf{p}+\mu\mathbf{q}$ be a parameterization of $\mathcal{S}_1$. By applying an appropriate rotation and translation to the scene we can assume that the matrix $\mathbf{A}_H$ that defines $\mathcal{S}_2$ is of the form

$$\begin{bmatrix} a_{11} & 0 & 0 & a_1 \\ 0 & a_{22} & 0 & a_2 \\ 0 & 0 & a_{33} & a_3 \\ a_1 & a_2 & a_3 & a_0 \end{bmatrix}.$$

We insert $\mathbf{x}(\lambda,\mu)$ into this quadratic form and obtain an implicit quadratic form for the intersection curve $\mathcal{C}$ between $\mathcal{S}_1$ and $\mathcal{S}_2$ in (λ,μ)-coordinates. We write this implicit form as

$$[\lambda, \mu]\mathbf{B}\begin{bmatrix} \lambda \\ \mu \end{bmatrix} + 2[\lambda, \mu]\mathbf{b} + b_0.$$

The tangential intersection points between $\mathcal{S}_1$ and $\mathcal{S}_2$ are exactly the singular points on this curve. We distinguish the cases that $\mathcal{S}_2$ is a central or a non-central surface. If it is a non-central surface, then we have seen that it must be an elliptic paraboloid. Then, w.l.o.g. $a_{33} = a_1 = a_2 = 0$ and $a_{11}, a_{22}, a_3 \neq 0$. In this case, the determinant of $\mathbf{B}$ evaluates to $\det \mathbf{B} = a_{11}a_{22}(p_2q_1 - p_1q_2)$. If this determinant is non-zero and $\mathcal{C}$ has a singular point, then this point has the coordinates $[\lambda_0, \mu_0]^T = -\mathbf{B}^{-1}\mathbf{b}$. We insert these coordinates into the implicit form of $\mathcal{C}$ to check whether the point lies on the curve. If this is the case then t_0 is a potential collision time and $\mathbf{x}(\lambda_0, \mu_0)$ is the tangential intersection point. If $\det \mathbf{B} = 0$, then the normal $\mathbf{n} = \mathbf{p} \times \mathbf{q}$ of $\mathcal{S}_1$ lies in the (x_1, x_2)-plane. But as all normals of the paraboloid are of the form $\mathbf{A}\mathbf{x} + \mathbf{a} = [a_{11}x_1, a_{22}x_2, a_3]^T$ and $a_3 \neq 0$, there can not be any tangential intersections in this case.

If $\mathcal{S}_2$ is a central surface, then it must be an ellipsoid or a two-sheet hyperboloid. Then, $a_1 = a_2 = a_3 = 0$ and we have

$$\begin{aligned} \det \mathbf{B} &= \begin{vmatrix} \mathbf{p}^T\mathbf{A}\mathbf{p} & \mathbf{p}^T\mathbf{A}\mathbf{q} \\ \mathbf{p}^T\mathbf{A}\mathbf{q} & \mathbf{q}^T\mathbf{A}\mathbf{q} \end{vmatrix} \\ &= \mathbf{p}^T\mathbf{A}\mathbf{p} \cdot \mathbf{q}^T\mathbf{A}\mathbf{q} - (\mathbf{p}^T\mathbf{A}\mathbf{q})^2 \\ &= \mathbf{n}^T\operatorname{adj}(\mathbf{A})\mathbf{n} \end{aligned}$$

with $\mathbf{n} = \mathbf{p} \times \mathbf{q}$ being the normal of $\mathcal{S}_1$. The last equality can easily be verified component-wise. If this determinant is non-zero and if $\mathcal{C}$ contains a singular point, then the coordinates of this point must again be $[\lambda_0, \mu_0]^T = -\mathbf{B}^{-1}\mathbf{b}$. By inserting these coordinates into the implicit form of $\mathcal{C}$ we check whether this point lies on both $\mathcal{S}_1$ and $\mathcal{S}_2$. If this is the case, then t_0 is a potential collision time and $\mathbf{x}(\lambda_0, \mu_0)$ is the tangential intersection point. If $\det \mathbf{B} = 0$, then $\mathcal{S}_2$ must be a two-sheet hyperboloid. W.l.o.g. we assume that $a_{11} = 1/\alpha^2, a_{22} = -1/\beta^2, a_{33} = -1/\gamma^2$ and $a_0 = -1$. A parameterization of this surface is

$$\mathbf{y}(u, v) = \begin{bmatrix} \alpha \cosh u \cosh v \\ \beta \sinh u \cosh v \\ \gamma \sinh v \end{bmatrix}.$$

The normal of $\mathcal{S}_2$ in the point $\mathbf{y}(u, v)$ can be computed as

$$\mathbf{m}(u, v) = \left(\frac{\partial \mathbf{y}}{\partial u}(u, v) \times \frac{\partial \mathbf{y}}{\partial v}(u, v)\right) / \cosh v = \begin{bmatrix} \beta\gamma \cosh u \cosh v \\ -\alpha\gamma \sinh u \cosh v \\ -\alpha\beta \sinh v \end{bmatrix}.$$

We easily verify that $\mathbf{m}(u, v)^T\operatorname{adj}(\mathbf{A})\mathbf{m}(u, v) = 1$ for all u, v. Thus, in the case $\det \mathbf{B} = 0$ there is no tangential intersection between $\mathcal{S}_1$ and $\mathcal{S}_2$.

Case 2: Neither $\mathcal{S}_1$ nor $\mathcal{S}_2$ is a plane. As usual, we denote the matrices defining $\mathcal{S}_1$ and $\mathcal{S}_2$ by $\mathbf{A}_H$ and $\mathbf{B}_H$, respectively. By our choice of t_0, the discriminant of $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ is zero.

Thus, this polynomial either has a multiple root or is constantly zero. We first consider the case that $\chi_{A,B}$ is not constantly zero. In this case we compute all positive multiple roots. If there is no such root, then by corollary 2.29 there is no tangential intersection point. Otherwise, let $\lambda_0 > 0$ be a multiple root of $\chi_{A,B}$. Then, by theorem 2.24 we have to look for points $\mathbf{p}_H$ with $\mathbf{p}_H^T \mathbf{A}_H \mathbf{p}_H = 0$ and $\mathbf{C}_H \mathbf{p}_H = \mathbf{0}$, where we set $\mathbf{C}_H = \mathbf{A}_H + \lambda_0 \mathbf{B}_H$. If the determinant $\det \mathbf{C}$ of the upper left (3×3)-submatrix of $\mathbf{C}_H$ is non-zero, then the only possibility for such a point is $\mathbf{p} = -\mathbf{C}^{-1}\mathbf{c}$. We check whether this point lies on both $\mathcal{S}_1$ and $\mathcal{S}_2$. If this is the case, then t_0 is a potential collision time and $\mathbf{p}$ is the tangential intersection point. If $\det \mathbf{C} = 0$, then we must check what surface is defined by the matrix $\mathbf{C}_H$. If this surface is a cylinder or two parallel planes, then it does not contain any singular points, i.e. points fulfilling $\mathbf{C}_H \mathbf{p}_H = \mathbf{0}$. Hence, t_0 is not a potential collision time in that case. If $\mathbf{C}_H$ defines a double plane, then we determine a linear implicit form $\mathbf{n}^T \mathbf{x} = n_0$ of this plane and intersect it with $\mathcal{S}_1$. If this intersection is not empty, then t_0 is a potential collision time and the intersection points are tangential intersection points. If $\mathbf{C}_H$ defines two intersecting planes, then we determine a parameterization of their intersection line and intersect this line with $\mathcal{S}_1$. In this way we obtain the tangential intersection points. Finally, if $\mathbf{C}_H$ defines a straight line, then we intersect this line with $\mathcal{S}_1$ in order to find the tangential intersection points.

Now, let $\chi_{A,B}$ be constantly zero. By [FNO89] this means, that all quadrics in the pencil $\mathcal{Q}_{\mathbf{A},\mathbf{B}}$ are (projective) cones and that the intersection between $\mathcal{S}_1$ and $\mathcal{S}_2$ consists of a conic $\mathcal{C}$ and a double line $\mathcal{L}$. We do not have to consider the tangential intersection points forming $\mathcal{L}$ because if some of them lie in $\mathcal{F}_1$ and $\mathcal{F}_2$ there must be a collision involving the boundaries of these faces. Hence, this case is covered by the other collision detection tests. All non-singular points on $\mathcal{C}$ are intersection points between $\mathcal{S}_1$ and $\mathcal{S}_2$ which are not tangential. If $\mathcal{C}$ is a double line, we can ignore it by the same argument we used to ignore $\mathcal{L}$. So let us assume that $\mathcal{C}$ has a singular point $\mathbf{p}$ and is not a double line. Then it must be a pair of lines intersecting in $\mathbf{p}$. This means that $\mathbf{p}$ is the apex of both (projective) cones $\mathcal{S}_1$ and $\mathcal{S}_2$. If $\mathbf{p}$ is not a point at infinity, $\mathcal{S}_1$ and $\mathcal{S}_2$ must be two cones with common apex. Since we do not consider faces containing singular points, we can ignore this case, as well. Thus, t_0 is not a potential collision time in the case that $\chi_{A,B}$ vanishes identically.

We have seen that the degrees of the polynomials that have to be solved to find the potential collision times and the witness points are always dominated by the degrees of the polynomials whose roots are the candidates for the potential collision times. We summarize the results of this section in the following theorem.

Theorem 3.21. *Finding the potential collision times between two quadrics $\mathcal{S}_1$ and $\mathcal{S}_2$ can be reduced to finding the roots of polynomials whose degrees are upper-bounded by the following table.*

	translation	*rotation*	*superposition*
plane vs. central quadric	2	4	6
plane vs. elliptic paraboloid	1	4	5
none of the quadrics is a plane	12	24	36

The Penetration Test

The penetration test checks whether a point $\mathbf{p}$ that lies on both $\partial\mathcal{O}_1$ and $\partial\mathcal{O}_2(t_0)$ witnesses a collision at the potential collision time t_0. Unfortunately, we are not able to present an efficient method to solve this problem in all cases. Therefore, we show for the different collision types how to perform this task in non-degenerate situations. For the case that there are degeneracies, i.e. if the approaches that we present are not applicable, we show afterwards exemplarily for a potential collision between two edges that the penetration test can be formulated as the problem to decide whether a cell in a semi-algebraic set is full-dimensional.

Vertex-Face We only consider the case that the vertex $\mathbf{p}$ belongs to the moving object $\mathcal{O}_2(t)$. The converse case works analogously. We write the time dependent vertex as

$$\mathbf{p}(t) = \mathbf{R}(t)(\tilde{\mathbf{p}} - \tilde{\mathbf{c}}) + \tilde{\mathbf{c}} + \mathbf{s}(t),$$

At time t_0, there is a potential collision between $\mathbf{p}$ and a face $\mathcal{F}$ of object $\mathcal{O}_1$. Let $\mathcal{F}$ be embedded in the surface with implicit form $S(\mathbf{x}) = 0$. We define the function $\delta(t) = \left(1 + (wt)^2\right)^k S(\mathbf{p}(t))$, where $k = 1$ in the case that S is linear and $k = 2$ if S is quadratic. The reason for the multiplication with $\left((1 + (wt)^2)\right)^k$ is just to ensure that $\delta(t)$ is a polynomial. Since this factor is positive for all real values of t this multiplication has no influence on the sign of $\delta(t)$. Obviously, $\delta(t_0) = 0$. If $\delta(t)$ is constantly zero, then we set $\Delta = 0$. Otherwise, we define Δ as the first non-zero derivative of δ at t_0. We observe, that the following holds.

- If $\Delta > 0$, then $\mathbf{p}$ does not witness a collision at time t_0.
- If $\Delta < 0$ and $\mathbf{p}(t_0)$ does not lie on an edge of $\mathcal{F}$, then $\mathbf{p}$ witnesses a collision at time t_0.

In the remaining cases, i.e. if $\Delta = 0$ or $\Delta < 0$ and $\mathbf{p}(t_0)$ lies on an edge of $\mathcal{F}$, we cannot make a decision. The reason why we cannot make a decision if $\mathbf{p}$ hits an edge of $\mathcal{F}$ is illustrated in figure 3.12. In both pictures the vertex hits an edge of the face, and in both cases it penetrates the surface containing the face. But on the left-hand side, the vertex does not witness a collision, whereas on the right-hand side it does. However,

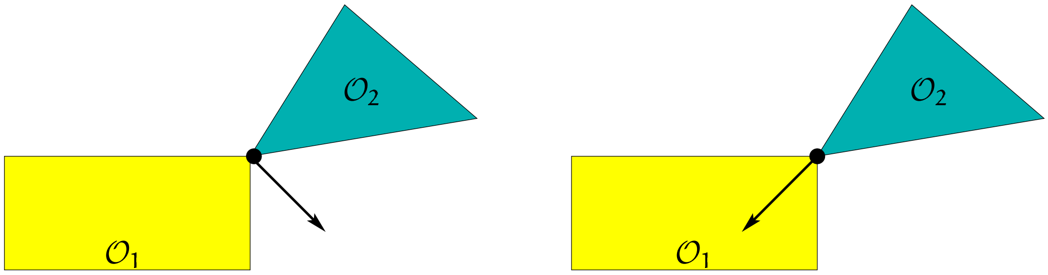


Figure 3.12: Special case in the penetration test for a vertex and a face. The vertex hits an edge of the face. The arrows show the current directions of motion of the vertex. In the left-hand picture, there is no penetration, whereas in the right-hand picture there is.

in many cases we can decide whether or not there is a collision in such a situation by using the approach which we will describe in the edge-edge penetration test in the next paragraph. We will come back to this at the end of that paragraph.

Figure 3.13 illustrates why we cannot make a decision when $\Delta = 0$. In both pictures the vertex stays on the surface containing the face. But on the left-hand side, there is no penetration. On the right-hand side however, every neighbourhood of the vertex contains a point of $\mathcal{O}_1$ that penetrates $\mathcal{F}$ as t increases.

Edge-Edge Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be the two edges that are in contact at time t_0, and let $\mathcal{C}_1$ and $\mathcal{C}_2$ be the two curves containing them. If there is a collision between $\mathcal{E}_1$ and $\mathcal{E}_2$, then $\mathcal{E}_2$ must penetrate one of the two faces that are adjacent to $\mathcal{E}_1$ immediately after the collision time t_0. We check this for one of these faces and if we do not find a penetration, we repeat the test for the other face. Let $\mathcal{F}$ be one of these faces and let the surface $\mathcal{S}$ containing $\mathcal{F}$ be given by the implicit form $S(\mathbf{x}) = 0$. We denote the parameterizations of $\mathcal{C}_1$ and $\mathcal{C}_2$ by $\mathbf{x}_1(\alpha)$ and $\mathbf{x}_2(\beta, t)$, respectively. Let the potential collision point be $\mathbf{p}$. We associate $\mathbf{p}$ with the object $\mathcal{O}_2$ and define α_0 and β_0 to be the curve parameters such that $\mathbf{p}(t_0) = \mathbf{x}_1(\alpha_0) = \mathbf{x}_2(\beta_0, t_0)$. Suppose we have a parameterization $\mathbf{f}(\alpha, \gamma)$ of $\mathcal{S}$ locally around the curve $\mathcal{C}_1$ such that $\mathbf{f}(\alpha, 0) = \mathbf{x}_1(\alpha)$ and such that for increasing (decreasing) values of γ we move towards the interior (exterior) of $\mathcal{F}$. Moreover, suppose that in a neighbourhood $\mathcal{I}$ of t_0 the curve $\mathcal{C}_2$ intersects the surface $\mathcal{S}$ in $\mathbf{x}_2(\beta(t), t)$. Then, there are functions $\alpha(t)$ and $\gamma(t)$ such that for $t \in \mathcal{I}$ it holds that

$$\mathbf{x}_2\big(\beta(t), t\big) = \mathbf{f}\big(\alpha(t), \gamma(t)\big).$$

Obviously, $\alpha(t_0) = \alpha_0$, $\beta(t_0) = \beta_0$ and $\gamma(t_0) = 0$. If we furthermore could compute the derivation of $\gamma(t)$ at t_0, we could make the following decision.

- If $\dot\gamma(t_0) > 0$, then $\mathbf{p}(t_0)$ witnesses a collision at time t_0.
- If $\dot\gamma(t_0) < 0$, then $\mathbf{p}(t_0)$ does not witness a collision at time t_0.

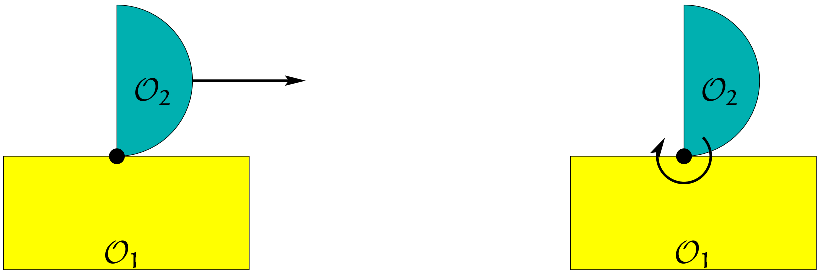


Figure 3.13: Another degenerate case in the vertex-face penetration test. The arrows indicate the motions of $\mathcal{O}_2$. On the left-hand side, $\mathcal{O}_2$ makes a translation such that the vertex does not penetrate the face. On the right-hand side $\mathcal{O}_2$ rotates around the vertex.

In the case $\dot{\gamma}(t_0) = 0$ we could not make a decision and therefore, we would consider this case as degenerate.

Let us define the tangents $\mathbf{u} = \frac{d}{d\alpha}\mathbf{x}_1(\alpha_0)$ and $\mathbf{w} = \frac{\partial}{\partial\beta}\mathbf{x}_2(\beta_0, t_0)$. We set $\boldsymbol{\varphi}(t) = \mathbf{x}_2\big(\beta(t), t\big) - \mathbf{f}\big(\alpha(t), \gamma(t)\big) \equiv \mathbf{0}$. Then, also for the derivative it holds $\dot{\boldsymbol{\varphi}}(t) \equiv \mathbf{0}$. Denoting the partial derivatives of $\mathbf{f}$ with respect to α and γ by $\mathbf{f}_\alpha$ and $\mathbf{f}_\gamma$, respectively, we get

$$\dot{\boldsymbol{\varphi}}(t_0) = \dot{\beta}(t_0)\mathbf{w} + \dot{\mathbf{p}}(t_0) - \dot{\alpha}(t_0)\mathbf{f}_\alpha(\alpha_0, 0) - \dot{\gamma}(t_0)\mathbf{f}_\gamma(\alpha_0, 0) = \mathbf{0}. \tag{3.28}$$

This is a system of linear equations in the unknowns $\dot{\alpha}(t_0), \dot{\beta}(t_0)$ and $\dot{\gamma}(t_0)$. If this system is regular we can use Cramer's rule to compute $\dot{\gamma}(t_0)$.

In order to apply this idea we need the function $\mathbf{f}(\alpha, \gamma)$ or at least its partial derivatives in the point $(\alpha_0, 0)$. Let $\mathbf{n}(\mathbf{x}) = \nabla S(\mathbf{x})$ be the normal of the surface $\mathcal{S}$ in the point $\mathbf{x}$. For the normal $\mathbf{n}\big(\mathbf{p}(t_0)\big)$ in the potential collision point we just write $\mathbf{n}$. We construct $\mathbf{f}$ as follows. For each value of α we consider the plane $\mathcal{P}_\alpha$ through $\mathbf{x}_1(\alpha)$ with normal $\mathbf{t}(\alpha) = \frac{d}{d\alpha}\mathbf{x}_1(\alpha)$. We define the vector $\mathbf{m}(\alpha) = \mathbf{n}\big(\mathbf{x}_1(\alpha)\big) \times \mathbf{t}(\alpha)$. In the point $\mathbf{x}_1(\alpha)$ this vector points locally towards the interior of $\mathcal{F}$. We define $\mathbf{f}(\alpha, \gamma)$ in such a way that for each fixed value of α we have a parameterization of the intersection curve between $\mathcal{P}_\alpha$ and $\mathcal{S}$. The plane $\mathcal{P}_\alpha$ is spanned by the vectors $\mathbf{n}\big(\mathbf{x}_1(\alpha)\big)$ and $\mathbf{m}(\alpha)$. We construct the point $\mathbf{f}(\alpha, \gamma)$ by starting at $\mathbf{x}_1(\alpha)$, then adding γ times the vector $\mathbf{m}(\alpha)$ and then adding some multiple of the vector $\mathbf{n}\big(\mathbf{x}_1(\alpha)\big)$ to reach the surface $\mathcal{S}$ again. Hence, we have

$$\mathbf{f}(\alpha, \gamma) = \mathbf{x}_1(\alpha) + \gamma\mathbf{m}(\alpha) + h(\alpha, \gamma)\mathbf{n}\big(\mathbf{x}_1(\alpha)\big).$$

Obviously, $h(\alpha, 0) = 0$ for all α. In this way, the desired condition $\mathbf{f}(\alpha, 0) = \mathbf{x}_1(\alpha)$ is fulfilled. Furthermore, our choice of $\mathbf{m}(\alpha)$ ensures that for increasing (decreasing) values of γ we move towards the interior (exterior) of $\mathcal{F}$. For the above described idea we need the partial derivatives of $\mathbf{f}(\alpha, \gamma)$ in the point $(\alpha_0, 0)$. Derivating, we obtain

$$\begin{aligned}
\mathbf{f}_\alpha(\alpha, \gamma) &= \mathbf{t}(\alpha) + \gamma\frac{d}{d\alpha}\mathbf{m}(\alpha) + h_\alpha(\alpha, \gamma)\mathbf{n}\big(\mathbf{x}_1(\alpha)\big) + h(\alpha, \gamma)\frac{d}{d\alpha}\mathbf{n}\big(\mathbf{x}_1(\alpha)\big), \\
\mathbf{f}_\gamma(\alpha, \gamma) &= \mathbf{m}(\alpha) + h_\gamma(\alpha, \gamma)\mathbf{n}\big(\mathbf{x}_1(\alpha)\big).
\end{aligned}$$

Substituting $\alpha = \alpha_0$ and $\gamma = 0$ and using $\mathbf{t}(\alpha_0) = \mathbf{u}$ and $h(\alpha, 0) = 0$ we get

$$\begin{aligned} \mathbf{f}_\alpha(\alpha_0, 0) &= \mathbf{u} + h_\alpha(\alpha_0, 0)\mathbf{n} \quad \text{and} & (3.29) \\ \mathbf{f}_\gamma(\alpha_0, 0) &= \mathbf{n} \times \mathbf{u} + h_\gamma(\alpha_0, 0)\mathbf{n}. & (3.30) \end{aligned}$$

Thus, we need the partial derivatives of $h(\alpha, \gamma)$ in the point $(\alpha_0, 0)$. We compute these by looking at the function $S\big(\mathbf{f}(\alpha, \gamma)\big) \equiv 0$. The partial derivatives of this functions are identically zero, as well. Derivating with respect to α yields

$$\begin{aligned} \mathbf{n}\big(\mathbf{f}(\alpha, \gamma)\big)^T \mathbf{f}_\alpha(\alpha, \gamma) &\equiv 0 \quad \text{and} \\ \mathbf{n}\big(\mathbf{f}(\alpha, \gamma)\big)^T \mathbf{f}_\gamma(\alpha, \gamma) &\equiv 0. \end{aligned}$$

Substituting $\alpha = \alpha_0$ and $\gamma = 0$ and using (3.29) and (3.30) and $\mathbf{n}^T\mathbf{u} = 0$ we obtain

$$h_\alpha(\alpha_0, 0)\mathbf{n}^2 = 0 \quad \text{and} \quad h_\gamma(\alpha_0, 0)\mathbf{n}^2 = 0. \tag{3.31}$$

Since we do not consider faces with singularities, we can assume that $\mathbf{n} \neq \mathbf{0}$ and hence it follows $h_\alpha(\alpha_0, 0) = h_\gamma(\alpha_0, 0) = 0$. With this, the derivatives (3.29) and (3.30) become

$$\mathbf{f}_\alpha(\alpha_0, 0) = \mathbf{u} \quad \text{and} \quad \mathbf{f}_\gamma(\alpha_0, 0) = \mathbf{n} \times \mathbf{u}.$$

Inserting this into (3.28) and reordering we obtain

$$\dot\gamma(t_0)(\mathbf{n} \times \mathbf{u}) + \dot\alpha(t_0)\mathbf{u} - \dot\beta(t_0)\mathbf{w} = \dot{\mathbf{p}}(t_0). \tag{3.32}$$

We use Cramer's rule to solve this system of equations for $\dot\gamma(t_0)$, which yields

$$\dot\gamma(t_0) = \frac{\dot{\mathbf{p}}(t_0)^T(\mathbf{w} \times \mathbf{u})}{\mathbf{n}^T\mathbf{w} \cdot \mathbf{u}^2}. \tag{3.33}$$

In the linear case, i.e. if both edges are straight, all faces involved are planar, and the motion is a pure translation, γ corresponds to the signed euclidian distance between $\mathcal{E}_1$ and the intersection point between $\mathcal{E}_2$ and one of the faces adjacent to $\mathcal{E}_1$. It is not hard to show that this distance is given by

$$\frac{\big(\mathbf{p}(t) - \mathbf{p}(t_0)\big)^T(\mathbf{w} \times \mathbf{u})}{\mathbf{n}^T\mathbf{w} \cdot \mathbf{u}^2}.$$

Derivating this with respect to t results in equation (3.33). Hence, our approach is equivalent to replacing the curved parts involved by linear ones and assuming a pure translation.

What are the degenerate cases in this approach? We cannot decide whether $\mathbf{p}$ witnesses a collision if $\dot\gamma(t_0) = 0$. This happens, whenever the tangents of $\mathcal{E}_1$ and $\mathcal{E}_2(t_0)$ in $\mathbf{p}$ and the current direction of motion $\dot{\mathbf{p}}(t_0)$ are linearly dependent. Moreover, we cannot evaluate (3.33) if the denominator equals zero, i.e. the tangent of $\mathcal{E}_2(t_0)$ in $\mathbf{p}$ is perpendicular to the normal of $\mathcal{F}$ in $\mathbf{p}$. We used one more precondition in the development of this approach, namely that there is a neighbourhood $\mathcal{I}$ of t_0 such that $\mathcal{C}_2(t)$

intersects $\mathcal{S}$ for all $t \in \mathcal{I}$. But if the denominator of (3.33) is non-zero, i.e. the tangent of $\mathcal{C}_2(t_0)$ is not perpendicular to $\mathbf{n}$, this precondition is fulfilled. Hence, these are all degenerate cases.

As already stated in the previous paragraph, we can use this approach in many cases to decide whether there is a vertex-face collision if the vertex $\mathbf{p}(t)$ hits an edge $\mathcal{E}_1$ of the face $\mathcal{F}$. If $\mathbf{p}$ witnesses a collision at time t_0 in this case, then there is an edge $\mathcal{E}_2$ which is adjacent to $\mathbf{p}$, that penetrates $\mathcal{F}$ immediately after the collision time. Hence, we can use the above approach for the edges $\mathcal{E}_1$ and $\mathcal{E}_2$ and the face $\mathcal{F}$ to compute the sign of $\dot{\gamma}(t_0)$. If this value is positive, then we know that immediately after the collision time the curve containing $\mathcal{E}_2$ penetrates $\mathcal{F}$ in a neighbourhood of $\mathbf{p}$. Since $\mathbf{p}$ is an endpoint of $\mathcal{E}_2$, we must additionally check whether the intersection point will lie inside or outside $\mathcal{E}_2$. This edge is given by the parameterization $\mathbf{x}_2(\beta)$ and a parameter interval $[a, b]$. Since $\mathbf{p}$ is a vertex of $\mathcal{E}_2$, we know that β_0 is one of the endpoints of this interval. Let us assume that $\beta_0 = a$. Using Cramer's rule, we determine $\dot{\beta}(t_0)$ from (3.32), which yields

$$\dot{\beta}(t_0) = \frac{\dot{\mathbf{p}}(t_0)^T \mathbf{n}}{\mathbf{w}^T \mathbf{n}}.$$

If this value is positive, then there is a penetration, if it is negative, there is not. If $\dot{\beta}(t_0) = 0$, we cannot make a decision. In the case $\beta_0 = b$ the interpretation of the sign is the other way around.

Edge-Face and Face-Face In the case of a potential collision between a face $\mathcal{F}$ of $\mathcal{O}_1$ and an edge or a face of $\mathcal{O}_2(t)$ we proceed similarly to the vertex-face case. Let $\mathbf{p}(t)$ be the potential collision point associated with object $\mathcal{O}_2(t)$ and let the surface containing $\mathcal{F}$ be given by the implicit form $S(\mathbf{x}) = 0$. As in the vertex-face case, we define $\delta(t) = \big(1 + (wt)^2\big)^k S\big(\mathbf{p}(t)\big)$. Again, we define Δ as the first non-zero derivative of $\delta(t)$ at time t_0 in the case that $\delta(t) \not\equiv 0$, and $\Delta = 0$, otherwise. If $\Delta > 0$, we conclude that $\mathbf{p}$ witnesses a collision at time t_0, and if $\Delta < 0$ we conclude that it does not. In the case $\Delta = 0$, we cannot make a decision and therefore consider this case as degenerate.

Degenerate Situations Now, we show exemplarily for the case of a potential collision between two edges $\mathcal{E}_1$ and $\mathcal{E}_2$ which lie on conics $\mathcal{C}_1$ and $\mathcal{C}_2$ that the penetration test can be formulated as the problem to decide whether the origin belongs to the boundary of a full-dimensional cell in a semi-algebraic set in $\mathbb{R}^4$. Let for $i = 1, 2$ the quadrics $\mathcal{S}_{i,1}$ and $\mathcal{S}_{i,2}$ be the surfaces containing the two faces that are adjacent to $\mathcal{E}_i$. We denote the quadratic form defining $\mathcal{S}_{i,j}$ at time t by $S_{i,j}(\mathbf{x}, t) = 0$. By an appropriate translation we achieve that the potential collision point $\mathbf{p}$ equals the origin. W.l.o.g. we can also assume that the potential collision time t_0 equals zero. Thus, we have $S_{i,j}(0, 0) = 0$ for $i, j = 1, 2$. Let $\mathbf{n}_{i,j}$ be the normal of $\mathcal{S}_{i,j}$ at the origin. Further, let $\mathbf{u}_i$ be the tangent of $\mathcal{E}_i$ at the origin with respect to the face embedded in $\mathcal{S}_{i,1}$, i.e. the interior of that face lies to the left of $\mathcal{E}_i$ with respect to the tangent $\mathbf{u}_i$. For the sake of simplicity, we assume that both edges $\mathcal{E}_1$ and $\mathcal{E}_2$ are locally convex, i.e. $(\mathbf{u}_i \times \mathbf{n}_{i,1})^T \mathbf{n}_{i,2} > 0$. In this case, the penetration test must check whether there is an $\varepsilon > 0$ such that for each neighbourhood $\mathcal{N}$ of the origin in $\mathbb{R}^3$ there is a point $x \in \mathcal{N}$ such that $S_{i,j}(x,t) < 0$ for all $t \in (0,\varepsilon)$. This is equivalent to the question whether for each neighbourhood $\mathcal{M}$ of the origin in $\mathbb{R}^4$ there is a vector $[x^T,t]^T \in \mathcal{M}$ such that

$$\begin{aligned} S_{i,j}(x,t) &< 0 \quad \text{for } i,j = 1,2, \\ t &> 0. \end{aligned}$$

In this way we have formulated the penetration test as the question whether the semi-algebraic set in $\mathbb{R}^4$ defined by the above inequalities contains a full-dimensional cell whose boundary contains the origin. Unfortunately, this problem turns out to be extremely difficult, and it is an open problem how to solve it efficiently. In principle this problem can be attacked algebraically with the so called cylindrical algebraic decomposition (CAD) algorithm (see e.g. [ACM84] for a detailed description). But this approach is impractical for our purposes because it constructs a huge number of univariate polynomials of high degree. The idea of the CAD algorithm is the following. In order to compute the cells of an arrangement of surfaces in $\mathbb{R}^d$ one performs a projection step and obtains an arrangement of surfaces in $\mathbb{R}^{d-1}$. Each cell in this lower dimensional arrangement corresponds to a finite number of sign-invariant regions in the original arrangement. By choosing a point from each cell in the $(d-1)$-dimensional arrangement and shooting a ray in the direction of the projection one reaches all sign-invariant regions in the d-dimensional arrangement. In this way one successively reduces the dimension of the problem until one finally has a set of univariate polynomials. By choosing a point from each sign-invariant interval defined by the roots of these polynomials one constructs points in the sign-invariant regions of the original arrangement by successive ray-shooting. The projection step from $\mathbb{R}^d$ to $\mathbb{R}^{d-1}$ comprises the computation of all pairwise resultants of the polynomials defining the d-dimensional arrangement and the resultant of each polynomial and its derivative with respect to the projection direction. If one starts with n polynomials in d variables, then after d projection steps one has $\Omega(n^{(2^d)})$ univariate polynomials in the worst case. Moreover, the degrees of the polynomials raise in each projection step.

3.3.3 Specialization for Natural Quadratic Complexes

We show, that also for the class of natural quadratic complexes the task of computing the potential collision times can be reduced to solving polynomial equations. As in the case of quadratic complexes, we analyze the degrees of these equations. Since most of the computations that are done in this section are specializations of those done in the previous section, we will keep the discussions of them relatively brief.

Vertex-Face Test

Since we have already analyzed the collision test for a vertex and a plane in section 3.3.2, we assume that the surface is a sphere, a circular cone or a circular cylinder. By verifying

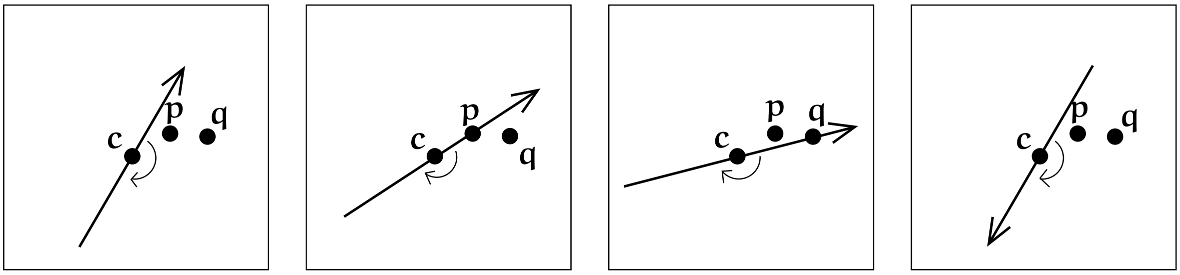


Figure 3.14: A rotating straight line colliding with two points in the plane.

the calculations in the vertex-face case in section 3.3.2, we see that the assumption that the surface is a circular cone or a circular cylinder does not lead to lower degrees. Intuitively, it is clear that the degree of a polynomial equation that has to be solved to determine the potential collision times in the case of a pure rotation has to be four in the cylinder and cone case. This is because a circular trajectory hits the surface in four points, in general. Similarly, it is also clear, that the degree in the case of a pure translation must be two. Inspecting the calculations, we observe that in the case of a superposition these two degrees sum up to six.

If the surface is a sphere, then we find that the degree in the case of a pure rotation drops to two. This is because any circular trajectory hits a sphere in at most two points. In the case of a pure translation, the degree is still two. If we have a superposition of a translation and a rotation, then we find that the degree is four. Thus, we have the following theorem.

Theorem 3.22. *Finding the potential collision times between a vertex and a sphere can be reduced to finding the roots of polynomials of degree at most two in the case of a pure translation or pure rotation and four in the case of a superposition. If the surface is a circular cone or cylinder, then the degrees are two, four and six, respectively.*

Edge-Edge Test

We have already analyzed the case of both curves being straight lines. If one curve is a straight line and one is a circle, then obviously, the degree of a polynomial equation whose roots are the potential collision times must be at least two in the case of a pure translation. The case of a pure rotation is illustrated in figure 3.14. Let the circle intersect the drawing plane in the points $\mathbf{p}$ and $\mathbf{q}$. The line lies in the drawing plane and passes through the point $\mathbf{c}$. Let the axis of rotation be normal to the drawing plane and let $\mathbf{c}$ be the center of rotation. The direction of the rotation is indicated by the arc. The leftmost picture shows the situation at time 0. The situation after a rotation about 180 degrees is shown in the rightmost picture. In between, the line hits the circle first in $\mathbf{p}$ and then in $\mathbf{q}$. As the line rotates on until 360 degrees, it hits the circle once more in $\mathbf{p}$ and $\mathbf{q}$. Thus, it is clear that the degree of the polynomial equation yielding the potential collision times must be at least four. Hence, the degree bounds are the same as in the line-conic case in section 3.3.2.

Let us now consider the case that both curves are circles. As in the previous section, let these curves be given by two quadrics and two planes. Thus, we have a system of equations of the form (3.22). By applying an appropriate translation and rotation we achieve that the stationary circle lies in the (x_1, x_2)-plane. This means that $\mathbf{A}$ has diagonal form with $a_{11} = a_{22}$ and that $a_1 = a_2 = 0$. Moreover we have $\mathbf{m} = [0, 0, 1]^T$ and $m_0 = 0$. W.l.o.g. we assume that the quadratic form given by $\mathbf{B}_H$ has the form $(\mathbf{x} - \mathbf{p})^2 - \rho^2 = 0$, which defines a sphere with center $\mathbf{p}$ and radius ρ. Furthermore, we assume that the plane defined by $\mathbf{n}$ and n_0 passes through $\mathbf{p}$, and hence $n_0 = \mathbf{n}^T\mathbf{p}$. The further calculations are analogous to the previous section. Because of $\mathbf{m} = [0, 0, 1]^T$ and $m_0 = 0$ we can replace $x_3 = 0$ in (3.22). We eliminate x_1 and x_2 by means of resultants and replace all occurrences of $\mathbf{p}$, $\mathbf{n}$ and n_0 in the Sylvester matrix $\mathbf{Syl}_{x_1}(g_1, g_2)$ by the corresponding time dependent functions. A slight difference to the previous section is that the first row of this Sylvester matrix is divisible by $n_1(t)^2 + n_2(t)^2$, which is a polynomial of degree four in t if $\boldsymbol{\omega} \neq 0$. Hence, the degree of the polynomial $r(t)$ in corollary 3.14 is four in the case of a pure translation or rotation and eight in the case of a superposition. Thus, we have the following result.

Theorem 3.23. *Finding the potential collision times between two circles can be reduced to finding the roots of polynomials of degree at most four in the case of a pure translation or rotation and eight in the case of a superposition. In case that at least one of the curves is a straight line the bounds given in theorem 3.16 hold.*

Edge-Face-Test

If $\mathcal{C}$ is a straight line, then by a similar argument as in the edge-edge case we see that the degrees of the polynomial whose roots are the candidates for the potential collision times cannot be less than those derived in the line-quadric case in section 3.3.2. Thus, we assume that $\mathcal{C}$ is a circle. The same holds for the case that $\mathcal{S}$ is a plane. If $\mathcal{S}$ is a circular cone or a cylinder, then by reviewing the calculations in section 3.3.2 we find that we do not achieve better degree bounds than in the conic-quadric case. Hence, we assume that $\mathcal{S}$ is a sphere. We first consider the case that $\mathcal{C}$ belongs to the stationary object $\mathcal{O}_1$. By applying an appropriate transformation to the scene we assume that $\mathcal{C}$ lies in the (x_1, x_2)-plane with its center at the origin. Let the radius of $\mathcal{C}$ be σ. Furthermore, let $\mathbf{r}$ be the center of $\mathcal{S}$ and let ρ be its radius. As in the previous section, we have to look for tangential intersection points between $\mathcal{C}$ and the intersection curve $\mathcal{D}$ between $\mathcal{S}$ and the (x_1, x_2)-plane. The quadratic forms of $\mathcal{C}$ and $\mathcal{D}$ are given by the matrices $\mathbf{A}_H = \mathrm{diag}(1, 1, -\sigma^2)$ and

$$\mathbf{C}_H = \begin{bmatrix} 1 & 0 & -r_1 \\ 0 & 1 & -r_2 \\ -r_1 & -r_2 & \mathbf{r}^2 - \rho^2 \end{bmatrix},$$

respectively. As in the previous section, we define $\Delta(\mathcal{C},\mathcal{S})$ as the discriminant of the polynomial $\chi_{\mathbf{A},\mathbf{C}}(\lambda)$. Again, by lemma 3.19 we have to look for the points in time when this discriminant vanishes. We compute $\Delta(\mathcal{C},\mathcal{S})$ symbolically with the help of Maple® and find a polynomial $\Delta'(\mathcal{C},\mathcal{S})$ such that $\Delta(\mathcal{C},\mathcal{S}) = (r_1^2+r_2^2)^2\Delta(\mathcal{C},\mathcal{S})$. Thus, the discriminant vanishes whenever $\mathbf{r}$ lies on the x_3-axis. If we set $r_1 = r_2 = 0$ in $\Delta'(\mathcal{C},\mathcal{S})$ we get $(\rho^2-\sigma^2-r_3^2)^2$. This expression is zero if and only if r_3 is chosen such that $\mathcal{C}$ and $\mathcal{S}$ touch. Hence, we can omit the factor $(r_1^2+r_2^2)^2$ and look for the roots of $\Delta'(\mathcal{C},\mathcal{S})$. In contrast to the case of a general conic and a general quadric, it is possible to replace $\mathbf{m}$ by the corresponding time dependent function $\mathbf{m}(t)$ in the symbolic representation of $\Delta'(\mathcal{S},\mathcal{C})$ using Maple® . After multiplying with the denominator we obtain a polynomial of degree four in the case of a pure translation or rotation and eight in the case of a superposition.

Now, we consider the case that $\mathcal{S}$ belongs to the stationary object $\mathcal{O}_1$. Let $\mathbf{A}_H = \text{diag}(1,1,1,-\rho^2)$ be the matrix defining $\mathcal{S}$ and let $\mathcal{C}$ lie in the plane $\mathcal{P}$ with the parameterization $\mathbf{x}(\alpha,\beta) = \mathbf{r}+\alpha\mathbf{p}+\beta\mathbf{q}$ where $\mathbf{p}$ and $\mathbf{q}$ are orthogonal unit vectors. Let $\mathbf{B}_H = \text{diag}(1,1,-\sigma^2)$ be the matrix defining $\mathcal{C}$ in (α,β)-coordinates. We obtain an implicit form in (α,β)-coordinates of the intersection circle $\mathcal{D}$ between $\mathcal{S}$ and $\mathcal{P}$ by inserting $\mathbf{x}(\alpha,\beta)$ into the implicit form of $\mathcal{S}$. Let $\mathbf{C}_H$ be the (3×3)-matrix corresponding to this quadratic form. Using Maple® we observe that $\chi_{\mathbf{B},\mathbf{C}}(\lambda) = -(\lambda+1)\cdot f(\lambda)$ for a polynomial $f(\lambda)$. Suppose that $\lambda_0 = -1$ is a multiple root of $\chi_{\mathbf{B},\mathbf{C}}(\lambda)$. Then λ_0 must be a root of $f(\lambda)$. With Maple® we easily verify that $f(-1) = (\mathbf{r}^T\mathbf{p})^2+(\mathbf{r}^T\mathbf{q})^2$, which is zero if and only if the vector $\mathbf{r}$ pointing from the center of $\mathcal{S}$ to the center of $\mathcal{C}$ is parallel to the normal $\mathbf{p}\times\mathbf{q}$ of $\mathcal{P}$. But in this case there is a tangential intersection between $\mathcal{C}$ and $\mathcal{S}$ if and only if $\mathbf{r}^2+\sigma^2-\rho^2 = 0$. Using Maple® again, we observe that in that case λ_0 is also a root of the derivative of f with respect to λ and hence, it is a multiple root of $f(\lambda)$. Thus, we define $\Delta(\mathcal{S},\mathcal{C})$ as the discriminant of the polynomial $f(\lambda)$. We replace all occurrences of $\mathbf{r},\mathbf{p}$ and $\mathbf{q}$ in $\Delta(\mathcal{S},\mathcal{C})$ by the corresponding time dependent function and multiply with the denominator. The result is a polynomial in t whose roots are the candidates for the potential collision times. The degree of this polynomial is four in the case of a pure translation or rotation and eight in the case of a superposition. We record this result in the following theorem.

Theorem 3.24. *If $\mathcal{C}$ and $\mathcal{S}$ in theorem 3.20 are a circle and a sphere, respectively, then their potential collision times can be found by solving polynomials of degree at most four in the case of a pure translation or rotation and at most eight in the case of a superposition.*

Face-Face Test

If one of the faces is embedded on a sphere or a circular cylinder, then we can use a configuration space approach to detect the collisions between them. This is because of the observation that the offset of a natural quadric consists of two natural quadrics. Based on this observation, we distinguish three cases.

Case 1: One of the surfaces is a sphere. We denote this sphere by $\mathcal{S}$, and the other surface by $\mathcal{Q}$. Let r be the radius and $\mathbf{c}$ the center of $\mathcal{S}$. Obviously, $\mathcal{S}$ and $\mathcal{Q}$ touch tangentially if and only if $\mathbf{c}$ lies on the $\pm r$-offset of $\mathcal{Q}$. Thus, we have to determine the potential collision times between a point and two natural quadrics. If $\mathcal{Q}$ is a plane, then by theorem 3.11, the degrees of the polynomial equations that have to be solved are one in the case of a pure translation, two in the case of a pure rotation and three in the case of a superposition. If $\mathcal{Q}$ is a sphere, then by theorem 3.22 the degrees are two, two and four respectively. This theorem also states that in the case of $\mathcal{Q}$ being a circular cone or cylinder the degrees are two, four and six.

Case 2: One of the surfaces is a circular cylinder which we denote by $\mathcal{S}$. Let $\mathcal{Q}$ denote the second surface. As $\mathcal{S}$ is ruled we do not have to consider the case that $\mathcal{Q}$ is a plane. We can also assume that $\mathcal{Q}$ is not a sphere, because this is covered by case 1. Hence, $\mathcal{Q}$ is a circular cone or cylinder. Let $\mathbf{r}$ be the radius and $\mathcal{L}$ the symmetry axis of $\mathcal{S}$. Obviously, $\mathcal{S}$ and $\mathcal{Q}$ have a tangential intersection if and only if $\mathcal{L}$ intersects the $\pm r$-offset of $\mathcal{Q}$ tangentially. This offset consists of two circular cylinders or two circular cones. By theorem 3.20, the degrees of the polynomial equations that have to be solved to compute the potential collision times between $\mathcal{L}$ and the offset of $\mathcal{Q}$ are at most two in the case of a pure translation, four in the case of a pure rotation and six in the case of a superposition.

Case 3: Both surfaces are circular cones. Let $\mathbf{A}_H$ and $\mathbf{B}_H$ be the matrices defining the quadratic forms of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. By corollary 2.29, the polynomial $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ has a positive multiple root if $\mathcal{S}_1$ and $\mathcal{S}_2$ touch externally. Hence, we are looking for the points in time when the discriminant of $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ is zero. By inspecting the entries of the degree matrix (3.27) we see that the degrees of the coefficients k_i of $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ after multiplying with their least common denominator is upper-bounded by six. Since the determinants of both $\mathbf{A}_H$ and $\mathbf{B}_H$ are zero, the degree of $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$ is three and the coefficient k_0 is zero. Hence, $\chi_{\mathbf{A},\mathbf{B}}(\lambda) = \lambda(k_3\lambda^2+k_2\lambda+k_1)$. Since we are only interested in positive multiple roots of $\chi_{\mathbf{A},\mathbf{B}}(\lambda)$, we compute the discriminant of the quadratic factor, which is $k_2^2 - 4k_1k_3$. Multiplying this with the denominator, we obtain a polynomial of degree at most twelve. Similarly, we find that the degree is eight in the case of a pure rotation and four in the case of a pure translation.

We summarize the results of the face-face case in the following theorem.

Theorem 3.25. *The potential collision times between two natural quadrics can be determined by solving polynomial equations whose degrees are upper-bounded by the following table.*

	translation	*rotation*	*superposition*
sphere vs. plane	1	2	3
sphere vs. sphere	2	2	4
one surface is a circular cylinder	2	4	6
circular cone vs. circular cone	4	8	12

4 Dynamics Simulation

In this chapter we give an overview over the different techniques to simulate the dynamics of a system of rigid bodies. These simulation techniques play an important role in a variety of interactive VR-applications such as virtual prototyping or ergonomy studies. For instance, in order to interactively simulate an assembly of mechanical parts in a virtual environment the simulation software must be able to react realistically on collisions between the involved objects, as demonstrated in figure 4.1. This figure shows the manual insertion of a bolt into a countersunk nut. This is a difficult task in a virtual environment if the motion is simply stopped as soon as a collision occurs. In reality the bolt automatically slides into the right direction when it comes into contact with the boundary of the hole. It is desirable to simulate this effect in order to perform virtual fitting operations in a more intuitive manner.

We describe two different approaches to calculate physically correct reactions to collisions. These are the *impulse based simulation* and the *constraint based simulation*. In the context of the latter we derive a new method to simulate the rolling motion of rigid objects on arbitrary surfaces.

4.1 Impulse Based Dynamics Simulation

In the impulse based approach the motion of the objects is partitioned into collision free time intervals and collision times. In the collision free intervals the objects move on ballistic trajectories. This means that there is no interaction between any two objects but there may be external forces. These external forces might be gravitational forces, magnetic forces, etc. At the collision times, the impulse responses at the contact points are computed. These responses depend on the coefficients of restitution and friction. This approach is described in detail in [MC95], [Mir96b] and [Len00]. The collision detection used in these publications is based on distance computations. Using the linear and angular velocities, the radii of the objects and their pairwise distances, lower bounds for the duration of the collision free interval are computed and the motions of the objects during this tentative interval are simulated. After that, new lower bounds are computed and the simulation over the collision free interval is continued. This is repeated until the distance between two objects falls below a given threshold. Then, the closest points between the pair of objects that have the smallest distance from each other are declared as collision points and the impulse response in these points is computed. This response causes the objects to move on ballistic trajectories again immediately after the collision.

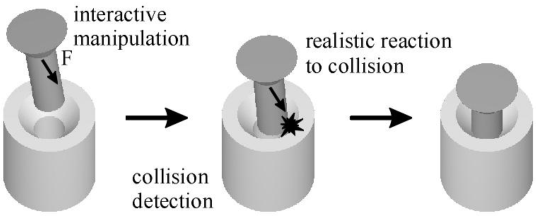


Figure 4.1: The insertion of a bolt into a conical nut

The duration of the collision itself is considered as infinitesimally short. Moreover, this approach only considers one contact point at a time. Permanent and multiple contacts are modelled as a series of so-called *micro contacts*.

We will briefly describe the simulation of the ballistic motion in the collision free intervals as well as the computation of the impulse response in the absence of friction.

The ballistic motion

If there is no interaction between any two objects, the motions are described by the motion equations (2.32) and (2.33). Let us assume that the only external force is the gravitational force $\mathbf{f}_G$. Since the objects do not interact, this is the only force and hence, the right-hand side of the upper equation in (2.33) is just $\mathbf{f}_G$ and the right-hand side of the lower one is $\mathbf{0}$. This is because the gravitation acts on the center of mass and thus causes no torque. The equations (2.32) and (2.33) form a system of first order ODEs that can be integrated by any standard numerical method, such as the Runge-Kutta method which can be found in [PTVF94].

Computation of the impulse response

As already mentioned, each collision is considered to be of duration zero. But in order to compute the impulses after a collision such an event is mathematically modelled as two phases, namely the *compression phase* and the *restitution phase*. In the compression phase, the kinetic energy of the colliding objects is transformed to potential energy. In the restitution phase, a part of this energy is transformed back to kinetic energy. The amount of the energy which is given back to the objects depends on the coefficients of friction and restitution. Since we neglect friction here for the sake of simplicity, we only have to consider the coefficient of restitution denoted by e. This coefficient lies in the interval $[0, 1]$ and relates the work after the collision to the work done during the compression phase. The coefficient of restitution is a material property of the object.

Let $\mathbf{n}$ be the contact normal. In case that the interior of a face is involved in the contact, $\mathbf{n}$ is the normal of that face. In the case of a contact between two edges, $\mathbf{n}$ is parallel to the cross-product of the tangents of the edges in the contact point.

Stronge's hypothesis Let W_z be the work done in the direction of $\mathbf{n}$. Then

$$W_z(t_f) = (1 - e^2)W_z(t_m),$$

where $W_z(t_f)$ denotes the work after the collision and $W_z(t_m)$ the work at the point of maximal compression.

The impulse of the system changes during both phases of the collision. Let $\Delta\mathbf{p}$ be this change. Because of the conservation of momentum, we must apply $\Delta\mathbf{p}$ to both objects but in opposite directions. Let $\mathbf{n}$ point locally away from object $\mathcal{O}_2$. Then $\Delta\mathbf{p}$ is applied to $\mathcal{O}_1$ and $-\Delta\mathbf{p}$ to $\mathcal{O}_2$. With $\Delta\mathbf{u}_i$ we denote the change of the velocity of the contact point of object $\mathcal{O}_i$. Since impulse changes are only applied to the contact point, the changes of the total linear and angular momenta are equal to the changes of the linear and angular momenta in the contact point. The changes of these momenta are

$$\begin{aligned}\Delta\mathbf{p} &= m\Delta\mathbf{v} \quad \text{and} \\ \Delta\mathbf{l} &= \mathbf{r}\times\Delta\mathbf{p} = \mathbf{I}\Delta\boldsymbol{\omega},\end{aligned}$$

where $\mathbf{r}$ is the vector from the center of mass to the contact point. For the changes of velocity it holds

$$\begin{aligned}\Delta\mathbf{u}_1 &= \Delta\mathbf{v}_1 + \Delta\boldsymbol{\omega}_1 \times \mathbf{r}_1 \\ &= \frac{1}{m_1}\Delta\mathbf{p} + \mathbf{I}_1^{-1}(\mathbf{r}_1\times\Delta\mathbf{p})\times\mathbf{r}_1, \quad \text{and similarly} \\ \Delta\mathbf{u}_2 &= -\frac{1}{m_2}\Delta\mathbf{p} - \mathbf{I}_2^{-1}(\mathbf{r}_2\times\Delta\mathbf{p})\times\mathbf{r}_2.\end{aligned}$$

The relative contact velocity is given by $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2$. The change of the contact velocity is linearly related to the change of the impulse by

$$\Delta\mathbf{u} = \mathbf{K}\Delta\mathbf{p}$$

with $\mathbf{K} = (\frac{1}{m_1} + \frac{1}{m_2})\mathbf{E} - \mathbf{r}_1^\times\mathbf{I}_1^{-1}\mathbf{r}_1^\times - \mathbf{r}_2^\times\mathbf{I}_2^{-1}\mathbf{r}_2^\times$. This so-called *collision matrix* is positive definite and therefore regular because for $\mathbf{x} \neq 0$ it holds

$$\mathbf{x}^T\mathbf{K}\mathbf{x} = (\frac{1}{m_1} + \frac{1}{m_2})\mathbf{x}^2 + (\mathbf{r}_1\times\mathbf{x})^T\mathbf{I}_1^{-1}(\mathbf{r}_1\times\mathbf{x}) + (\mathbf{r}_2\times\mathbf{x})^T\mathbf{I}_2^{-1}(\mathbf{r}_2\times\mathbf{x}) > 0.$$

Hence, if we know the change of the relative velocity we can compute the change of the impulse as $\Delta\mathbf{p} = \mathbf{K}^{-1}\Delta u$. Since a collision has no duration, we may assume that the vectors $\mathbf{r}_i$ and the matrices $\mathbf{I}_i$ are constant and hence, $\mathbf{K}$ is constant, as well.

For the following computations we assume that we have applied an appropriate rotation and translation to the coordinate system such that the contact point lies in the origin and the contact normal $\mathbf{n}$ coincides with the x_3-axis. Figure 4.2 sketches the curves of the work done during the collision and the component of the contact velocity in the direction of $\mathbf{n}$. Since $\mathbf{n}$ points from $\mathcal{O}_2$ to $\mathcal{O}_1$ and by our definition of $\mathbf{u}$, the value u_z is non-positive at the beginning of the collision and non-negative at the end. Hence,

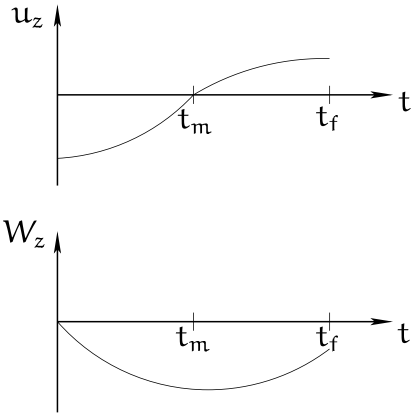


Figure 4.2: The component of the contact velocity and the work done during the collision in the direction of the contact normal.

the work measured during the collision in the direction of $\mathbf{n}$ is always non-positive. At the end of the compression phase the value of u_z is zero.

Since we model the collision as two phases, let these be parameterized by γ. At first glance it seems natural to choose the time parameter as γ. But as the collision has no duration, t would be inappropriate as the collision parameter. We require γ to be monotonic in t, and we demand that all parameterized quantities are continuous in γ. In [Mir96b] it is shown that the third component u_z of the relative contact velocity $\mathbf{u}$ as well as the work W_z done in the direction of the contact normal fulfill these requirements. The idea how to compute the change of the impulse during the collision is the following. For each parameterized quantity q we write $q(t_s)$ for the value immediately before the collision, $q(t_m)$ for the value at maximum compression and $q(t_f)$ for the value immediately after the collision.

1. Compression integration. At the end of the compression phase, the component u_z of the relative contact velocity $\mathbf{u}$ must be zero, i.e. $u_z(t_m) = 0$. We choose $\gamma = u_z$ as the collision parameter for this phase and compute the work $W_z(t_m)$ and the velocity $\mathbf{u}(t_m)$ as

$$\mathbf{u}(t_m) = \mathbf{u}(t_s) + \int_{u_z(t_s)}^{0} \frac{d\mathbf{u}}{du_z}(u_z)du_z \quad \text{and} \tag{4.1}$$

$$W_z(t_m) = \int_{u_z(t_s)}^{0} \frac{dW_z}{du_z}(u_z)du_z. \tag{4.2}$$

Then, we compute the work done at the end of the collision using Stronge's hypothesis, namely $W_z(t_f) = (1 - e^2)W_z(t_m)$.

2. Restitution integration. Since we now know the amount of work done in the direction of the contact normal at the end of the collision, we choose $\gamma = W_z$ as

the collision parameter for this phase and compute the relative contact velocity at the end of the collision as

$$\mathbf{u}(t_f) = \mathbf{u}(t_m) + \int_{W_z(t_m)}^{W_z(t_f)} \frac{d\mathbf{u}}{dW_z}(W_z)dW_z. \tag{4.3}$$

The change of the relative contact velocity is $\Delta\mathbf{u} = \mathbf{u}(t_f) - \mathbf{u}(t_s)$. Now, the change of the impulse can be computed as $\Delta\mathbf{p} = \mathbf{K}^{-1}\Delta\mathbf{u}$.

We compute the integrands in these three equations. Since $\mathbf{K}$ is constant, we observe that

$$\frac{d\mathbf{u}}{d\gamma} = \mathbf{K}\frac{d\mathbf{p}}{d\gamma} = \mathbf{K}\frac{d\mathbf{p}}{dt}\frac{dt}{d\gamma} = \mathbf{K}\mathbf{f}_r\frac{dt}{d\gamma},$$

where $\mathbf{f}_r$ denotes the reaction force causing the impulse change. If there is no friction, $\mathbf{f}_r = f_r\mathbf{n} = [0, 0, f_r]^T$. Therefore we have

$$\frac{d\mathbf{u}}{d\gamma} = \mathbf{K}\mathbf{n}f_r\frac{dt}{d\gamma}. \tag{4.4}$$

Similarly, we obtain

$$\frac{d\mathbf{u}}{dt} = \mathbf{K}\mathbf{n}f_r. \tag{4.5}$$

With this, we get

$$\begin{aligned}
\frac{d\mathbf{u}}{du_z} &\overset{(4.4)}{=} \mathbf{K}\mathbf{n}f_r\frac{dt}{du_z} \overset{(4.5)}{=} \frac{\mathbf{K}\mathbf{n}}{K_{33}}, && (4.6)\\
\frac{dW_z}{du_z} &= \frac{dW_z}{dt}\frac{dt}{du_z} \overset{(4.5)}{=} \mathbf{f}_r^T\mathbf{u}\frac{1}{K_{33}f_r} = \frac{u_z}{K_{33}} \quad \text{and} && (4.7)\\
\frac{d\mathbf{u}}{dW_z} &= \frac{d\mathbf{u}}{du_z}\frac{du_z}{dW_z} \overset{(4.6)(4.7)}{=} \frac{\mathbf{K}\mathbf{n}}{u_z}. && (4.8)
\end{aligned}$$

If we insert (4.6) into (4.1) and (4.7) into (4.2) we obtain $\mathbf{u}(t_m) = \mathbf{u}(t_s) - \frac{\mathbf{K}\mathbf{n}}{K_{33}}u_z(t_s)$ and $W_z(t_m) = -\frac{1}{2K_{33}}u_z(t_s)^2$. By (4.7) and $W_z(t_s) = 0$ we find $W_z(u_z) = \frac{1}{2K_{33}}(u_z^2 - u_z(t_s)^2)$. We conclude that $u_z(W_z) = \pm\sqrt{2K_{33}W_z + u_z(t_s)^2}$. In the compression phase, we have to choose the minus sign, and in the restitution phase the plus sign. Using this and $W_z(t_f) = \frac{e^2-1}{2K_{33}}u_z(t_s)^2$, we obtain from (4.3)

$$\mathbf{u}(t_f) = \mathbf{u}(t_s) - (1+e)\frac{\mathbf{K}\mathbf{n}}{K_{33}}u_z(t_s),$$

and hence

$$\Delta\mathbf{p} = -(1+e)\frac{u_z(t_s)}{K_{33}}\mathbf{n}.$$

In the presence of friction it is not so easy to solve the above integrals symbolically. In that case one uses numerical integration methods such as the Runge-Kutta method. But then, the division by u_z in (4.8) causes a problem in the integration of (4.3). Since

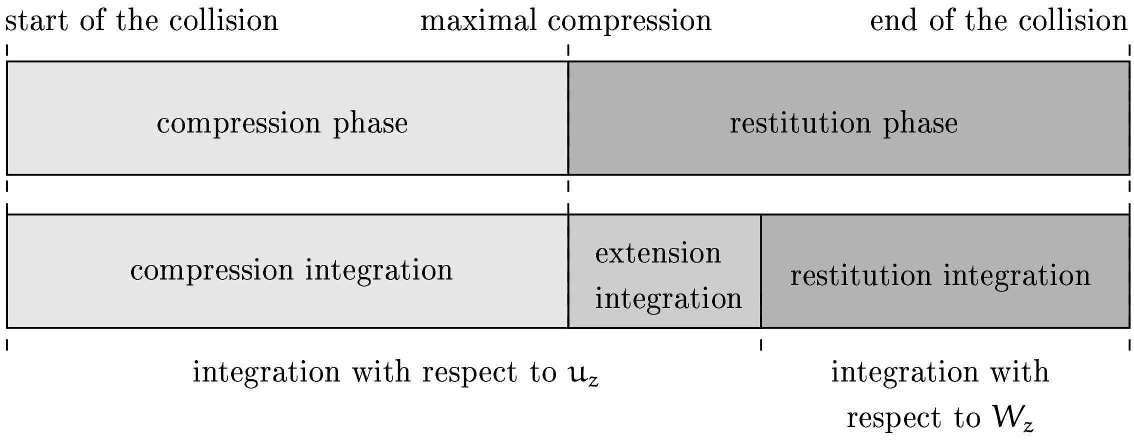


Figure 4.3: The integrations during the collision phases.

$u_z(t_m) = 0$, one gets a division by zero at the beginning of the restitution phase. To avoid this, one extends the compression integration until a parameter value $\overline{u_z} > u_z(t_m)$, where $\overline{u_z}$ is chosen in such a way that the work done up to that value does not exceed $W_z(t_f)$. Then, the restitution integration starts at that point. We call the integration from $u_z = u_z(t_m)$ to $u_z = \overline{u_z}$ the *extension integration*. In the above mentioned publications it is shown how $\overline{u_z}$ can be chosen. Figure 4.3 illustrates the phases of the collision and the required integrations.

Although the publications mentioned before only consider polyhedral objects it is clear that the just described approach also works fine for curved objects, provided one can compute good upper bounds for the collision free intervals as well as the points where the objects get in touch. The dynamic collision test that we described in section 3.3 can be used for this purpose. Instead of determining the earliest collision time in the interval $[0, 1]$ one extends this interval to $[0, \infty)$. Our algorithm does not only compute the earliest collision time but also the points on the objects that get in touch at that time.

4.2 Constraint Based Dynamics Simulation

In the constraint based approach the motions of the objects are described by the motion equations (2.32) and (2.33) plus additional constraints. These constraints are equalities and inequalities involving the position and orientation parameters of the objects, their linear and angular velocities, the reaction forces occurring at contact points, etc. Examples for constraints include

- inequalities stating that the local distances at the contact points are non-negative. This ensures that there are no interpenetrations.
- inequalities describing the frictional forces at the contact points.

- equalities saying that there should be a rolling motion at certain contact points by demanding that the relative contact velocities in these points are zero.

A well studied constraint based method is the simulation by so-called *contact forces.* In 4.2.1 we will briefly describe this method for the case that the objects are polyhedra which goes back to [Bar94] and was refined by several later works including [BS98] and [War99]. The publications [ST95b] and [SS98] extend the approach by including friction. In 4.2.2 we will describe a new constraint based method to simulate the rolling motions of curved objects. We published this method in [WS01].

4.2.1 Simulation by Contact Forces

Contact forces are the forces that occur in the contact points and prevent the objects from interpenetrating by pushing them apart. We describe a generic algorithm for a simulation with contact forces and how these forces can be computed in the case of polyhedral objects. We start with the computation of the contact forces. For the sake of simplicity we neglect friction.

Computation of contact forces

Suppose we have a scene with n objects $\mathcal{O}_1, \ldots, \mathcal{O}_n$. At each point in time during the simulation there is a number of mutual contacts between these objects. Suppose we are in a situation with k such contacts. We introduce for each such contact a contact force $\mathbf{f}_l, l = 1, \ldots, k$. Since we do not consider friction, the force $\mathbf{f}_l$ is parallel to the contact normal $\mathbf{n}_l$. Therefore, we write $\mathbf{f}_l = f_l \mathbf{n}_l$. Let $\mathcal{O}_{i_l}$ and $\mathcal{O}_{j_l}$ be the objects involved in the lth contact. We assume that the contact normal $\mathbf{n}_l$ locally points away from $\mathcal{O}_{j_l}$ and towards $\mathcal{O}_{i_l}$. According to Newton's third axiom, each contact force acts upon both objects involved but in opposite direction. We assume that $\mathbf{f}_l$ acts upon $\mathcal{O}_{i_l}$ and $-\mathbf{f}_l$ acts upon $\mathcal{O}_{j_l}$. We define the vectors $\mathbf{r}_{li} = \mathbf{p}_l - \mathbf{c}_i$ pointing from the center of mass of $\mathcal{O}_i$ to the point of the lth contact. Figure 4.4 illustrates a two-dimensional example with four contacts. Now we can rewrite the Newton-Euler dynamics equations (2.33) for object $\mathcal{O}_i$ in the following way.

$$\begin{aligned} \dot{\mathbf{v}}_i &= \frac{1}{m_i}\left(\sum_{i_l=i} f_l \mathbf{n}_l - \sum_{j_l=i} f_l \mathbf{n}_l\right) + \mathbf{g}, \\ \dot{\boldsymbol{\omega}}_i &= \mathbf{I}_i^{-1}\left(\sum_{i_l=i} f_l(\mathbf{r}_{li} \times \mathbf{n}_l) - \sum_{j_l=i} f_l(\mathbf{r}_{li} \times \mathbf{n}_l) - \boldsymbol{\omega}_i \times \mathbf{I}_i\boldsymbol{\omega}_i\right). \end{aligned}$$

We use the symbol $\mathbf{g}$ for the gravitational acceleration. We describe the rotations of the objects using quaternions. Therefore we use (2.34) instead of the second equation in (2.32)

The motion equations form a system of differential equations whose solution describes the motion of the objects exactly, provided we know the functions $\mathbf{F}(t) = \sum \mathbf{f}_l$ and

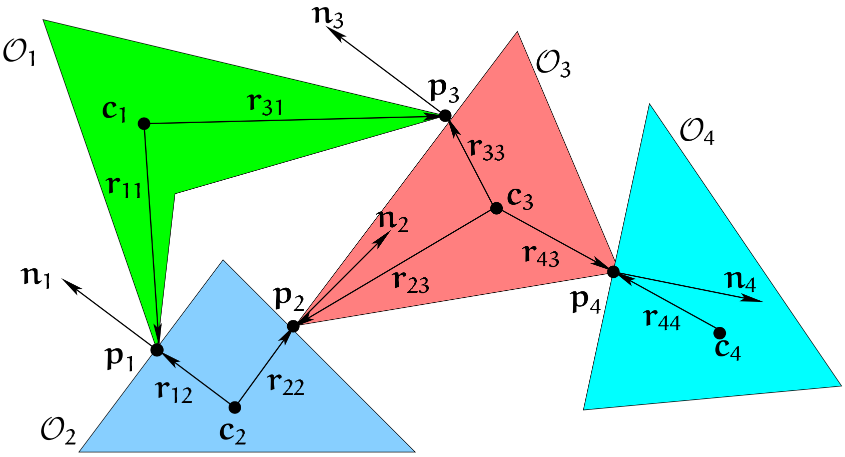


Figure 4.4: Example for a situation with four contacts.

$\mathbf{D}(t) = \sum \mathbf{r}_{li} \times \mathbf{f}_l$. But in the following we assume that we do not know these functions. This makes sense because we want to allow interactive manipulations of the objects and we cannot make predictions on the user's behaviour. Therefore, we assume that we know F and $\mathbf{D}$ only at discrete times $t, t+\Delta t, \ldots$ and that these functions are constant over each interval $[t, t+\Delta t)$. Moreover, we assume that the number k of mutual contacts is constant over each such interval. In order to approximate the positions and orientations of the objects at time $t+\Delta t$, we discretize the motion equations using Euler-scheme and obtain

$$\begin{aligned} \mathbf{c}_i^{t+\Delta t} &= \mathbf{c}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t}, \\ \mathbf{q}_i^{t+\Delta t} &= \left(\mathbf{q}_i^t + \frac{1}{2} \Delta t \boldsymbol{\omega}_i^{t+\Delta t} \mathbf{q}_i^t \right)^0, \end{aligned} \tag{4.9}$$

where $\mathbf{q}^0$ means $\mathbf{q}/|\mathbf{q}|$ and

$$\begin{aligned} \mathbf{v}_i^{t+\Delta t} &= \mathbf{v}_i^t + \frac{\Delta t}{m_i} \left(\sum_{i=i_l} f_l \mathbf{n}_l - \sum_{i=j_l} f_l \mathbf{n}_l + m_i \mathbf{g} \right), \\ \boldsymbol{\omega}_i^{t+\Delta t} &= \boldsymbol{\omega}_i^t + \Delta t (\mathbf{I}_i^t)^{-1} \left(\sum_{i=i_l} f_l (\mathbf{r}_{li} \times \mathbf{n}_l) \right. \\ &\qquad \left. - \sum_{i=j_l} f_l (\mathbf{r}_{li} \times \mathbf{n}_l) - \boldsymbol{\omega}_i^t \times \mathbf{I}_i^t \boldsymbol{\omega}_i^t \right). \end{aligned} \tag{4.10}$$

We used forward differentiation to obtain the equations (4.9) and backward differentiation to obtain (4.10). This enables us to insert (4.10) into (4.9). In this way we can approximate the positions and orientations of the objects at time $t + \Delta t$ provided we know the contact forces at time t.

In order to compute the contact forces we introduce the notion of *contact distances*. Let $\mathbf{p}_{li}$ be the point on object $\mathcal{O}_i$ that is involved in the lth contact. The contact distance in the lth contact is defined as $\delta_l = \mathbf{n}_l^T(\mathbf{p}_{li_l} - \mathbf{p}_{lj_l})$. The δ_j's can be viewed as functions taking the position and orientation parameters of $\mathcal{O}_{i_l}$ and $\mathcal{O}_{j_l}$ and computing the local signed distance in the contact point. The contact distance is zero if the objects are actually in contact in that point. It increases if the objects move locally apart from each other and it decreases if they locally approach. Now we are ready to formulate the constraints that together with the motion equations describe the dynamic behaviour of the objects:

$$f_l \geq 0, \quad \delta_l^{t+\Delta t} \geq 0 \quad \text{and} \quad f_l \cdot \delta_l^{t+\Delta t} = 0$$

for all $l = 1, \dots, k$. The meaning of these constraints is that the contact forces should not be attractive, the contact distances should be non-negative and there should only be a force pushing the objects apart in a contact if they actually touch.

We have argued that the positions and orientations at time $t + \Delta t$ can be viewed as functions of the contact forces. Since the contact distances depend on these parameters, they can be viewed as functions of these forces, as well. We define the vectors $\mathbf{F} = [f_1, \dots, f_k]^T$ and $\boldsymbol{\delta} = [\delta_1, \dots, \delta_k]^T$. We can reformulate the constraints in the form

$$\mathbf{F} \geq \mathbf{0}, \quad \boldsymbol{\delta}(\mathbf{F}) \geq \mathbf{0} \quad \text{and} \quad \mathbf{F}^T\boldsymbol{\delta}(\mathbf{F}) = 0.$$

This is a non-linear complementarity problem (NCP) with the contact forces as variables. In [Kan96] we find an approach to solve such an NCP. Therefore we look at the so-called Fischer function defined as $\varphi : \mathbb{R}^2 \to \mathbb{R}$ with $\varphi(a, b) = \sqrt{a^2 + b^2} - a - b$. Obviously, the Fischer function has the property

$$\varphi(a, b) = 0 \quad \Longleftrightarrow \quad a \geq 0, \quad b \geq 0 \quad \text{and} \quad ab = 0.$$

We use this function to define $\mathbf{G} : \mathbb{R}^{2k} \to \mathbb{R}^{2k}$ as

$$\mathbf{G}(\mathbf{F}, \boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{\delta}(\mathbf{F}) - \boldsymbol{\xi} \\ \boldsymbol{\Phi}(\mathbf{F}, \boldsymbol{\xi}) \end{bmatrix},$$

where $\mathbf{F}, \boldsymbol{\xi} \in \mathbb{R}^k$ and $\boldsymbol{\Phi}(\mathbf{F}, \boldsymbol{\xi}) = [\varphi(f_1, \xi_1), \dots, \varphi(f_k, \xi_k)]^T$. Let now $\mathbf{F} \in \mathbb{R}^k$. We observe that there is a vector $\boldsymbol{\xi} \in \mathbb{R}^k$ with $\mathbf{G}(\mathbf{F}, \boldsymbol{\xi}) = \mathbf{0}$ if and only if $\mathbf{F}$ solves the above NCP. In this way we have reduced the NCP to a non-linear equation system which can be solved using numerical techniques, e.g. the Newton-Raphson method.

There are other methods to compute the contact forces. One of them is to use the constraints $\boldsymbol{\delta}^{t+\Delta t}(\mathbf{F}) = \mathbf{0}$, which leads directly to a system of non-linear equations. This approach allows negative, i.e. attractive contact forces. In the next time step, all those contacts where a negative force occurred are released. The drawback of this approach is that some objects kind of stick together for one simulation step.

Another method, which we will not describe here, is to derive a series of linear complementarity problems (LCPs) that are solved iteratively. The solutions of these LCPs converge towards the solution of the above derived NCP. A detailed description of this approach can e.g. be found in [SS98] or in [War99].

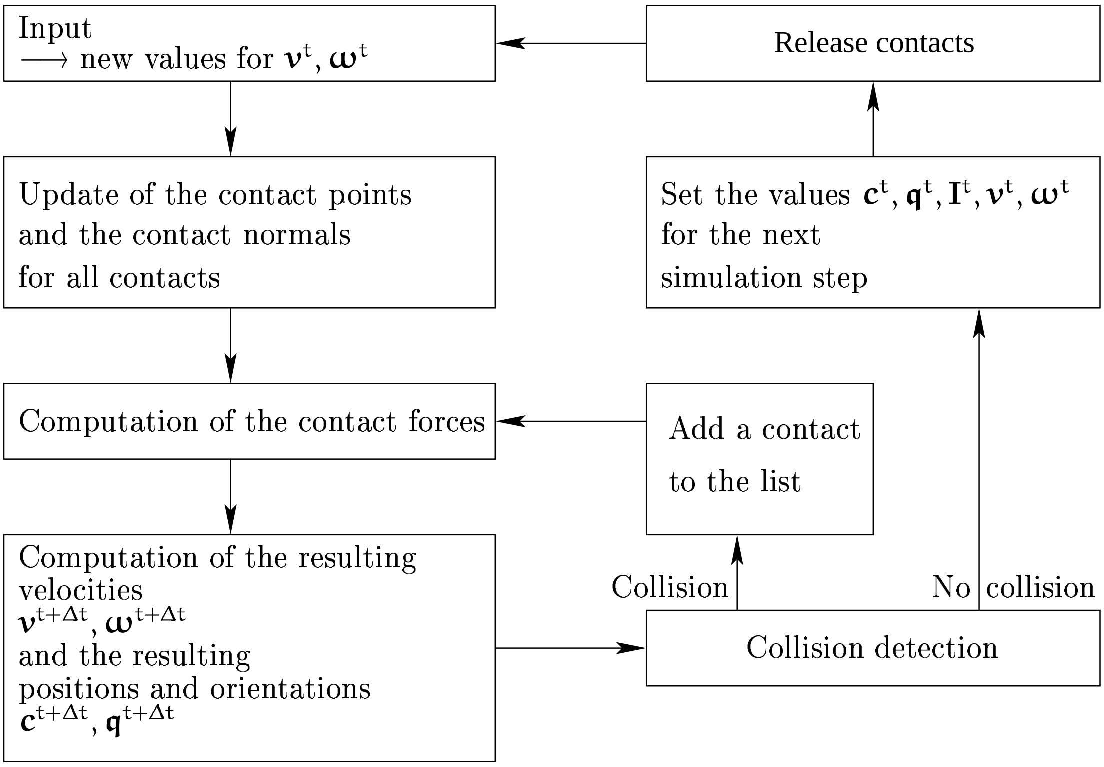

Input
⟶ new values for $\boldsymbol{\nu}^t, \boldsymbol{\omega}^t$
Release contacts
Update of the contact points and the contact normals for all contacts
Set the values $\mathbf{c}^t, \mathbf{q}^t, \mathbf{I}^t, \boldsymbol{\nu}^t, \boldsymbol{\omega}^t$ for the next simulation step
Computation of the contact forces
Add a contact to the list
Computation of the resulting velocities $\boldsymbol{\nu}^{t+\Delta t}, \boldsymbol{\omega}^{t+\Delta t}$ and the resulting positions and orientations $\mathbf{c}^{t+\Delta t}, \mathbf{q}^{t+\Delta t}$
Collision
No collision
Collision detection

Figure 4.5: Schematic chart of a generic algorithm for a constraint based simulation with contact forces.

A generic algorithm for a simulation with contact forces

In this paragraph we will describe a generic algorithm for a constraint based simulation using contact forces. This algorithm maintains a list of contacts between the objects. The data that is stored in a contact includes the type of the contact (vertex-face, edge-face, etc), the contact point, and the contact normal. The list of contacts is empty at the beginning of the simulation. The input of the algorithm consists of the user's input (e.g. via mouse) that can either be interpreted as new velocities or as forces, and the external forces such as gravitational forces. Figure 4.5 illustrates the algorithm. The first step after the input has been taken and the velocities have been set according do this input is to update the contacts in the list by computing the coordinates of the contact points and the contact normals. Then, the contact forces are computed for the current contact situation. Now, the resulting velocities, positions and orientations can be determined using the equations (4.9) and (4.10). Then, we start the collision detection algorithm to check whether this resulting motion causes a collision. If this is the case we add a new contact to the list. This contact stores those parts of the objects that will collide, the collision point as well as the contact normal in that point. Then, the contact forces are computed for this new contact situation. This is repeated until there is no new collision between any two objects. Then, the position, orientation and dynamics parameters of the objects are set for the next simulation step. The last step before the simulation loop is repeated is to release some contacts. A contact is released if geometrically it does not exist any more. This is the case if the corresponding contact distance δ is positive or if the contact point has left one of the faces or edges that were involved.

4.2.2 Simulation of Rolling Motions

In this section we describe the simulation of an object $\mathcal{O}$ rolling on an arbitrary surface $\mathcal{S}$, which has been published in [WS01]. Let $\mathbf{p}$ be an arbitrary contact point between $\mathcal{O}$ and $\mathcal{S}$. As usual, we denote the center of mass of $\mathcal{O}$ by $\mathbf{c}$, its mass and inertia matrix by m and $\mathbf{I}$, respectively, and its linear and angular velocity by $\boldsymbol{v}$ and $\boldsymbol{\omega}$. The relative contact velocity in $\mathbf{p}$ is the velocity of $\mathbf{p}$ as a point on $\mathcal{O}$ relative to the coordinate system associated with the local coordinate system of $\mathcal{S}$. Here, we assume that $\mathcal{S}$ is stationary. Therefore, the relative contact velocity is given by $\boldsymbol{v} + \boldsymbol{\omega} \times \mathbf{r}$, where $\mathbf{r} = \mathbf{p} - \mathbf{c}$. The object rolls on the surface if in each contact point the condition

$$\boldsymbol{v} + \boldsymbol{\omega} \times \mathbf{r} = \mathbf{0} \tag{4.11}$$

is fulfilled. We call this equation the *rolling condition* in $\mathbf{p}$. This condition states that the particles on $\mathcal{O}$ that are currently in contact with $\mathcal{S}$ do not slide along $\mathcal{S}$, i.e. their relative velocity with respect to $\mathcal{S}$ is zero. The rolling conditions are constraints that – together with the motion equations – describe the motion of $\mathcal{O}$. If $\mathbf{r}_1$ and $\mathbf{r}_2$ are the vectors pointing from $\mathbf{c}$ to two distinct contact points, then by subtracting the two corresponding rolling conditions from one another we obtain the equation $\boldsymbol{\omega} \times (\mathbf{r}_1 - \mathbf{r}_2) = \mathbf{0}$. Thus, in order to satisfy all rolling conditions, all contact points must be collinear. Hence, the general case can be reduced to the case of two contact points. Note that the rolling conditions also imply that the contacts are bilateral, i.e. $\mathcal{O}$ is neither allowed to approach nor to move away from $\mathcal{S}$ in any contact point. But we will see that the situations where an attractive force would be necessary to maintain a contact are easy to detect. We start by describing the case of one single contact point.

One Contact Point

The ideas used here are similar to those in [Mac60], where the rolling motion of a sphere on a given surface is studied. Let $\mathbf{f}$ be the sum of all external forces (such as gravity) acting on $\mathcal{O}$. We assume that $\mathbf{f}$ acts on the center of mass, hence it does not cause a torque. Let $\mathbf{f}_r$ denote the reaction force of the surface acting in the contact point. Then the Newton-Euler dynamics equations have the form

$$\begin{aligned} m\dot{\boldsymbol{v}} &= \mathbf{f} + \mathbf{f}_r \quad \text{and} && (4.12)\\ \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} &= \mathbf{r} \times \mathbf{f}_r. && (4.13) \end{aligned}$$

Derivating the rolling condition with respect to time yields

$$\dot{\boldsymbol{v}} = \dot{\mathbf{r}} \times \boldsymbol{\omega} + \mathbf{r} \times \dot{\boldsymbol{\omega}}. \tag{4.14}$$

We can easily eliminate the reaction force $\mathbf{f}_r$ by multiplying (4.12) by $\mathbf{r}^\times$ and subtracting the result from (4.13). We obtain

$$m\mathbf{r} \times \dot{\mathbf{v}} - \mathbf{I}\dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} = \mathbf{r} \times \mathbf{f}. \tag{4.15}$$

Together with (4.14) we get a system of differential equations

$$\begin{bmatrix} m\mathbf{r}^\times & -\mathbf{I} \\ \mathbf{E} & -\mathbf{r}^\times \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{r} \times \mathbf{f} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \\ \dot{\mathbf{r}} \times \boldsymbol{\omega} \end{bmatrix}. \tag{4.16}$$

The right-hand side of this system contains the derivation of the vector $\mathbf{r}$ with respect to time. By the definition of $\mathbf{r}$, this derivation is $\dot{\mathbf{r}} = \dot{\mathbf{p}}_c - \mathbf{v}$, where $\mathbf{p}_c(t)$ is the contact point at time t (in contrast to $\mathbf{p}(t)$, which is fixed in the local coordinate system of $\mathcal{O}$). The point $\mathbf{p}_c$ is always a vertex or a locally closest point on an edge or a surface to the surface $\mathcal{S}$. In [ACP95] differential geometry techniques are used to compute $\dot{\mathbf{p}}_c$ as functions of the velocities $\mathbf{v}$ and $\boldsymbol{\omega}$. Hence, together with (2.32), the equations (4.16) form a system of ODEs which describes the rolling motion of $\mathcal{O}$.

Lemma 4.1. *The matrix on the left-hand side of (4.16) is regular.*

Proof. Let $\mathbf{A}$ be the matrix in question. It suffices to show that the zero vector is the only solution of the system $\mathbf{A}[\mathbf{x}^T, \mathbf{y}^T]^T = \mathbf{0}$. We write this system in the form

$$\begin{aligned} m\mathbf{r}^\times \cdot \mathbf{x} - \mathbf{I} \cdot \mathbf{y} &= \mathbf{0} \\ \mathbf{x} - \mathbf{r}^\times \cdot \mathbf{y} &= \mathbf{0}. \end{aligned}$$

If we multiply the lower equation with $m\mathbf{r}^\times$ and subtract the upper one we obtain

$$(\mathbf{I} - m\mathbf{r}^\times\mathbf{r}^\times)\mathbf{y} = \mathbf{0}.$$

If we can show that the matrix $\mathbf{I} - m\mathbf{r}^\times\mathbf{r}^\times$ is regular, it follows that $\mathbf{y} = \mathbf{0}$. Then the lower one of the above equations implies that $\mathbf{x} = \mathbf{0}$, as well, and we are done. Let $\mathbf{u} \neq \mathbf{0}$. Then,

$$\begin{aligned} \mathbf{u}^T(\mathbf{I} - m\mathbf{r}^\times\mathbf{r}^\times)\mathbf{u} &= \mathbf{u}^T\mathbf{I}\mathbf{u} - m\mathbf{u}^T(\mathbf{r} \times (\mathbf{r} \times \mathbf{u})) \\ &= \mathbf{u}^T\mathbf{I}\mathbf{u} + m(\mathbf{r}^2\mathbf{u}^2 - (\mathbf{r}^T\mathbf{u})^2) \\ &= \mathbf{u}^T\mathbf{I}\mathbf{u} + m(\mathbf{r} \times \mathbf{u})^2 \\ &> 0, \end{aligned}$$

since $\mathbf{I}$ is positive definite. Thus, the matrix $\mathbf{I} - m\mathbf{r}^\times\mathbf{r}^\times$ is positive definite and therefore regular. □

Consequently, the system of ODEs is always non-singular. This means that the motion of the object is uniquely determined.

As already mentioned, we want to identify the situations in which an attractive force would be necessary to maintain the contact. If $\mathbf{n}$ is the normal of $\mathcal{S}$ in the contact point, then these situations are characterized by $\mathbf{n}^T\mathbf{f}_r < 0$. Thus, we only have to compute the reaction force $\mathbf{f}_r$. This can be done by using equation (4.12) after we have computed $\dot{\mathbf{v}}$ and $\dot{\boldsymbol{\omega}}$ as solutions of the system (4.16).

Two Contact Points

Now suppose that $\mathcal{O}$ touches the surface in two distinct contact points $\mathbf{p}_1$ and $\mathbf{p}_2$. In order to obtain more readable equations we define the vector $\mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2 \neq \mathbf{0}$. As before, $\mathbf{f}$ is the sum of all external forces and we assume that $\mathbf{f}$ does not cause a torque. We denote the reaction forces of the surface in the two contact points by $\mathbf{f}_{r_1}$ and $\mathbf{f}_{r_2}$. The Newton-Euler dynamics equations then have the form

$$\begin{aligned} m\dot{\mathbf{v}} &= \mathbf{f} + \mathbf{f}_{r_1} + \mathbf{f}_{r_2} \quad \text{and} && (4.17)\\ \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} &= \mathbf{r}_1 \times \mathbf{f}_{r_1} + \mathbf{r}_2 \times \mathbf{f}_{r_2}. && (4.18)\end{aligned}$$

From the rolling conditions for the two contact points follows that the angular velocity is always parallel to the line between the two contact points, i.e.

$$\boldsymbol{\omega} \times \mathbf{r}_{12} = \mathbf{0}. \tag{4.19}$$

Also from the rolling conditions we obtain by differentiation w.r.t. t the two equations

$$\begin{aligned} \dot{\mathbf{v}} &= \dot{\mathbf{r}}_1 \times \boldsymbol{\omega} + \mathbf{r}_1 \times \dot{\boldsymbol{\omega}} \quad \text{and}\\ \dot{\mathbf{v}} &= \dot{\mathbf{r}}_2 \times \boldsymbol{\omega} + \mathbf{r}_2 \times \dot{\boldsymbol{\omega}}. \end{aligned} \tag{4.20}$$

As in the case of one contact point we want to eliminate the reaction forces. Therefore, we first multiply equation (4.18) by $\mathbf{r}_{12}^T$ and obtain

$$\begin{aligned} \mathbf{r}_{12}^T\mathbf{I}\dot{\boldsymbol{\omega}} &= \mathbf{r}_{12}^T(\mathbf{r}_1 \times \mathbf{f}_{r_1} + \mathbf{r}_2 \times \mathbf{f}_{r_2} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega})\\ &= \mathbf{f}_{r_1}^T(\mathbf{r}_{12} \times \mathbf{r}_1) + \mathbf{f}_{r_2}^T(\mathbf{r}_{12} \times \mathbf{r}_2) \quad \text{since } \mathbf{r}_{12} \| \boldsymbol{\omega}\\ &= (\mathbf{f}_{r_1} + \mathbf{f}_{r_2})^T(\mathbf{r}_1 \times \mathbf{r}_2). \end{aligned}$$

Now we can use equation (4.17) to replace $\mathbf{f}_{r_1} + \mathbf{f}_{r_2}$ by $m\dot{\mathbf{v}} - \mathbf{f}$. We use the upper equation of (4.20) to eliminate $\dot{\mathbf{v}}$ from the result and obtain

$$\begin{aligned} &\mathbf{r}_{12}^T\mathbf{I}\dot{\boldsymbol{\omega}} = \big(m(\dot{\mathbf{r}}_1 \times \boldsymbol{\omega} + \mathbf{r}_1 \times \dot{\boldsymbol{\omega}}) - \mathbf{f}\big)^T(\mathbf{r}_1 \times \mathbf{r}_2)\\ \Leftrightarrow \quad &\big(\mathbf{I}\mathbf{r}_{12} - m(\mathbf{r}_1 \times \mathbf{r}_2) \times \mathbf{r}_1\big)^T\dot{\boldsymbol{\omega}} = \big(m(\dot{\mathbf{r}}_1 \times \boldsymbol{\omega}) - \mathbf{f}\big)^T(\mathbf{r}_1 \times \mathbf{r}_2). \end{aligned}$$

If we use the lower equation of (4.20) to eliminate $\dot{\mathbf{v}}$ we obtain a similar result. We add both results to obtain an equation which is symmetric in the vectors $\mathbf{r}_1$ and $\mathbf{r}_2$:

$$\big(2\mathbf{I}\mathbf{r}_{12} - m(\mathbf{r}_1 \times \mathbf{r}_2) \times (\mathbf{r}_1 + \mathbf{r}_2)\big)^T\dot{\boldsymbol{\omega}} = \big(m(\dot{\mathbf{r}}_1 + \dot{\mathbf{r}}_2) \times \boldsymbol{\omega} - 2\mathbf{f}\big)^T(\mathbf{r}_1 \times \mathbf{r}_2).$$

We write this equation in the form

$$\mathbf{u}^T\dot{\boldsymbol{\omega}} = c. \tag{4.21}$$

Subtracting the two equations (4.20) from one another yields

$$\mathbf{r}_{12} \times \dot{\boldsymbol{\omega}} = \boldsymbol{\omega} \times \dot{\mathbf{r}}_{12}. \tag{4.22}$$

We combine (4.22) and (4.21) to obtain a system of equations for the vector $\dot{\omega}$. This system has the form

$$\begin{aligned} \mathbf{r}_{12} \times \dot{\omega} &= \mathbf{b} \\ \mathbf{u}^T \dot{\omega} &= c, \end{aligned} \tag{4.23}$$

where we denoted the right-hand side of (4.22) by $\mathbf{b}$ for the sake of convenience. Since ω and $\mathbf{r}_{12}$ are parallel, it holds that $\mathbf{b}^T \mathbf{r}_{12} = 0$.

Lemma 4.2. $\mathbf{u}^T \mathbf{r}_{12} \neq 0$ *and the vector*

$$\dot{\omega} = \frac{c\mathbf{r}_{12} + \mathbf{b} \times \mathbf{u}}{\mathbf{u}^T \mathbf{r}_{12}}$$

solves the system (4.23). Moreover, this system has full rank which implies that the solution is uniquely determined.

Proof. Suppose we have already shown that $\mathbf{u}^T \mathbf{r}_{12} \neq 0$. Then, we can define $\dot{\omega}$ as stated in the lemma. By inserting this into (4.23) and using $\mathbf{b}^T \mathbf{r}_{12} = 0$ we immediately verify that this vector indeed solves the system. In order to prove that the rank of the system is maximal it suffices to show that $\mathbf{y} = \mathbf{0}$ is the only solution of

$$\begin{aligned} \mathbf{r}_{12} \times \mathbf{y} &= \mathbf{0} \\ \mathbf{u}^T \mathbf{y} &= 0. \end{aligned}$$

The solution of this system is the intersection of the line through the origin with direction $\mathbf{r}_{12}$ and the plane through the origin with normal $\mathbf{u}$. Since $\mathbf{u}^T \mathbf{r}_{12} \neq 0$, this intersection consists of exactly one point, namely the origin. It remains to show that $\mathbf{u}^T \mathbf{r}_{12} \neq 0$. We have

$$\begin{aligned} \mathbf{u}^T \mathbf{r}_{12} &= 2\mathbf{r}_{12}^T \mathbf{I} \mathbf{r}_{12} - m\big((\mathbf{r}_1 \times \mathbf{r}_2) \times (\mathbf{r}_1 + \mathbf{r}_2)\big)^T \mathbf{r}_{12} \\ &= 2\mathbf{r}_{12}^T \mathbf{I} \mathbf{r}_{12} - m(\mathbf{r}_1 \times \mathbf{r}_2)^T \big((\mathbf{r}_1 + \mathbf{r}_2) \times \mathbf{r}_{12}\big) \\ &= 2\big(\mathbf{r}_{12}^T \mathbf{I} \mathbf{r}_{12} + m(\mathbf{r}_1 \times \mathbf{r}_2)^2\big). \end{aligned}$$

This expression is greater than zero, since $\mathbf{I}$ is positive definite and $\mathbf{r}_{12} \neq \mathbf{0}$. □

Thus, the vector $\dot{\omega}$ given in the lemma and the equation (4.20) and (2.32) form a non-singular system of ODEs that uniquely describe the rolling motion of the object $\mathcal{O}$.

Similarly to the case of one contact point, we want to determine the situations in which a contact has to be released because an attractive reaction force would be necessary to maintain it. Let therefore $\mathbf{n}_1$ and $\mathbf{n}_2$ be the normals of $\mathcal{S}$ in the contact points. A reaction force $\mathbf{f}_{r_i}$ is attractive if $\mathbf{n}_i^T \mathbf{f}_{r_i} < 0$. Thus, we want to compute these forces from the equations (4.17) and (4.18) after we have determined $\dot{\mathbf{v}}$ and $\dot{\omega}$. But these two equations form a system of linear equations for the reaction forces whose rank is five. Hence, $\mathbf{f}_{r_1}$ and $\mathbf{f}_{r_2}$ are not uniquely determined. But the following lemma shows that these forces are unique up to their components in the direction of $\mathbf{r}_{12}$.

Lemma 4.3. *Let the forces $\mathbf{f}_{r_1}$ and $\mathbf{f}_{r_2}$ fulfill the equations (4.17) and (4.18). The forces $\tilde{\mathbf{f}}_{r_1}$ and $\tilde{\mathbf{f}}_{r_2}$ also satisfy these equations if and only if there is a $\boldsymbol{\Delta}$ which is parallel to $\mathbf{r}_{12}$ such that $\tilde{\mathbf{f}}_{r_1} = \mathbf{f}_{r_1} + \boldsymbol{\Delta}$ and $\tilde{\mathbf{f}}_{r_2} = \mathbf{f}_{r_2} - \boldsymbol{\Delta}$.*

Proof. We define $\boldsymbol{\Delta}_i = \tilde{\mathbf{f}}_{r_i} - \mathbf{f}_{r_i}$ for $i = 1, 2$. Then from (4.17) we get

$$\begin{aligned} m\dot{\mathbf{v}} &= \mathbf{f} + \mathbf{f}_{r_1} + \mathbf{f}_{r_2} = \mathbf{f} + \mathbf{f}_{r_1} + \boldsymbol{\Delta}_1 + \mathbf{f}_{r_2} + \boldsymbol{\Delta}_2 \\ &\Leftrightarrow \boldsymbol{\Delta}_1 = -\boldsymbol{\Delta}_2 =: \boldsymbol{\Delta}. \end{aligned}$$

Moreover, (4.18) yields

$$\begin{aligned} \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} &= \mathbf{r}_1 \times \mathbf{f}_{r_1} + \mathbf{r}_2 \times \mathbf{f}_{r_2} = \mathbf{r}_1 \times (\mathbf{f}_{r_1} + \boldsymbol{\Delta}) + \mathbf{r}_2 \times (\mathbf{f}_{r_2} - \boldsymbol{\Delta}) \\ &\Leftrightarrow \mathbf{r}_{12} \times \boldsymbol{\Delta} = 0. \end{aligned}$$

□

Thus, if we choose a $\boldsymbol{\Delta}$ parallel to $\mathbf{r}_{12}$, add it to $\mathbf{f}_{r_1}$ and subtract it from $\mathbf{f}_{r_2}$, we do not change the dynamic behaviour of the object. We choose

$$\boldsymbol{\Delta} = \frac{1}{2r_{12}^2}\mathbf{r}_{12}^T(\mathbf{f}_{r_2} - \mathbf{f}_{r_1}) \cdot \mathbf{r}_{12}.$$

With this definition, we easily verify that $\mathbf{r}_{12}^T(\tilde{\mathbf{f}}_{r_1} - \tilde{\mathbf{f}}_{r_2}) = 0$. This means that the components of the reaction forces $\tilde{\mathbf{f}}_{r_1}$ and $\tilde{\mathbf{f}}_{r_2}$ in the direction of $\mathbf{r}_{12}$ are equal. We add the requirement for this equality to the Newton-Euler equations to obtain a system of linear equations that uniquely determines the reaction forces. We write this system in the form

$$\begin{aligned} \mathbf{f}_{r_1} + \mathbf{f}_{r_2} &= \mathbf{a} &(4.24)\\ \mathbf{r}_1 \times \mathbf{f}_{r_1} + \mathbf{r}_2 \times \mathbf{f}_{r_2} &= \mathbf{b} &(4.25)\\ \mathbf{r}_{12}^T(\mathbf{f}_{r_1} - \mathbf{f}_{r_2}) &= 0. &(4.26) \end{aligned}$$

We show how to solve this system for $\mathbf{f}_{r_1}$. Computing the second reaction force works analogously. First, we multiply (4.24) from the left-hand side by $\mathbf{r}_2^\times$ and subtract the result from (4.25). This gives us

$$\mathbf{r}_{12} \times \mathbf{f}_{r_1} = \mathbf{b} - \mathbf{r}_2 \times \mathbf{a}. \tag{4.27}$$

Next, we multiply (4.24) from the left-hand side by $\mathbf{r}_{12}^T$ and add the result to (4.26). This yields

$$\mathbf{r}_{12}^T\mathbf{f}_{r_1} = \frac{1}{2}\mathbf{r}_{12}^T\mathbf{a}. \tag{4.28}$$

If we multiply (4.27) from the left-hand side by $\mathbf{r}_{12}^\times$ we obtain

$$\mathbf{r}_{12}^T\mathbf{f}_{r_1} \cdot \mathbf{r}_{12} - r_{12}^2 \cdot \mathbf{f}_{r_1} = \mathbf{r}_{12} \times (\mathbf{b} - \mathbf{r}_2 \times \mathbf{a}).$$

We insert (4.28) into this and solve for $\mathbf{f}_{r_1}$. The result is

$$\mathbf{f}_{r_1} = \frac{\mathbf{r}_{12}^T\mathbf{a} \cdot \mathbf{r}_{12} - 2\mathbf{r}_{12} \times (\mathbf{b} - \mathbf{r}_2 \times \mathbf{a})}{2r_{12}^2}.$$

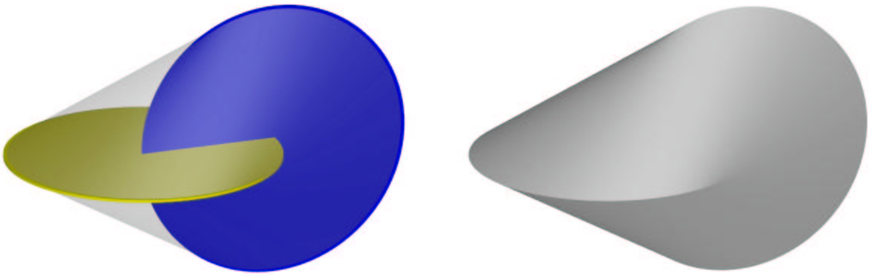


Figure 4.6: The left image shows a transparent Oloid with the two circles defining it inscribed. The right image shows the solid shaded Oloid.

An example

As an evaluation example we simulated the rolling of an Oloid[1] on an inclined plane. The Oloid is the convex hull of two circles that lie in perpendicular planes such that each of them contains the center of the other. Figure 4.6 shows the two circles defining the Oloid and the Oloid itself. In figure 4.7 you see a sequence of snapshots of the rolling motion. The blue curves are the curves of the endpoints of the line segment that touches the plane. In [DS97] the development of the bounding torse of the Oloid has been computed, so we could verify that our result coincides with the curves given there. This simulation could be done in real time on a Sun workstation with a 440 MHz processor.

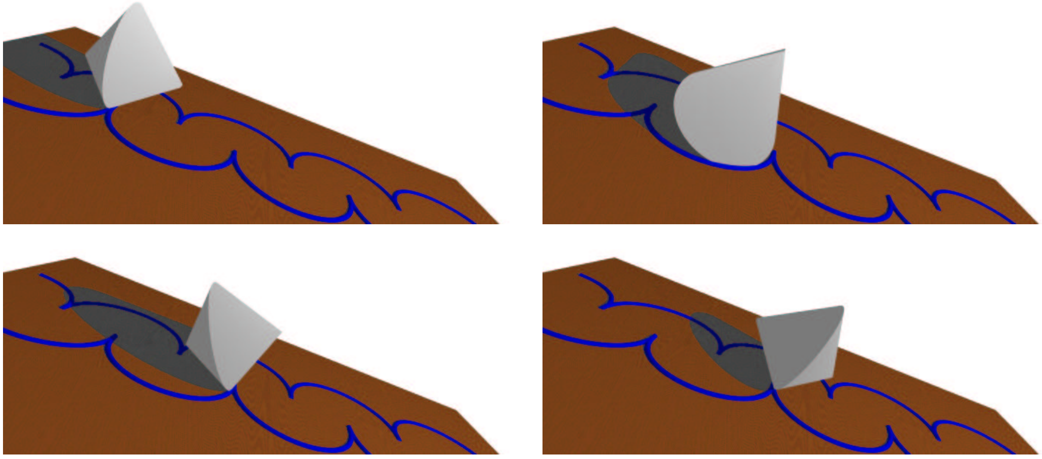


Figure 4.7: An Oloid rolling down an inclined plane and the contact curves.

[1] The Oloid was invented by Paul Schatz (1898-1979) who took a patent on it in 1933 (Deutsches Reichspatent Nr. 589 452).

5 Summary

5.1 Conclusion

We presented a new algorithm for the static collision detection problem for curved rigid objects. We described this algorithm generically and then specialized it for the special classes

- quadratic complexes,
- natural quadratic complexes plus torus and
- quadratic complexes plus quadric intersection curves.

We showed that all computations can be reduced to finding the roots of polynomials in one variable. In particular, we saw that for the first two classes the degrees of these polynomials are at most four. Hence, their roots can be computed very efficiently and accurately using Cardano's and Ferrari's formulae. For the third class of objects we showed that the degrees of the occurring polynomials are at most eight. Since one subproblem of our algorithm is the computation of the points of intersection between an edge and a face, this bound is tight. This is because a quadric intersection curve generally intersects a quadric in eight points. With our prototypical implementation of the algorithm for the class of quadratic complexes we showed by comparison with the collision detection software SWIFT++ which works on polyhedra that it is possible to implement our method in such a way that its performance is good compared to algorithms for polyhedral objects. Moreover, this comparison showed that it actually makes sense to work with the curved objects directly instead of approximating them.

We also presented a new algorithm for the dynamic collision detection problem for rigid objects with curved boundaries. We made the assumption that one object moves whereas the other one is stationary. After a generic description of the algorithm we gave specializations for the object classes

- quadratic complexes and
- natural quadratic complexes.

Again, we reduced the computational tasks to the root finding problem for univariate polynomials. We proved upper bounds for the degrees of these polynomials. In order to keep these as low as possible we made case distinctions on the specific types of curves and surfaces involved in the collision tests as well as on the types of motion. We saw that the degrees in the case of pure translations were always lower than those in the case of pure rotations which were again lower than in the case of superpositions. The case distinctions on the curve and surface types showed that it makes sense to treat the cases separately where straight lines or planes are involved.

We identified a subproblem of the dynamic collision detection which we could not solve efficiently in all cases. We called this problem the *penetration test*. The problem was the following. Given a moving object that is at time $\mathtt{t}$ in touch with a stationary object at point $\mathbf{p}$. Decide whether there is a penetration locally at $\mathbf{p}$ in the immediate future. We showed that the penetration test can be formulated as the problem to decide whether a point belongs to the boundary of a full-dimensional cell in a semi-algebraic set in $\mathbb{R}^4$. It is an open question how this problem can be solved efficiently in all cases. For non-degenerate situations, however, we presented a simple and efficient way to perform this test.

We derived an approach to simulate the rolling motion of an object on a surface. We distinguished the cases whether the object is in contact with the surface in one point or several points. In the case of several contact points we showed that rolling is only possible if all these points are collinear. Hence, we could assume that there are only two contact points. For both situations we derived a system of ordinary differential equations for which we proved non-singularity. This means that the behaviour of the object is uniquely determined by its initial velocities. We implemented this approach to simulate an oloid rolling down an inclined plane. We were able to perform this simulation in real-time.

5.2 Further Research

Since the performance of our prototypical implementation of our static collision test for the class of natural quadratic complexes compared to the software SWIFT++ was very promising, one area for future work would be an elaborate implementation of our method for all classes of objects considered in this thesis. In this context also research on more sophisticated methods for fast feature culling would be interesting. For instance, temporal and spacial coherence between two successive motion steps could be exploited as was done in [EL01] for polyhedral models. Moreover, it would be a challenging task to implement and test our dynamic collision detection algorithm. To this end it would be important to do research on how to perform the penetration test in degenerate situations, since in applications such as interactive dynamics simulations these are very likely to occur.

Moreover, it would be interesting to extend our algorithms to larger object classes. For example, in our static collision test the class of natural quadratic complexes could be extended by the torus without raising the degrees of the occurring polynomials. This was due to the fact that we could exploit that the torus is the offset surface of a simple curve, namely a circle. Hence, it is a natural question how the degrees change if our object classes are extended by offset surfaces of general conics.

Bibliography

[ACM84] Dennis S. Arnon, George E. Collins, and Scott McCallum. Cylindrical algebraic decomposition i: The basic algorithm. *SIAM Journal of Computation*, 13(4):865–889, November 1984.

[ACP95] M. Anitescu, J. F. Cremer, and F. A. Potra. Formulating 3d contact dynamics problems. Technical Report, Department of Mathematics, University of Iowa, 1995.

[Bar94] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH*, pages 174–203, July 1994.

[Ber96] Dimitri P. Bertsekas. *Constrained optimization and Lagrange mulitplier methods*. Athena Scientific, 1996.

[BS98] M. Buck and E. Schömer. Interactive rigid body manipulation with obstacle contacts. In 6^{th} *Int. Conference in Central Europe on Computer Graphics and Visualization, WSCG'98*, pages 49–56, 1998.

[BSMM93] Il'ja N. Bronstein, K.A. Semendjajew, Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch, neubearbeitung auf basis der letzten russischen ausgabe!! edition, 1993.

[Buc99] M. Buck. *Simulation interaktiv bewegter Objekte mit Hinderniskontakten*. PhD thesis, Fachbereich Informatik, Universität des Saarlandes, 1999.

[CGM91] E.-W. Chionh, R.N. Goldman, and J. R. Miller. Using multivariate resultants to find the intersection of three quadric surfaces. *ACM Transactions on Graphics*, 10(4):378–400, 1991.

[CLMP95] J. Cohen, M.C. Lin, D. Manocha, and K. Ponamgi. I-Collide: An interactive and exact collision detection system for large-scaled environments. In *Proc. ACM Interactive 3D Graphics Conference*, pages 189–196, 1995.

[CLO97] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag New York, Inc., 2nd edition, 1997.

[Cra89] J. Craig. *Introduction to Robotics, Mechanics and Control*. Addison-Wesley Publishing, 1989.

[DLLP03] Laurent Dupont, Daniel Lazard, Sylvain Lazard, and Sylvain Petitjean. Near-optimal parameterization of the intersection of quadrics. In *Proceedings of the nineteenth conference on Computational geometry*, pages 246–255. ACM Press, 2003.

[Dok97] T. Dokken. *Aspects of Intersection Algorithms and Approximation.* Ph.D. thesis, University of Oslo, July 1997.

[DS97] Hans Dirnböck and Hellmuth Stachel. The development of the oloid. *Journal for Geometry and Graphics*, 1:105–118, 1997.

[Eck99] J. Eckstein. *Echtzeitfähige Kollisionserkennung für Virtual Reality Anwendungen.* PhD thesis, Fachbereich Informatik, Universität des Saarlandes, 1999.

[EL01] S. Ehmann and M. Lin. Accurate and fast proximity queries between polyhedra using surface decomposition. In *Computer Graphics Forum (Proc. Eurographics)*, 2001.

[ES99] J. Eckstein and E. Schömer. Dynamic collision detection in virtual reality applications. In 7$^{\text{th}}$ *International Conference in Central Europe on Computer Graphics and Visualization and Interactive Digital Media*, pages 71–78, 1999.

[ES03] David Eisenbud and Frank-Olaf Schreyer. Resultants and chow forms via exterior syzygies. *Journal of the American Mathematical Society*, 16:537–579, 2003.

[FNO89] R.T. Farouki, C.A. Neff, and M.A. O'Connor. Automatic parsing of degenerate quadric-surface intersection. *ACM Transactions on Graphics*, 8(3):174–203, 1989.

[Gei02] N. Geismann. *An Exact and Efficient Approach for Computing a Cell in an Arrangement of Quadrics.* Ph.D. thesis, Universität des Saarlandes, 2002.

[GH81] Marvin J. Greenberg and John R. Harper. *Algebraic topology : a first course*, volume 58 of *Mathematical lecture notes series.* Addison-Wesley, 1981.

[GLM96] S. Gottschalk, M. C. Lin, and D. Manocha. OBB-tree: A hierarchical structure for rapid interference detection. *Computer Graphics*, pages 171–180, August 1996. Proc. SIGGRAPH'96.

[Got96] S. Gottschalk. Separating axis theorem. Technical Report TR96-024, Department of Computer Science, University of North Carolina - Chapel Hill, 1996.

[HHKR95] Rolf Hammer, Matthias Hocks, Ulrich Kulisch, and Dietmar Ratz. *C ++ toolbox for verified computing : basic numerical problems: theory, algorithms, and programs.* Springer, 1995.

[HKL$^+$99] G. Hotz, A. Kerzmann, C. Lennerz, R. Schmid, E. Schömer, and T. Warken. Silvia–a simulation library for virtual reality applications. In *procedings of IEEE Virtual Reality*, page 82, 1999.

[HKM$^+$96] M. Held, J. T. Klosowski, J. S. B. Mitchell, H. Sowizraland, and K. Zirkan. Efficicient collision detection using bounding volume hierarchies of k-DOPs. In *ACM SIGGRAPH'96 Visual Proceedings New Orleans*, August 1996.

[Hof89] C.M. Hoffmann. *Geometric and solid modeling.* Morgan Kaufmann Publishers, 1989.

[Hub95] P. M. Hubbard. Collision detection for interactive graphics applications. *IEEE Trans. on Visual. and Comput. Graph.*, 1(3):218–230, September 1995.

[Hub96] P. M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, July 1996.

[HZLM02] Kenneth E. Hoff III, Andrew Zaferakis, Ming Lin, and Dinesh Manocha. Fast 3D geometric proximity queries between rigid and deformable models using graphics hardware acceleration. Technical Report TR02-004, Department of Computer Science, UNC Chapel Hill, 2002.

[Imm01] S. Immich. Berechnung der Momente von quadratischen Komplexen. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, 2001.

[Kan96] C. Kanzow. Global convergence properties of some iterative methods for linear complementarity problems. *SIAM Journal of Optimization*, 6(2):326–341, May 1996.

[Kim98] Ku-Jin Kim. *Torus and Simple Surface Intersection Based on a Configuration Space Approach.* Ph.D. thesis, Department of Computer Science and Engineering, POSTECH, February 1998.

[KM97] Shankar Krishnan and Dinesh Manocha. An efficient surface intersection algorithm based on lower-dimensional formulation. *ACM Trans. Graph.*, 16(1):74–106, 1997.

[Kow79] H.-J. Kowalsky. *Lineare Algebra.* de Gruyter, Berlin, 1979.

[Len00] C. Lennerz. Impulsbasierte Dynamiksimulation starrer Körper unter Verwendung von Hüllkörperhierarchien. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, 2000.

[Len04] Christian Lennerz. *Distance Computation between Extended Quadratic Complexes.* PhD thesis, Max-Planck-Institut für Informatik, Saarbrücken, 2004. In preparation.

[Lev76] J. Levin. A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces. *Commun. ACM*, 19(10):555–563, 1976.

[LM95] M. C. Lin and D. Manocha. Fast interference detection between geometric models. *Visual Comput.*, 11(10):542–561, 1995.

[LRSW02] Christian Lennerz, Joachim Reichel, Elmar Schömer, and Thomas Warken. Efficient collision detection for curved solid objects. In 7^{th} *ACM Symposium on Solid Modeling and Applications*, 2002.

[Mac02] F. S. Macaulay. On some formula in elimination. In *Proc. London Mathematical Society*, pages 3–27, May 1902.

[Mac60] W. D. MacMillan. *Dynamics of Rigid Bodies*, chapter VIII, pages 257–287. Dover Publications, Inc., New York, 1960.

[Man94] Dinesh Manocha. Computing selected solutions of polynomial equations. In *Proceedings of the international symposium on Symbolic and algebraic computation*, pages 1–8. ACM Press, 1994.

[MC91] Dinesh Manocha and John Canny. A new approach for surface intersection. In *Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications*, pages 209–219. ACM Press, 1991.

[MC95] B. Mirtich and J. Canny. Impulse-based dynamic simulation. In K. Goldberg, D. Halperin, J. C. Latombe, and R. Wilson, editors, *The Algorithmic Foundations of Robotics.* A. K. Peters, Wellesley, MA, 1995.

[Mig92] Maurice Mignotte. *Mathematics for computer algebra.* Springer, 1992.

[Mir96a] B. Mirtich. Fast and accurate computation of polyhedral mass properties. *J. Graphics Tools*, 1(2):31–50, 1996.

[Mir96b] B. Mirtich. *Impulse-based dynamic simulation of rigid body systems.* PhD thesis, University of California, Berkeley, 1996.

[Mir97] B. Mirtich. Efficient algorithms for two-phase collision detection. Technical Report TR97-23, MERL, 1997.

[MK96] Dinesh Manocha and Shankar Krishnan. Solving algebraic systems using matrix computations. *SIGSAM Bull.*, 30(4):4–21, 1996.

[MS81] A. Marchetti-Spaccamela. The p-center problem in the plane is NP-complete. In *Proc. 19th Allerton Conf. Commun. Control Comput.*, pages 31–40, 1981.

[Ove92] M. H. Overmars. Point location in fat subdivisions. *Inform. Process. Lett.*, 44:261–265, 1992.

[PG95] I.J. Palmer and R.L. Grimsdale. Collision detection for animation using sphere-trees. *Proc. Eurographics*, 14(2):105–116, 1995.

[PTVF94] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C : the art of scientific computing.* Cambridge Univ. Press, 2nd edition, 1994.

[Qui94] S. Quinlan. Efficient distance computation between non-convex objects. In *Proc. Int. Conf. on Robotics and Automation*, pages 3324–3329, 1994.

[Rei01] J. Reichel. Optimale Hüllkörper für Objekte mit gekrümmten Oberflächen. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, 2001.

[RKC01] S. Redon, A. Kheddar, and S. Coquillart. Contact: Arbitrary in-between motions for collision detection. *IEEE Roman 2001*, September 2001.

[RKC02] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. In *Eurographics*, 2002.

[Sau03] J. Sauer. *Nichtholonome Mehrkörperdynamik mit Coulombscher Reibung - Ein Skalierbares Iteratives Time-Stepping Verfahren dargestellt am Beispiel physikalischer Spielzeuge.* PhD thesis, Universität des Saarlandes, 2003.

[Sch94] E. Schömer. *Interaktive Montagesimulation mit Kollisionserkennung.* PhD thesis, Fachbereich Informatik, Universität des Saarlandes, 1994.

[SKL98] R. Steffan, T. Kuhlen, and A. Loock. A virtual workplace including a multimodal computer interface for interactive assembly planning. In *IEEE Int. Conf. Intelligent Engineering Systems*, 1998.

[Sny92] John M. Snyder. Interval analysis for computer graphics. In Edwin E. Catmull, editor, *Proceedings of the 19th Annual ACM Conference on Computer Graphics and Interactive Techniques*, pages 121–130. ACM Press, July 1992.

[SS98] J. Sauer and E. Schömer. A constraint-based approach to rigid body dynamics for virtual reality applications. In *Proc. ACM Symposium on Virtual Reality Software and Technology*, pages 153–161, 1998.

[SS00] J.R. Sendra and J. Sendra. Rationality analysis and direct parameterization of generalized offsets to quadrics. In *Applicable Algebra in Engineering, Communication and Computing*, volume 11, pages 111–139, 2000.

[SSW95] E. Schömer, J. Sellen, and M. Welsch. Exact geometric collision detection. In *Proc.* 7^{th} *Canadian Conference on Computational Geometry*, pages 211–216, 1995.

[ST95a] E. Schömer and C. Thiel. Efficient collision detection for moving polyhedra. In *Proc.* 11^{th} *Annual ACM Symposium on Computational Geometry*, pages 51–60, 1995.

[ST95b] D. E. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, submitted 1995.

[ST96] E. Schömer and C. Thiel. Subquadratic algorithms for the general collision detection problem. In 12$^{\text{th}}$ *European Workshop on Computational Geometry*, pages 95–101, 1996.

[SWF$^+$93] J.M. Snyder, A.R. Woodbury, K. Fleischer, B. Currin, and A.H. Barr. Interval methods for multi-point collisions between time-dependent curved surfaces. In *Computer Graphics Proceedings, Annual Conference Series*, pages 321–334, 1993.

[Uhl76] F. Uhlig. A canonical form for a pair of real symmetric matrices that generate a nonsingular pencil. *Linear Algebra and its Applications*, 14:198–209, 1976.

[vHBZ90] Brian von Herzen, Alan H. Barr, and Harold R. Zatz. Geometric collisions for time-dependent parametric surfaces. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 39–48. ACM Press, 1990.

[War99] T. Warken. Berechnung von Kontaktkräften für eine zwangsbasierte Dynamiksimulation. Master's thesis, Fachbereich Informatik, Universität des Saarlandes, 1999.

[WS01] T. Warken and E. Schömer. Rolling rigid objects. In 9$^{\text{th}}$ *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'01*, volume 1, pages 57–62, 2001.

[WWK01] W. Wang, J. Wang, and M.-S. Kim. An algebraic condition for the separation of two ellipsoids. In *Computer Aided Geometric Design*, volume 18, pages 531–539, 2001.

[Zac97] G. Zachmann. Real-time and exact collision detection for interactive virtual prototyping. In *Proc. ASME Design Engineering Technical Conferences*, 1997.

[ZF95] G. Zachmann and W. Felger. The BoxTree: Enabling real time and exact collision detection of arbitrary polyhedra. In 1$^{\text{st}}$ *Workshop on Simulation and Interaction in Virtual Environments*, pages 104–113, 1995.