

Spring 2005

Theory and algorithms for swept manifold intersections

Yuriy Mileyko

New Jersey Institute of Technology

Follow this and additional works at: <https://digitalcommons.njit.edu/dissertations>



Part of the [Mathematics Commons](#)

Recommended Citation

Mileyko, Yuriy, "Theory and algorithms for swept manifold intersections" (2005). *Dissertations*. 707.
<https://digitalcommons.njit.edu/dissertations/707>

This Dissertation is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Dissertations by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen



The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

THEORY AND ALGORITHMS FOR SWEEP MANIFOLD INTERSECTIONS

by
Yuriy Mileyko

Recent developments in such fields as computer aided geometric design, geometric modeling, and computational topology have generated a spate of interest towards geometric objects called swept volumes. Besides their great applicability in various practical areas, the mere geometry and topology of these entities make them a perfect testbed for novel approaches aimed at analyzing and representing geometric objects. The structure of swept volumes reveals that it is also important to focus on a little simpler, although a very similar type of objects – swept manifolds. In particular, effective computability of swept manifold intersections is of major concern.

The main goal of this dissertation is to conduct a study of swept manifolds and, based on the findings, develop methods for computing swept surface intersections. The twofold nature of this goal prompted a division of the work into two distinct parts. At first, a theoretical analysis of swept manifolds is performed, providing a better insight into the topological structure of swept manifolds and unveiling several important properties. In the course of the investigation, several subclasses of swept manifolds are introduced; in particular, attention is focused on regular and critical swept manifolds. Because of the high applicability, additional effort is put into analysis of two-dimensional swept manifolds – swept surfaces. Some of the valuable properties exhibited by such surfaces are generalized to higher dimensions by introducing yet another class of swept manifolds – recursive swept manifolds.

In the second part of this work, algorithms for finding swept surface intersections are developed. The need for such algorithms is necessitated by a specific structure of swept surfaces that precludes direct employment of existing intersection

methods. The new algorithms are designed by utilizing the underlying ideas of existing intersection techniques and making necessary technical modifications. Such modifications are achieved by employing properties of swept surfaces obtained in the course of the theoretical study.

The intersection problems is also considered from a little different prospective. A novel, homology based approach to local characterization of intersections of submanifolds and s -subvarieties of a Euclidean space is presented. It provides a method for distinguishing between transverse and tangential intersection points and determining, in some cases, whether the intersection point belongs to a boundary.

At the end, several possible applications of the obtained results are described, including virtual sculpting and modeling of heterogeneous materials.

**THEORY AND ALGORITHMS FOR SWEEP MANIFOLD
INTERSECTIONS**

by
Yuriy Mileyko

**A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology and
Rutgers, The State University of New Jersey – Newark
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Mathematical Sciences**

**Department of Mathematical Sciences
Department of Mathematics and Computer Science, Rutgers-Newark**

May 2005

Copyright © 2005 by Yuriy Mileyko

ALL RIGHTS RESERVED

APPROVAL PAGE

**THEORY AND ALGORITHMS FOR SWEEP MANIFOLD
INTERSECTIONS**

Yuriy Mileyko

Dr. Denis Blackmore, Dissertation Advisor
Professor of Mathematical Sciences, NJIT

Date

Dr. Vladislav Goldberg, Committee Member
Distinguished Professor of Mathematical Sciences, NJIT

Date

Dr. Frank Shih, Committee Member
Professor of Computer Science, NJIT

Date

Dr. Amitabha Bose, Committee Member
Associate Professor of Mathematical Sciences, NJIT

Date

Dr. Roy Goodman, Committee Member
Assistant Professor of Mathematical Sciences, NJIT

Date

BIOGRAPHICAL SKETCH

Author: Yuriy Mileyko
Degree: Doctor of Philosophy
Date: May 2005

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mathematical Sciences,
New Jersey Institute of Technology, Newark, NJ, 2005
- Specialist in Applied Mathematics,
National Taras Shevchenko Kyiv University, Kyiv, Ukraine, 2001
- Bachelor of Science in Applied Mathematics,
National Taras Shevchenko Kyiv University, Kyiv, Ukraine, 2000

Major: Mathematical Sciences

Presentations and Publications:

- D. Blackmore, Y. Mileyko, M.C. Leu, W.C. Regli, and W. Sun, "Computational topology and swept volumes," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2005.
- B.V. Rublev and Y. Mileyko, "Construction of the smallest enclosing ellipse and the smallest enclosing ellipsoid for a finite set of points in m -dimensional euclidian space," *Proc. of 5th Ukrainian Intl. Conf. 'UkrOBRAZ'*, Kyiv, pp.79-82, 2000
- B.V. Rublev and Y. Mileyko, "An algorithm for constructing the smallest enclosing ellipsoid for a finite set of points in \mathbf{R}^n ," *Proc. of 8th Intl. Conf. in honor of M. Kravchuk*, Kyiv, p.328, 2000
- Y. Mileyko and B.V. Rublev and Y.I. Petunin "Geometric properties of the smallest enclosing ellipse and some related questions," *Kyiv University Publisher*, Kyiv, 74p., 2000
- B.V. Rublev and O.I. Molodsov and Y. Mileyko, "Fast method for constructing Fisher linear discriminant functions," *Kyiv University Bulletin*, Vol. 4, pp. 217-222, 1999

- B.V. Rublev and Y. Mileyko, "Construction of the smallest enclosing ellipsoid for elementary polyhedra," *Kyiv University Bulletin*, Vol. 3, pp. 236-242, 1999
- Y.I. Petunin and B.V. Rublev and Y. Mileyko, "An optimal algorithm for constructing the smallest enclosing ellipse in a finite number of steps," *Kyiv University Bulletin*, Vol. 3, pp. 87-95, 1998
- Y. Mileyko, "Differential Equation Approach to Manifold Intersections," *SIAM Conference on Geometric Design and Computing*, Seattle, Washington, Nov. 10-13, 2003.

To the Memory of Vira Mileyko

ACKNOWLEDGMENT

Dealing with a big project, it is usually the very last piece of work that requires too much, as we think to ourselves, effort and concentration. And it is not because of the real difficulty of that piece, but rather due to a worn out state of our mind after many laborious hours spent in trying to break the unyielding walls surrounding the main goal. Therefore, I felt a little uneasy leaving this section to be the very last one, but now it seems to me that the only thing I should worry about is being able to finish my ‘speech’ before they ‘turn on the music’.

I would like to thank my advisor, Professor Denis Blackmore. He has been a great inspiration, and his constant support and encouragement helped me a lot during the tough years of graduate studies. His way of advising was just perfect for me, since he managed to point me in the right direction without restricting my thoughts and my methods, and for that I am very grateful. He was always available to discuss my questions, and those discussions helped me to sharpen my mathematical intuition and made me a more mathematically mature person. In addition to his advising skills, Professor Blackmore is a wonderful conversationalist and a talented writer. I always enjoyed talking to him, and the progress in my technical writing should be credited entirely to him.

I would also like to thank the other members of my committee: Dr. Amitabha Bose, Dr. Vladislav Goldberg, Dr. Roy Goodman, and Dr. Frank Shih. Their suggestions and comments were very helpful, and I greatly appreciate their input.

Special thanks are due to Professor Daljit Ahluwalia for his continuing support throughout my stay at NJIT. He was always ready to help me with my problems, and he always did it in the most efficient way.

Thanks to the office staff: Susan Sutton, Padma Gulati, Sherri Brown, and Liliana Boland. Their help with the administrative routine made my life considerably easier.

Speaking about making life easier, I would like to express my sincere gratitude to Filippo Posta, Tsezar Seman, and Ivan Zorych. I am very lucky to have their friendship. Without Ivan, I would probably spend the first half a year at NJIT trying to figure out what is what and who is who. With his help, I adapted myself to the new environment in just several weeks; having him around made my first months at NJIT a smooth ride. In addition to being helpful, Ivan is just a great fellow whose company I have always enjoyed. After meeting Filippo and Tsezar, my life at NJIT has become much more lively and entertaining. This dissertation would become twice as long if I just briefly mentioned all the great moments we had together. I do not even know how I would get through my graduate years without these guys. So, once again my friends, thanks a lot!

Thanks are also due to my friends Nebo Murisic and Jyoti Champanerkar for their encouragement and support, and to my good old friend Dr. Bogdan Rublyov. He has been a great teacher for me, and his upbeat attitude and enthusiasm have always made me feel better.

It is hard for me to find appropriate words to thank Dr. Roman Voronka for his warm support and his indispensable advice. He helped me during my trying first months in the U.S. without even knowing me personally. I will always be in his debt, and he will always have my deference and heartfelt gratefulness.

Finally, I want to express my deepest gratitude to my family for their continuing support, goodwill, and love.

The research in this work has been partially supported by NSF/DARPA CARGO Grant CCR-0310619.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
2 BACKGROUND INFORMATION	9
2.1 Spaces and Maps	9
2.2 Manifolds and S -varieties	16
2.3 Vector Fields and Flows	18
2.4 Morse Theory	20
2.5 Homology Theory	22
2.5.1 Simplices and Singular Simplices	22
2.5.2 Chains, Boundary Maps and Homology Groups	23
2.5.3 Computing Homology: Exact Sequences	24
3 SWEPT MANIFOLDS	27
3.1 Overview of Sweeps and Swept Manifolds	27
3.2 Introduction to Swept Manifolds	32
3.2.1 Basic Properties of Swept Manifolds	34
3.2.2 Critical Swept Manifolds	38
3.2.3 Swept Surfaces	42
3.2.4 Recursive Swept Manifolds	48
3.3 Smooth Manifolds vs. Swept Manifolds	53
4 INTERSECTION PROBLEM	58
4.1 Overview of the Surface-to-Surface Intersection Problem	59
4.1.1 Intersection Methods	61
4.1.2 Geometric and Topological Consistency of Intersection Algorithms	67
4.2 Swept Manifold Intersections	70
4.2.1 Rendering Swept Surfaces	71
4.2.2 Simple Discretization Algorithm	76

TABLE OF CONTENTS
(Continued)

Chapter	Page
4.2.3 Subdivision Algorithm	78
4.2.4 Marching Algorithm	87
4.2.5 Intersections of Restricted Swept Surfaces	93
4.3 Homological Intersection Detection	95
4.3.1 Manifolds with Boundary	102
5 APPLICATIONS	105
5.1 Virtual Sculpting	105
5.2 Tissue Engineering	107
5.3 Homological Intersection Characterization	110
6 CONCLUSIONS AND FUTURE RESEARCH	113
REFERENCES	118

LIST OF FIGURES

Figure	Page
1.1 A ‘curvy’ parallelepiped.	3
1.2 An example of swept volume.	5
3.1 Swept volume of a ball.	28
3.2 Subdivision of a generalized swept manifold into two intersecting manifolds.	33
3.3 A sphere and its diffeomorph.	35
3.4 Regular swept surface of a line segment.	36
3.5 Non-regular swept surface of a line segment.	36
3.6 Generalized critical swept surface of a line segment.	39
3.7 Non-periodical critical swept surface of a line segment.	40
3.8 Periodic critical swept surface of a circle.	41
3.9 Examples of knotted curves.	43
3.10 A torus as a swept surface.	47
3.11 Some trajectories of Hopf foliation of S^3	47
3.12 Possible topological configurations of non-periodic critical swept surfaces.	48
3.13 Some of the possible topological configurations of periodic critical swept surfaces.	49
3.14 A regular swept surface that does not admit the coordinate functions as Morse functions without critical points.	55
4.1 Tracing an intersection curve around a tangential intersection point. . .	65
4.2 Example of geometrically close but topologically different curves. . . .	67
4.3 Example of topologically equivalent but geometrically different curves. .	68
4.4 Well rendered self-intersecting generalized swept surface.	71
4.5 Badly rendered self-intersecting generalized swept surface.	72
4.6 Two approaches to rendering periodic swept surfaces.	74
4.7 Intersecting surfaces.	79

LIST OF FIGURES
(Continued)

Figure	Page
4.8 Example of employment of bounding boxes for discarding trivially disjoint parts of surfaces.	79
4.9 Subdivision of a swept patch.	81
4.10 ‘Good’ and ‘bad’ intersections for the subdivision algorithm.	83
4.11 Transverse intersection two surfaces.	96
4.12 Tangentially intersecting surfaces.	102
4.13 Intersection set of the surfaces	102
5.1 Virtual reality sculpting system.	106
5.2 Intersection of a tool with a workpiece as the swept surface intersection problem.	107
5.3 Simplified picture of internal bone morphology.	108

CHAPTER 1

INTRODUCTION

Having dealt with the intersection problem for a while, one may develop a habit of quickly skimming through an introduction of every other ‘intersection’ paper, without paying much attention to sometimes worthwhile descriptions of possible applications of the surface-to-surface intersection problem and completely ignoring parts lauding its importance. Why? Because the amount of attention paid to the intersection problem over recent years generated so many papers on different intersection techniques that most phrases about the importance of the problem became mantras which can be learned by heart after just a few repetitions. For the same reason, it is virtually impossible to describe the work related to the intersection problem without repeating already known and often hackneyed words and phrases. Having said this, we still venture to present the intersection problem from a little different point of view, hoping that emphasis on the new class of objects involved in intersections will make it less trite. These objects are swept manifolds. They represent a new concept that has been conceived as a simplified version of the more general notion of a swept volume ([14]). It should be understood that our choice of swept manifolds for the intersection problem was not arbitrary. In fact, it was our work on swept volumes that eventually extended one of its branches into the realm of intersection methods. Unfortunately, or perhaps fortunately, the setup of the occurring intersections was not quite conventional, which necessitated adjustments to existing intersection techniques and ultimately resulted in the development of intersection algorithms for an entirely new class of surfaces. Taking such a route towards the intersection problem stimulated us to focus more on the structure of intersecting objects, shifting some attention from the intersections per se. Interestingly, evolution of extant intersection

methods was also largely determined by the class of objects under consideration. The choice of such objects was, and still is, notably effected by the developments in the field of computer aided geometric design – a discipline concerned with design and implementation of methods for analysis and graphical representation of geometric objects.

The intersection problem and computer aided geometric design have a long and fascinating common history that can be traced as far back as 1960s, when such companies as Renault and Boeing started investigating a new approach to automobile and airplane design. A nice review of the history of curves and surfaces in computer aided geometric design is given in [32]. It would be fair to say, though, that a real outburst of interest towards the surface-to-surface intersection problem occurred after boundary representation (B-rep) became one of the major tools for modeling solids. The basic idea of B-rep is quite simple: an n -dimensional solid is completely determined by its boundary, which, in its turn, is determined by its own boundary, and so on. This leads to a recursive representation of a solid that involves lists of boundary elements of the solid in every dimension. Each boundary element is usually assumed to be smooth, implying that solids with a non-smooth boundary have to be somehow subdivided into smooth components. Interestingly, it turns out that finding such a subdivision boils down to a number of intersection problems. To see this, consider the following simple example. The boundary of a 3-dimensional parallelepiped consists of six rectangles (six smooth components). Each of these rectangles is defined by the plane in which it lies and the lines of intersection with four other rectangles. Notice that the intersections of these lines determine vertices of the parallelepiped. Thus, the boundary of the parallelepiped can be found by computing intersections of six planes and sixteen lines. It should be noted that the latter line intersection problem can be efficiently absorbed into the former plane intersection problem. Obviously, the above toy problem is not hard to solve, but imagine that we want to create a 'curvy'

parallelepiped, as shown in Figure 1.1. Suppose we know all the surfaces that should represent faces of the parallelepiped. Then finding the boundary implies computing intersections of several nonlinear surfaces, which is by no means an easy problem.



Figure 1.1 A ‘curvy’ parallelepiped.

Linear B-rep models were, and still are, quite popular due to their simplicity and efficiency, but ever growing demands coming from practice created a need for more robust models that would be able to handle nonlinearities naturally. Obviously, using general nonlinear surfaces would be too impractical, but algebraic surfaces of some specific form seemed promising and eventually established strong dominance in the field. There are quite a few classes of such surfaces; for example, Coons patches, or Bezier patches, but the center stage clearly belongs to Non-Uniform Rational B-Splines (NURBS). Applications of these types of surfaces to geometric modeling have been analyzed in many texts. For example, a classical work of de Boor [23] contains a thorough presentation of the general spline theory, emphasizing its geometric aspects, while Farin [30] focuses on Bezier and Coon’s patches. A nice introduction to NURBS is given in [31]; a more comprehensive study of NURBS and their applications is given in [80]. Specific and condensed accounts of the topic can be found in [29, 28, 82, 33]. The popularity of NURBS and the other algebraic surfaces in B-rep models gener-

ated a strong interest in the corresponding intersection problems – many intersection methods have been designed specifically for one particular class of algebraic surfaces. For example, Lasser [57], Aziz et al. [5], and Peng [78] present intersection algorithms for Bezier surfaces, Piegl [81] provides a method for finding intersections of quadrics and extruded surfaces, Meyer [66] focuses on quadratic surfaces, Kreizis and Patrikalakis [53] explore ideas that allow to evaluate intersections of rational polynomial surfaces, and Manocha et al. [62] give an intersection method for NURBS and algebraic surfaces. With time, most intersection algorithms became more advanced and could handle fairly general surfaces. For instance, in [6, 8, 9, 7, 22, 48, 92, 77, 59] methods for computing intersections of general parametric surfaces are described, and the algorithm developed by Grandine and Klein [39], which is known as Grandine-Klein intersector, became a standard for CAD/CAM libraries. Still, the shift of focus towards NURBS and similar patches is quite noticeable. In fact, the majority of intersection methods suffer a significant fall off in efficiency when applied to non-algebraic surfaces.

One might argue that the described deficiencies of extant intersection algorithms are a little contrived. Indeed, as long as NURBS-like patches prevail in geometric modeling there seems to be no much need for sophisticated generalizations. Thus, the following question arises: Is the dominance of algebraic surfaces in computer aided geometric design and related disciplines strong enough to render more general models practically unnecessary? As one might have expected, the answer to this question is ‘no’ – there are many cases when the standard algebraic techniques do not work well enough. In fact, one such class of examples is provided by swept volumes. A swept volume is a trajectory of a moving and possibly deforming object in a space, as illustrated in Figure 1.2. We shall provide a more detailed exposition of swept volumes in the sequel; for our purposes here, the above intuitive definition is sufficient. Suppose we want to construct a computer model of a swept volume.



Figure 1.2 An example of swept volume.

The law of motion should allow us to compute the image of the initial object at each time. Thus, one possible approach would be to create a B-rep model based on NURBS, or even linear approximations, by incrementally computing the boundary of the swept volume. Notice that possible self-intersections could be handled by applying already known methods. While at first glance such an idea seem to work fine, there is a significant drawback – once the model is constructed, it is no longer a swept volume since the information about it being swept is lost. Furthermore, the computed approximation of the swept volume may not be accurate enough to capture features crucial for some geometric operations that one may need to perform later. If the initial representation of the swept volume is discarded, a new approximation simply cannot be constructed. Therefore, it becomes necessary to keep the original representation of a swept volume and compute its B-rep models dynamically, as the need for some geometric operation arises. In general, such an approach would be very inefficient, making practical applications infeasible, and it seems more reasonable to look into the possibility of handling swept volumes by using its original representation directly. As shown in [19] and [15], this representation provides a lot of information about the geometric as well as topological structure of a swept volume. But when applied straightforwardly, it may not meet expected efficiency requirements, primar-

ily because it is based on a volumetric description. In an attempt to rectify this problem, a modified representation of a swept volume was proposed ([16]). It uses the fact that swept volumes are determined by their boundaries and significantly reduces the amount of computations by decreasing dimensionality. Unfortunately, the level of sophistication of this method turned out to be too high and prevented its wide spread use by practitioners. Still, the idea of focusing directly on the boundary of a swept volume seems to be promising, and in this work we describe methods that may be employed to create analogues of B-reps suited for swept volumes. As in a regular B-rep model, parts of the boundary of a swept volume can be regarded as separate entities – swept manifolds. The reduced dimensionality of such entities (in a 3-dimensional space they would be 2-dimensional surfaces) may allow us to perform geometric operations more efficiently. Moreover, similar to regular B-reps, effective methods for computing intersections of swept manifolds would provide us with almost all the tools needed for constructing boundary representations of swept volumes.

Although the connection between swept manifolds and swept volumes is very close and, furthermore, swept manifolds can be regarded as a special case of swept volumes, practical employment of these two classes of objects may be completely unrelated. As we shall see, swept manifolds have a much broader range of applications than just the one described above. In fact, because of its unconventional setup, considering the swept manifold intersection problem alone may provide new ideas for the general intersection theory. In spite of the existing favoritism towards algebraic surfaces, techniques within the algorithmic facet of intersection theory have been constantly evolving, becoming more general and sophisticated. There has been a constant inflow of ideas from geometric topology in an endeavor to create effective methods for correctly resolving the topology of an intersection set, which became a crucial part of many existing intersection algorithms. A sample of intersection research in a more topological vein includes the work of Farouki et al. [36] on developing algorithms for

piecewise linear, topologically consistent representations self-intersections, the use of index theories for vector fields to characterize intersections in Kreizis et al. [54], and the development by Peters and his collaborators (see e.g. [69]) of error bounds on differential geometric surface characteristics that insure the correct ambient isotopy type of their intersection sets. This strong topological influence on computational techniques eventually led to the creation of a new field of inquiry – computational topology – which can be considered as an amalgam of elements of computer aided geometric design and geometric topology ([17, 10, 25, 93]). It turns out that swept manifolds and, obviously, swept volumes provide a paradigm for computational topology, and we believe that studying swept manifolds and their intersections will not only enrich the nascent discipline with new ideas, but also help in testing and assessing its existing methods and techniques.

Talking about computational topology it is impossible not to mention a successful employment of homology theory in several computational problems ([49, 27]). Homology is a powerful tool for analyzing shapes of topological spaces, and making it algorithmically tractable ([49, 24]) significantly strengthens our computational abilities. As a part of this work, we try to investigate how homology theory can be used to perform local intersection analysis. Because of its generality, this approach is applicable to a very wide class of topological spaces, including general smooth manifolds and s -varieties.

This dissertation is organized as follows. In Chapter 2, we provide an overview of basic mathematical concepts that are extensively employed in the sequel. It includes elements of differential topology, algebraic topology, and dynamical systems, along with references that give more detailed accounts on the standard mathematical nomenclature that we use. Chapter 3 comprises a theoretical part of this work and presents a rather detailed analysis of swept manifolds. The notion of a swept manifold is introduced after giving a rigorous mathematical definition of a swept volume and

stating several important results relevant to our study. The introductory part of the chapter is followed by a thorough investigation of properties of swept manifolds under different assumptions and restrictions. Results obtained for general swept manifolds are later refined and improved for the case of swept surfaces. We switch to more practical issues in Chapter 5, which contains descriptions of several algorithms for finding intersections of swept surfaces. At the beginning of the chapter, we provide a short overview of existing types of intersection methods. Then we modify some basic ideas on which these methods are based in order to devise our own intersection algorithms, which are well suited for swept surfaces. The description of each of the three algorithms that we present includes a fairly deep analysis of its complexity and a discussion of possible advantages and drawbacks. The last section of the chapter changes the topic to a local analysis of intersections based on homology theory and presents several useful theorems that allow us to interrogate possibly non-transversal intersections of submanifolds and s -subvarieties of a Euclidean space. Some of the real-world applications of the results obtained in this work are illustrated in Chapter 6. Finally, in Chapter 7, we summarize the contributions of this dissertation and discuss possible future research directions.

Throughout the text, most of the results and concepts are illustrated by pictures that were generated using the Swept Surface software developed by the author. The software is written in C++ programming language and employs OpenGL graphic library for low-level management of 3-dimensional objects. At the higher-levels, geometric objects are managed using implementation of the main ideas presented in this work.

CHAPTER 2

BACKGROUND INFORMATION

This work incorporates elements from several different areas of mathematics, including differential topology, dynamical systems, and algebraic topology. In this chapter, we present a brief description of basic mathematical notions that will be extensively used in the sequel. Additional information on the topics can be obtained from the references provided throughout the discussion.

2.1 Spaces and Maps

In general, a spaces can be thought of as a set endowed with some additional structure.

We shall start with the definition of a topological space.

Definition 2.1.1 *Let X be a set and \mathcal{T} a collection of its subsets. \mathcal{T} is called a topology on X if it satisfies the following conditions:*

1. $A, B \in \mathcal{T}$ implies $A \cap B \in \mathcal{T}$;
2. $A_i \in \mathcal{T}$, for $i \in \mathcal{I}$, an index set, implies $\cup_{i \in \mathcal{I}} A_i \in \mathcal{T}$;
3. $\emptyset, X \in \mathcal{T}$.

An element of a topology is called an *open set*; the complement of an open set is a *closed set*. Notice that the first condition implies that a finite intersection of open sets is open, while the second condition means that any union of open sets is open.

Definition 2.1.2 *Let X be a set and \mathcal{T} a topology on X . A pair (X, \mathcal{T}) is called a topological space.*

If (X, \mathcal{T}) is a topological space and $S \subset X$ is a subset, we can define topology on S by $\mathcal{T}_S = \{S \cap U | U \in \mathcal{T}\}$; this topology is called an induced topology. Thus, every subset of a topological space can be regarded as a topological space in itself.

Considering topological spaces, it is always useful to define the notion of a neighborhood.

Definition 2.1.3 *Let (X, \mathcal{T}) be a topological space, and let $S \subset X$. A set $U \subset X$ is a neighborhood of S if there is an open set O such that $S \subset O \subset U$.*

If $S = \{x\}$, a one element set, we obtain a definition of a neighborhood of a point.

In general, topological spaces can have a very complicated structure with counterintuitive properties. If one would rather that such properties be eliminated, additional restrictions should be imposed.

Definition 2.1.4 *A topological space, X , is called Hausdorff if any two distinct points have disjoint neighborhoods, that is, for any $x, y \in X$ such that $x \neq y$ there are two open sets U and V such that $x \in U$, $y \in V$, and $U \cap V = \emptyset$.*

Definition 2.1.5 *A topological space, X , is called normal if any two disjoint closed sets have disjoint neighborhoods, that is, for any closed sets $A, B \subset X$ such that $A \cap B = \emptyset$ there are two open sets U and V such that $A \subset U$, $B \subset V$, and $U \cap V = \emptyset$.*

Definition 2.1.6 *Let X be a topological space. A family of open sets, $\mathcal{U} = \{U_\lambda\}_{\lambda \in \Lambda}$, is called an open cover of X if $X \subset \cup_{\lambda \in \Lambda} U_\lambda$. An open cover, \mathcal{U} , is called locally finite if every point of X meets only a finite number of sets in \mathcal{U} , that is, for every $x \in X$ there are sets $U_{\lambda_1}, \dots, U_{\lambda_k} \in \mathcal{U}$ such that $x \notin U_\lambda$ for $\lambda \neq \lambda_j$, $1 \leq j \leq k$.*

Definition 2.1.7 *Let X be a topological space and $\mathcal{U} = \{U_\lambda\}_{\lambda \in \Lambda}$ an open cover. An open cover $\mathcal{V} = \{V_\gamma\}_{\gamma \in \Gamma}$ is called a refinement of \mathcal{U} if for every $V_\gamma \in \mathcal{V}$ there is a set $U_\lambda \in \mathcal{U}$ such that $V_\gamma \subset U_\lambda$.*

Definition 2.1.8 *A topological space X is called paracompact if every open cover has a locally finite refinement.*

There is an important relation between paracompact, Hausdorff and normal spaces.

Theorem 2.1.9 *A paracompact Hausdorff space is normal.*

Very often instead of defining a topology on a set we can define a 'distance' between elements of the set. In this case we obtain a *metric space*.

Definition 2.1.10 *A map $d : X \times X \rightarrow \mathbb{R}$ is a metric if it satisfies the following conditions:*

1. $d(x, y) \geq 0$ for all $x, y \in X$, and $d(x, y) = 0$ if and only if $x = y$;
2. $d(x, y) = d(y, x)$ for all $x, y \in X$;
3. $d(x, z) \leq d(x, y) + d(y, z)$, for all $x, y, z \in X$

Definition 2.1.11 *Let X be a set and d be a metric defined on X . A pair (X, d) is called a metric space.*

It is clear that every subset of a metric space is a metric space in itself. To show how a metric defines a topology on X , we need the following definition.

Definition 2.1.12 *Let (X, d) be a metric space. Suppose $x \in X$ and $r \in \mathbb{R}$, $r > 0$. An open ball in X with center x and radius r is the set $B_{r,x} = \{y \in X \mid d(x, y) < r\}$.*

The topology on X induced by the metric d is defined as the collection of all possible unions of open balls.

In this work we are mainly concerned with subsets of specific metric spaces – Euclidean spaces. An n -dimensional Euclidean space, \mathbb{R}^n , consists of n -tuples, $\mathbf{x} = (x_1, \dots, x_n)$, where $x_k \in \mathbb{R}$, $1 \leq k \leq n$, and has the *standard* metric defined by

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}.$$

If fact, a Euclidean space is a *vector space* over real numbers, which means that its elements can be added together and multiplied by a real scalar, subject to usual associative, distributive and commutative laws. For n -tuples, addition and multiplication

by a scalar are defined elementwise. The *standard* topology of \mathbb{R}^n is the topology induced by the standard metric. The previous result about a topology induced by a metric implies that the standard topology on \mathbb{R}^1 is determined by open intervals; in the case of \mathbb{R}^2 and \mathbb{R}^3 the standard topology is defined by open disks and balls, respectively.

Analysis of different spaces would be impossible without the notion of a map. Let us remind a reader that, intuitively, a map f between two sets X and Y is a correspondence between elements of X and Y such that each $x \in X$ has a unique element $f(x) \in Y$ corresponding to it. More rigorously, a map f is an ordered triplet (X, Y, Γ) , where $\Gamma \subset X \times Y$ is such that $\{x \in X | (x, y) \in \Gamma\} = X$ and for any $(x_1, y_1), (x_2, y_2) \in \Gamma$, $x_1 = x_2$ implies $y_1 = y_2$. If also $y_1 = y_2$ implies $x_1 = x_2$, then the map f is called *one-to-one*, or *injective*. The *image* of a set $A \subset X$ under the map f is the set $f(A) = \{y \in Y | y = f(x), x \in A\}$. The set $f(X)$ is called the *range* of f , and if $f(X) = Y$ then f is called *surjective*. A one-to-one, surjective map is called *bijective*. If a map $f = (X, Y, \Gamma)$ is one-to-one, we can define the *inverse* map, $f^{-1} : Y \rightarrow X$, by $f^{-1} = (f(X), X, \Gamma^{-1})$, where $\Gamma^{-1} = \{(y, x) | (x, y) \in \Gamma\}$. The *inverse image*, or *preimage*, of a set $B \in Y$ is the set $f^{-1}(B) = \{x \in X | \exists y \in B \text{ such that } y = f(x)\}$. If we want to restrict f to some subset $R \in X$, the corresponding map, (R, Y, Γ_R) , where $\Gamma_R = \{(x, y) \in \Gamma | x \in R\}$, is denoted by $f|_R$.

Since spaces are sets with an additional structure, maps between spaces exhibit some additional properties. One of the most important properties of maps between topological spaces is continuity.

Definition 2.1.13 *Let (X, \mathcal{T}_1) and (Y, \mathcal{T}_2) be topological spaces. A map $f : X \rightarrow Y$ is continuous if $f^{-1}(V) \in \mathcal{T}_1$ for any $V \in \mathcal{T}_2$, that is, the inverse image of any open set is open.*

Notice that we would obtain an equivalent definition if we used closed sets instead of open sets.

Definition 2.1.14 Let (X, \mathcal{T}_1) and (Y, \mathcal{T}_2) be topological spaces. A map $f : X \rightarrow Y$ is continuous at a point $x \in X$ if for any neighborhood $N_x \subset X$ of x there is a neighborhood $N_y \subset Y$ of $y = f(x)$ such that $f^{-1}(N_y) \in N_x$

Clearly, a function is continuous on a whole space if and only if it is continuous at every point of the space.

For metric spaces, continuity of a map can be defined in a different, although equivalent fashion.

Definition 2.1.15 Let (X, d_1) and (Y, d_2) be metric spaces. A map $f : X \rightarrow Y$ is continuous at a point $x \in X$ if for any $\epsilon > 0$ there is $\delta > 0$, which may depend on x and ϵ , such that $d_1(x, y) < \delta$ implies $d_2(f(x), f(y)) < \epsilon$.

The following definition, which is available only for functions between metric spaces, is also very important.

Definition 2.1.16 A map $f : X \rightarrow Y$ between metric spaces (X, d_1) and (Y, d_2) is Lipschitz if there is a constant $L > 0$ such that $d_2(f(x), f(y)) \leq Ld_1(x, y)$ for all $x \in X, y \in Y$.

Considering Euclidean spaces, it is also possible to define the concept of a C^r function.

Definition 2.1.17 A function $f : U \rightarrow \mathbb{R}^m$, where $U \subset \mathbb{R}^n$ is open, $f = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, $\mathbf{x} = (x_1, \dots, x_n)$, is C^r , $r \in \mathbb{N}$, if it is continuous and its partial derivatives,

$$\frac{\partial^k f_j}{\partial x_1^{k_1} \dots \partial x_n^{k_n}}, \text{ where } 1 \leq k \leq r, 1 \leq j \leq m, \text{ and } k_1 + \dots + k_n = k,$$

exist and are continuous on $U \subset \mathbb{R}^n$. If U is not an open set, f is C^r on U if it can be extended to a function that is C^r on an open neighborhood of U . The function f is called smooth, or C^∞ , if it is C^r for all $r \geq 1$.

A derivative of a function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ at a point \mathbf{x}_0 is a linear map $f'(x_0) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ defined by

$$f'(\mathbf{x}_0) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}_0) & \cdots & \frac{\partial f_1}{\partial x_n}(\mathbf{x}_0) \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1}(\mathbf{x}_0) & \cdots & \frac{\partial f_m}{\partial x_n}(\mathbf{x}_0) \end{pmatrix}$$

A point $\mathbf{x} \in \mathbb{R}^n$ is called *regular* if the rank of $f'(\mathbf{x})$ is maximal. Otherwise the point \mathbf{x} is called *critical* and the value $f(\mathbf{x}) \in \mathbb{R}^m$ is called a *critical value*. If $\mathbf{y} \in \mathbb{R}^m$ is not a critical value, it is called a *regular value*, even if $\mathbf{y} \notin f(\mathbb{R}^n)$.

Having defined some properties of maps, we can now use them to define various notions of equivalence between spaces.

Definition 2.1.18 A continuous map $f : X \rightarrow Y$, where (X, \mathcal{T}_1) and (Y, \mathcal{T}_2) are topological spaces, is a *homeomorphism* if it is a bijection and its inverse, $f^{-1} : Y \rightarrow X$, is also continuous. Two topological spaces, X and Y , are called *homeomorphic*, denoted by $X \approx Y$, if there exists a homeomorphism between them.

Homeomorphic spaces exhibit the same topological properties. Thus, we say that two subsets of a topological space are equivalent if there is a homeomorphism between them. If topological spaces under consideration are, actually, Euclidean spaces, it is useful to distinguish the case when a homeomorphism is differentiable.

Definition 2.1.19 Two open subsets $U \subset \mathbb{R}^n$ and $V \subset \mathbb{R}^n$ are *C^r -diffeomorphic*, $r \geq 1$, if there is a homeomorphism $f : U \rightarrow V$ such that f and f^{-1} are C^r . In this case f is a *diffeomorphism*.

A little less restrictive relation between subsets of \mathbb{R}^n is provided by a *C_p^r -isomorphism*.

Definition 2.1.20 A function $f : U \rightarrow \mathbb{R}^m$, $U \subset \mathbb{R}^n$, is *C_p^r* if it is continuous on U and C^r on $U \setminus S$, where S is a subset of a countable union of non-accumulating sets, each of which is C^r -diffeomorphic with a q -flat ($0 \leq q < n$), that is, with a set of the

form $\{\mathbf{x} \in \mathbb{R}^n : x_i = 0, q < i\}$. If f is a homeomorphism and both f and f^{-1} are C_p^r , then f is called a C_p^r -isomorphism.

We say that two subsets, $X \subset \mathbb{R}^n$ and $Y \subset \mathbb{R}^n$, are C_p^r -isomorphic, or piecewise C^r -diffeomorphic, if there is a C_p^r -isomorphism $f : X \rightarrow Y$.

Considering different subsets of a space, we might be interested not only in their (topological) equivalence, but also in a possibility of ‘deforming’ one set into the other. Mathematically this can be expressed in terms of *homotopy*.

Definition 2.1.21 *Let (X, \mathcal{T}_1) and (Y, \mathcal{T}_2) be topological spaces. Continuous functions $f : X \rightarrow Y$ and $g : X \rightarrow Y$ are said to be homotopic if there exists a continuous function $h : X \times [0, 1] \rightarrow Y$, called a homotopy between f and g , such that $h_0(x) = h(x, 0) = f(x)$ and $h_1(x) = h(x, 1) = g(x)$.*

A special case of a homotopy is provided by a *deformation retraction*. We say that a homotopy $h : X \times [0, 1] \rightarrow X$ is a deformation retraction of a space X onto a subspace A if $h_1(X) = A$ and $h_t|_A = id_A$, for all $t \in [0, 1]$.

Definition 2.1.22 *Let (X, \mathcal{T}_1) and (Y, \mathcal{T}_2) be topological spaces. A function $f : X \rightarrow Y$ is called a homotopy equivalence if there is a function $g : Y \rightarrow X$ such that $f \circ g$ is homotopic to id_Y , and $g \circ f$ is homotopic to id_X , where id_X and id_Y denote the corresponding identity functions. In this case, the spaces X and Y are called homotopy equivalent.*

Notice that if there is a deformation retraction of X onto A , then A and X are homotopy equivalent. Homotopy equivalence provides a nice generalization of topological equivalence and lies at the core of homotopy theory, which is described in detail in [91, 95]. More detailed accounts of spaces and maps can be found in [96, 72, 65, 26].

2.2 Manifolds and S -varieties

Most objects that we consider in this work are subsets of a Euclidean space that, besides being topological, or metric spaces, have an additional structure that renders them *manifolds*. Manifolds lie at the core of differential topology and have been extensively studied throughout the years. The reader can find thorough and deep discussions of manifolds in such works as [43, 71, 37, 85]. Analysis of more complicated objects, *varieties*, can be found in [1, 75].

Definition 2.2.1 *A Hausdorff paracompact topological space M is an n -dimensional (topological) manifold if there is an open cover $\{U_\lambda\}_{\lambda \in \Lambda}$ such that each U_λ is homeomorphic to an open (with respect to the induced topology) subset of a half-space, $\mathbb{H}^n = \{\mathbf{x} \in \mathbb{R}^n | x_n \geq 0\}$.*

This definition implies that for any $x \in M$ there is an open set $U_\lambda \subset M$ and a map $\phi_\lambda : U_\lambda \rightarrow \mathbb{H}^n$ such that ϕ_λ maps U_λ homeomorphically onto its image. A pair $(U_\lambda, \phi_\lambda)$ is called a *chart*, and the collection of charts, $\mathcal{A} = \{(U_\lambda, \phi_\lambda)\}_{\lambda \in \Lambda}$ is called an *atlas*. A point $x \in M$ is called a boundary point if $x \in \phi_\lambda^{-1}(\partial\mathbb{H}^n)$, where $\partial\mathbb{H}^n = \{\mathbf{x} \in \mathbb{R}^n | x_n = 0\}$, for some chart $(U_\lambda, \phi_\lambda)$. The set of all boundary points of M is called the *boundary* of M and is denoted by ∂M . If $\partial M \neq \emptyset$, then M is called a manifold with boundary, or ∂ -manifold. Otherwise, it is called a manifold without boundary, or just a manifold.

Consider an n -dimensional manifold, M , and two charts, (U, ϕ) and (V, ψ) . The map $\psi \circ \phi^{-1} : \phi(U) \rightarrow \psi(V)$ is a homeomorphism between subsets of a Euclidean space, which makes it possible to distinguish the case when $\psi \circ \phi^{-1}$ is differentiable. This distinction leads to a smoother version of a manifold.

Definition 2.2.2 *Let M be an n -dimensional manifold with an atlas \mathcal{A} . If for any two charts $(U, \phi) \in \mathcal{A}$ and $(V, \psi) \in \mathcal{A}$ such that $U \cap V \neq \emptyset$ the map $\psi \circ \phi^{-1} : \phi(U) \rightarrow \psi(V)$ is a C^r diffeomorphism, $1 \leq r \leq \infty$, then M is called a C^r manifold.*

Notice that every open subset of a manifold is also a manifold. Indeed, if $\mathcal{A} = \{(U_\lambda, \phi_\lambda)\}_{\lambda \in \Lambda}$ is an atlas for a manifold M and $O \subset M$ is open, then $\mathcal{A}_O = \{(U_\lambda \cap O, \phi_\lambda|_O)\}_{\lambda \in \Lambda}$ is an atlas for O . Subsets of a manifold that are manifolds themselves are called *submanifolds*.

Definition 2.2.3 *Let M be an n -dimensional manifold. A set $N \subset M$ is a submanifold of dimension k (codimension $(n - k)$) if for every $x \in N$ there is a chart $(U_\lambda, \phi_\lambda)$ such that $x \in U_\lambda$ and $N \cap U_\lambda = \phi_\lambda^{-1}(\mathbb{H}^k)$, $\mathbb{H}^k \subset \mathbb{H}^n$.*

It is fairly easy to show that the boundary, ∂M , of a ∂ -manifold M is a codimension-1 submanifold of M .

When dealing with C^r submanifolds of \mathbb{R}^n , it is usually useful to consider an alternative approach to their definition. Applying the Implicit Function Theorem, one can show that $M \subset \mathbb{R}^n$ is a k -dimensional, C^r submanifold, $1 \leq r \leq \infty$, if for every point $\mathbf{x} \in M$ there is an open neighborhood U in \mathbb{R}^n for which $M \cap U = f^{-1}(0)$, if \mathbf{x} is an interior point, and $M \cap U = f^{-1}(0) \cap g^{-1}([0, \infty))$, if \mathbf{x} is a boundary point, where $f : U \rightarrow \mathbb{R}^{n-k}$ and $g : U \rightarrow \mathbb{R}^k$ are C^r functions such that \mathbf{x} a regular point of f and g . Interestingly, if we drop the last condition, the resulting space becomes a subvariety. That is, a subset $V \subset \mathbb{R}^n$ is called a k -dimensional subvariety if for every point $\mathbf{x} \in V$ there is an open neighborhood U in \mathbb{R}^n for which $V \cap U = f^{-1}(0)$, if \mathbf{x} is an interior point, and $V \cap U = f^{-1}(0) \cap g^{-1}([0, \infty))$, if \mathbf{x} is a boundary point. If $V \cap U$ is comprised of only finitely many semi-disks, it is an s -subvariety. The point \mathbf{x} can be a critical point in this case. In general, an s -variety is defined similarly to a topological manifold: A topological space V is called a *topological s -variety* of dimension n if each $x \in V$ has an open neighborhood $U \subset V$ such that U is the union of finitely many sets, each of which is homeomorphic with an open subset of \mathbb{H}^n , and the intersection set of these homeomorphisms is comprised of finitely many manifolds of lower dimensions.

Very often one encounters submanifolds and s -subvarieties that, although not completely smooth, exhibit some level of smoothness. For example, the boundary of a square is not a smooth submanifold, but it can be easily subdivided into four smooth submanifolds – its sides. Such manifolds and s -varieties are called *piecewise smooth*, and their rigorous definition can be readily obtained from the foregoing definitions by merely changing C^r diffeomorphisms and maps to C_p^r isomorphisms and maps.

We conclude this section by introducing an important concept pertaining to manifold intersections.

Definition 2.2.4 *Let M and N be C^r submanifolds of \mathbb{R}^n , $r \geq 1$. A point $\mathbf{x} \in M \cap N$ is called a *transverse intersection point* if the tangent spaces of the two submanifolds at \mathbf{x} span \mathbb{R}^n , that is, $T_{\mathbf{x}}(M) + T_{\mathbf{x}}(N) = \mathbb{R}^n$.*

If the submanifolds are transverse at all their intersection points - denoted as $M \pitchfork N$ - it is easy to show that $M \pitchfork N$ is a C^r submanifold and $\dim(M \pitchfork N) = \dim M + \dim N - n$. We also note that if f is a diffeomorphism, then $M \pitchfork N$ implies that $f(M) \pitchfork f(N)$.

2.3 Vector Fields and Flows

As we have mentioned, swept volumes and swept manifolds are inherently related to some laws of motion, and therefore can be regarded as systems evolving in time. This point of view brings about such notions as a vector field and a flow, which belong to the dynamical systems discipline. The area of dynamical systems has become very popular in recent years, generating many excellent texts on the subject. For example, [4] and [79] provide a simple and intuitive description of the main concepts and results, while [44, 68, 12] give a more advanced presentation of the subject. In our brief discussion here, we introduce basic nomenclature, utilizing ideas from several sources.

Definition 2.3.1 *A vector field is a map $F : U \rightarrow \mathbb{R}^n$, where $U \subset \mathbb{R}^n$.*

Thus, a vector field assigns to each point $\mathbf{x} \in U$ a vector $F(\mathbf{x})$ which, when regarded as a law of motion, indicates in what direction and how fast a particle should move. For reasons explained below, we usually assume that a vector field is C^r , $r \geq 1$, or at least Lipschitz.

Recall that a curve in \mathbb{R}^n at a point \mathbf{x} is a map $c : I \rightarrow \mathbb{R}^n$, where I is an open interval, $I = (-a, a)$, such that $c(0) = \mathbf{x}$.

Definition 2.3.2 *Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a vector field. An integral curve of F at $\mathbf{x} \in \mathbb{R}^n$ is a C^1 curve at \mathbf{x} such that $c'(t) = F(c(t))$ for all $t \in I$.*

The question about existence of an integral curve is answered by the following theorem.

Theorem 2.3.3 *Let $U \subset \mathbb{R}^n$ be open, and $F : U \rightarrow \mathbb{R}^n$ be a C^r vector field, $r \geq 1$. Then for each $\mathbf{x}_0 \in U$ there is an integral curve at \mathbf{x}_0 and any two such curves coincide on the intersection of their domains. Moreover, there is a neighborhood $O_{\mathbf{x}_0}$ of \mathbf{x}_0 , $\epsilon > 0$, and a C^r map $\phi : O_{\mathbf{x}_0} \times I \rightarrow \mathbb{R}^n$, where $I = (-\epsilon, \epsilon)$, such that the curve $c_{\mathbf{x}} : I \rightarrow \mathbb{R}^n$ defined by $c_{\mathbf{x}} = \phi(\mathbf{x}, t)$ is an integral curve at \mathbf{x} .*

The map ϕ in the above theorem is called the *flow* generated by F . Notice that for $t = 0$ the map $\phi_0(\mathbf{x}) = \phi(\mathbf{x}, 0)$ is just the identity map, and as t grows, ϕ_t describes how $O_{\mathbf{x}_0}$ ‘flows’. It is not difficult to show that for a fixed t the map $\phi_t : O_{\mathbf{x}_0} \rightarrow \phi_t(O_{\mathbf{x}_0})$ is a C^r -diffeomorphism and its inverse is given by $\phi_t^{-1} = \phi_{-t}$. More generally, $\phi_t \circ \phi_s = \phi_s \circ \phi_t = \phi_{t+s}$, whenever the compositions are defined. This property is called the *group property*.

Sometimes it is necessary to consider vector fields that change with time. Such a vector field is a function $F : U \times I \rightarrow \mathbb{R}^n$, where U is an open subset of \mathbb{R}^n and I is an open interval. In this case, an integral curve, $c(t)$, satisfies $c'(t) = F(c(t), t)$. The existence and uniqueness theorem still holds, but the flow, ϕ , becomes time-dependent, that is, $\phi = \phi(\mathbf{x}, t, \lambda) = \phi_{t,\lambda}(\mathbf{x})$, $\lambda \in I$. It is still a diffeomorphism for

fixed t and λ , but the group property transforms into $\phi_{\tau,t}(\mathbf{x}) \circ \phi_{t,\lambda}(\mathbf{x}) = \phi_{\tau,\lambda}(\mathbf{x})$, $\phi_{t,t} = \text{identity}$.

2.4 Morse Theory

It is reasonable to assume that the topological structure of a space can be studied by analyzing certain types of maps defined on the space. Morse theory provides a specific method for describing the topology of a manifold M by evaluating critical points of a particular class of smooth functions from M to \mathbb{R} . This theory has been, and still is, one of the major tools for studying manifolds, and its employment significantly simplifies our later analysis of swept manifolds. In what follows, we provide a quick exposition of the basic principles of Morse theory. More comprehensive accounts of the topic can be found in such texts as [64, 67, 43].

Definition 2.4.1 *Let $f : U \rightarrow \mathbb{R}$, $U \subset \mathbb{R}^n$, be a C^r function, $r \geq 2$. A critical point \mathbf{x} of f is called nondegenerate if the Hessian matrix, $H(\mathbf{x}) = (\partial_{x_i x_j} f(\mathbf{x}))_{i,j=1}^n$, has rank n .*

Notice that nondegeneracy implies that $H(\mathbf{x})$ does not have zero eigenvalues. The number of its negative eigenvalues is called the *index* of the critical point \mathbf{x} .

A function $f : M \rightarrow \mathbb{R}$ defined on an n -manifold M is usually analyzed in local coordinates, which means that instead of f itself we consider a function $f \circ \phi^{-1} : \phi(U) \rightarrow \mathbb{R}$, where (U, ϕ) is a chart. Thus, $\mathbf{x} \in M$ is a nondegenerate critical point of f if it is a nondegenerate critical point of $f \circ \phi^{-1}$ for some, and hence for every chart (U, ϕ) such that $\mathbf{x} \in U$. Obviously, the same function can have different representations in different local coordinates, making it reasonable to look for the simplest one. For example, if \mathbf{x} is a regular point of f , then there is a chart (U, ϕ) such that $f \circ \phi^{-1}(y_1, \dots, y_n) = y_1$. Unfortunately, we cannot obtain the same form if \mathbf{x} is a critical point, but some level of simplification is still achievable, as provided by the following lemma.

Lemma 2.4.2 *Let M be a C^{r+1} manifold, $r \geq 1$, and let $\mathbf{x} \in M$ be a nondegenerate critical point of a C^{r+1} function $f : M \rightarrow \mathbb{R}$. Suppose that the index of \mathbf{x} is k . Then there is a C^r chart (U, ϕ) such that*

$$f \circ \phi^{-1}(y_1, \dots, y_n) = f(\mathbf{x}) - \sum_{i=1}^k y_i^2 + \sum_{i=k+1}^n y_i^2.$$

This lemma lies at the core of Morse theory and serves as a base for the following result, which is preceded by some technical definitions.

Definition 2.4.3 *A Morse function $f : M \rightarrow [a, b]$ is called admissible if $\partial M = f^{-1}(a) \cup f^{-1}(b)$ and a and b are regular values.*

Definition 2.4.4 *A Morse function $f : M \rightarrow [a, b]$ is of type $(\lambda_0, \dots, \lambda_n)$ if it has λ_k critical points of index k , $0 \leq k \leq n$.*

Definition 2.4.5 *Let M be a manifold, and let $D^k = \{\mathbf{x} \in \mathbb{R}^k \mid d(0, \mathbf{x}) \leq 1\}$, where d is the standard metric on \mathbb{R}^k . A k -cell in M is the image of D^k under a continuous map $f : D^k \rightarrow M$ such that $f : D^k \rightarrow f(D^k)$ is a homeomorphism.*

Theorem 2.4.6 *Let M be an n -dimensional compact manifold, and let $f : M \rightarrow [a, b]$ be an admissible Morse function of type $(\lambda_0, \dots, \lambda_n)$. If f has only one critical value $c \in (a, b)$, then there is a deformation retraction of M onto $f^{-1}(a) \cup \left\{ \bigcup_{k=0}^n \bigcup_{i=1}^{\lambda_k} e_i^k \right\}$, where the k -cells e_i^k , $1 \leq i \leq \lambda_k$, are pairwise disjoint and such that $e_i^k \subset M \setminus f^{-1}(b)$ and $e_i^k \cap f^{-1}(a) = \partial e_i^k$.*

Thus, passing a critical value of a Morse function corresponds to attaching cells, types of which are determined by critical points and their indices. In the simple case of only one critical point of index k , the manifold is homotopy equivalent to $f^{-1}(a)$ with a k -cell attached.

2.5 Homology Theory

In the interest of laying the groundwork for the homological intersection criteria that we derive in the sequel, we shall now give a very brief introduction to the basic concepts from homology theory that lie at the core of our approach. The reader can find a wealth of additional information and details in such texts as [63], [52], [42], and [73].

2.5.1 Simplices and Singular Simplices

Definition 2.5.1 *A subset of \mathbb{R}^d is an n -simplex, $0 \leq n \leq d$, if it is a convex hull of a set S of $n + 1$ affinely independent points.*

Thus, a 0-simplex, or vertex, is just a point, and a 1-simplex, or edge, is a line segment. Triangles and tetrahedra represent 2- and 3-simplices, respectively.

Definition 2.5.2 *Let Δ^n be an n -simplex, $n > 0$, defined by points in S . A $(k + 1)$ -element subset $F \subset S$ defines a k -simplex called a k -face of Δ^n .*

A 0-face of an n -simplex is a vertex, and a 1-face is an edge. For convenience, an $(n - 1)$ -face is often referred to as just a face.

Simplices lie at the core of simplicial homology, that is, homology defined for simplicial complexes. While simple and intuitive, simplicial homology does not provide a level of generality that can be found in *singular homology*, the type we concentrate on here. This kind of homology is better suited for more complicated topological spaces, and is based on the notion of a *singular simplex*.

Definition 2.5.3 *A singular n -simplex in a topological space X is a continuous map $\sigma : \Delta^n \rightarrow X$.*

A k -face of a singular n -simplex σ is defined as a restriction of σ to the corresponding k -face of Δ^n .

2.5.2 Chains, Boundary Maps and Homology Groups

Let us denote by $C_n(X)$ a free abelian group generated by all singular n -simplices in a topological space X . Then $C_n(X)$ consists of elements of the form $\sum_k n_k \sigma_k$, where each σ_k is a singular n -simplex in X , and $n_k \in \mathbb{Z}$. Elements of $C_n(X)$ are called n -chains. Considering $C_n(X)$ and $C_{n-1}(X)$, one can define a *boundary map* $\partial_n : C_n(X) \rightarrow C_{n-1}(X)$ as follows. On the basis elements,

$$\partial_n(\sigma) = \sum (-1)^k \sigma|_{\bar{\Delta}_k^n}.$$

Here $\bar{\Delta}_k^n$ denotes the k -th face of Δ^n , and $\sigma|_{\bar{\Delta}_k^n}$ is the corresponding restriction of σ , which is a singular $(n-1)$ -simplex. For an arbitrary element of $C_n(X)$, $\partial_n(\sum_k n_k \sigma_k) = \sum_k n_k \partial_n(\sigma_k)$. Boundary maps are homomorphisms of abelian groups, and it is easy to show that $\partial_n \partial_{n+1} = 0$. The collection $\{(C_n(X), \partial_n)\}$ of groups of n -chains and the corresponding boundary maps is called a *chain complex*, and is often represented by the following diagram:

$$\cdots \longrightarrow C_{n+1}(X) \xrightarrow{\partial_{n+1}} C_n(X) \xrightarrow{\partial_n} C_{n-1}(X) \xrightarrow{\partial_{n-1}} \cdots \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} 0$$

Since ∂_n is a homomorphism, its kernel, denoted by $Z_n(X)$, is a subgroup of $C_n(X)$. Elements of $Z_n(X)$ are called *cycles*, the name that becomes clear if one considers the 1-dimensional case, when a cycle is just a closed curve. Another important subgroup of $C_n(X)$ is the image of ∂_{n+1} , denoted by $B_n(X)$. Naturally, elements of this subgroup are called *boundaries*. Due to the property $\partial_n \partial_{n+1} = 0$, every boundary is a cycle. This allows us to define the quotient groups, $H_n(X) = Z_n(X)/B_n(X)$, which, consequently, contain those cycles that are not boundaries. These groups are called *homology groups* of X , and the collection $\{H_n(X)\}_{n \in \mathbb{Z}}$ is the *homology* of X .

2.5.3 Computing Homology: Exact Sequences

Below we provide several basic results, some of which follow fairly easily from the definition of homology groups.

Theorem 2.5.4 *Let X_i , $1 \leq i \leq k$, be path-connected components of a topological space X . Then $H_n(X)$ is isomorphic to the direct sum $\bigoplus_{i=1}^k H_n(X_i)$.*

This theorem becomes quite obvious if one recalls that a singular simplex is a continuous map and, consequently, preserves path-connectedness, thus splitting the corresponding groups.

Theorem 2.5.5 *For a point x , $H_n(x) = 0$ for $n > 0$, and $H_0(x) \approx \mathbb{Z}$.*

Theorem 2.5.6 *For a path-connected topological space X , $H_0(X) \approx \mathbb{Z}$*

As a corollary, we obtain that $H_0(X)$ is isomorphic to $\mathbb{Z}^k = \bigoplus_{i=1}^k \mathbb{Z}$, where k is the number of path-connected components of X .

The fact that a point has only one nontrivial homology group, H_0 , can cause some inconvenience. This can be remedied by introducing *reduced* homology groups, which are obtained by considering the following augmented chain complex:

$$\cdots \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} \mathbb{Z} \longrightarrow 0,$$

where the boundary map ∂_0 is defined by $\partial_0(\sum n_k \sigma_k) = \sum n_k$. Reduced homology groups are denoted by $\tilde{H}_n(X)$, and it is easy to show that $\tilde{H}_n(X) \approx H_n(X)$ for $n > 0$, and $H_0(X) = \tilde{H}_0(X) \oplus \mathbb{Z}$. Therefore, a point has only trivial reduced homology groups.

Consider now a continuous map $f : X \rightarrow Y$. Notice that if σ is a singular n -simplex in X , then $f \circ \sigma$ is a singular n -simplex in Y . Therefore, we can define a chain map $f_\# : C_n(X) \rightarrow C_n(Y)$ by $f_\#(\sigma) = f \circ \sigma$. The map $f_\#$ is a homomorphism, and it can be shown that it induces a homomorphism between homology groups,

$f_* : H_n(X) \rightarrow H_n(Y)$. Two basic properties of such induced homomorphisms are: (i) $(f \circ g)_* = f_* \circ g_*$; (ii) $Id_* = Id$; and the main result is given by the following theorem.

Theorem 2.5.7 *Let maps $f, g : X \rightarrow Y$ be homotopic. Then $f_* = g_*$, that is, homotopic maps induce the same homomorphism between homology groups.*

As a corollary, we obtain the following: a contractible space has the homology of a point.

While it is sometimes possible to find homology groups of a space proceeding directly from the definition, complex topological spaces require employment of additional tools, because computations become cumbersome, if not unfeasible. Since one is dealing with groups, it is natural to borrow some techniques from the realm of algebra, and the main concept that is needed in the sequel is that of an *exact sequence*.

Definition 2.5.8 *The sequence of homomorphisms*

$$\cdots \longrightarrow G_{n+1} \xrightarrow{a_{n+1}} G_n \xrightarrow{a_n} G_{n-1} \xrightarrow{a_{n-1}} \cdots$$

is exact if $\text{Ker } a_n = \text{Im } a_{n+1}$

There are many kinds of exact sequences for homology groups, but we will need only two of them. The first one involves *relative* homology groups, which are defined as follows. Let $A \subset X$ be a subspace of X and consider the groups of n -chains in A , $C_n(A)$. These are subgroups of $C_n(X)$, and we can form the quotient groups, $C_n(X, A) = C_n(X)/C_n(A)$, thus making chains in A trivial. Notice that the boundary map $\partial_n : C_n(X) \rightarrow C_{n-1}(X)$ takes $C_n(A)$ to $C_{n-1}(A)$, which allows us to define the quotient boundary map and obtain the following chain complex:

$$\cdots \longrightarrow C_{n+1}(X, A) \xrightarrow{\partial} C_n(X, A) \longrightarrow \cdots$$

CHAPTER 3

SWEPT MANIFOLDS

Before trying to design a geometric or topological algorithm, it is crucial that the corresponding structure of objects under consideration be thoroughly investigated, elucidating properties that may facilitate the process of the algorithm development. In this work, we are primarily interested in a new class of objects – swept manifolds – and this chapter is devoted to investigating basic properties of such objects. Since such an investigation is the first attempt of its kind, most of the results presented here are new, although many of them are based on standard concepts of differential topology and geometry. The few cases when we refer to a classical theorem are indicated explicitly.

In order to properly introduce the notion of a swept manifold, we start by providing a brief overview of sweeps and swept volumes, introducing basic concepts and citing major results. Then follows the main part of the chapter, in which the topological and geometric structure of swept manifolds is studied. Finally, the relation between smooth manifolds and swept manifolds is considered, which leads to possible generalizations and some questions.

3.1 Overview of Sweeps and Swept Manifolds

Perhaps one of the most natural occurrences of sweeps can be found in the field of robotics. Planning a motion of a robot requires solving the collision detection problem ([56, 47]), which naturally involves trajectories of some geometric entities. Consider, for example, a moving arm of a robot. As it moves, it *sweeps* another more complicated geometric shape called *swept volume*, and if one needs to check for possible obstructions on the way, it suffices to check whether the swept volume intersects the objects around the arm. In general, a moving object can also be deforming, which

leads us to the following intuitive definition: a swept volume as a set of points in space traversed by a moving and possibly deforming object.

Arguably, the most attractive feature of swept volumes is a great variety of shapes of different complexity that can be obtained by sweeping a very simple object. For example, look at the object in Figure 3.1. This is a swept volume of a ball. Obviously, its geometry is nothing like the geometry of a ball. Moreover, its topology



Figure 3.1 Swept volume of a ball.

is also more complicated, since it is not simply connected¹. This property of swept volumes can be a very effective tool for representing sophisticated geometric objects. In fact, sweeping techniques became quite popular in solid modeling in recent years. The main problem of such an approach is the computation of a swept volume given the initial object and sweep. This is by no means a trivial task, and a lot of effort was put into it ([16, 18, 14]). Some of the findings on swept volumes are directly related to our work, and we shall now present a brief, but fairly rigorous discussion of the underlying concept and main results.

First, let us define precisely what we mean by an object. In what follows, an object will refer to a compact, connected, orientable and piecewise smooth submanifold of \mathbb{R}^n , with or without a boundary. If it is a submanifold with boundary, its interior should be smooth. Where it does not lead to a confusion, we may abuse notation and call an object simply a manifold or submanifold.

The motion of an object can be mathematically described as follows.

¹A reader familiar with the fundamental group can easily see that $\pi_1(X) = \mathbb{Z} * \mathbb{Z}$, where X is the swept volume

Definition 3.1.1 A sweep is a continuous map $\sigma : I \rightarrow \text{Diff}_c(\mathbb{R}^n)$, where $I = [0, 1]^2$, and $\text{Diff}_c(\mathbb{R}^n)$ is the space of all smooth diffeomorphisms of \mathbb{R}^n fixing points in the complement of a compact subset and such that $\sigma_0 = \sigma(0) = \text{id}$, the identity map.

Seemingly very abstract, this definition of a sweep provides a great deal of generality, easily encompassing the most complicated deforming motions. Simpler sweeps can be obtained by merely restricting the range of σ .

Definition 3.1.2 A sweep σ is called rigid if $\sigma(I) \subset \text{Euc}(n)$, the Lie group of Euclidean isometries of \mathbb{R}^n . Otherwise, σ is a deforming sweep.

Given an object M and sweep σ , there is the sweep map, $\Sigma : M \times I \rightarrow \mathbb{R}^n$, associated to them, which is defined by $\Sigma(\mathbf{x}, t) = \sigma_t(\mathbf{x})$.

Definition 3.1.3 A swept volume of an object M obtained with a sweep σ is $S_\sigma(M) = \Sigma(M \times I)$.

Considering a swept volume, $S = S_\sigma(M)$, it is useful to define a t -section, $S_t = \sigma_t(M) = \{\sigma_t(\mathbf{x}) | \mathbf{x} \in M\}$, which represents the moving object, M , at time t . Similarly, a trajectory of a point $\mathbf{x} \in M$ is defined by $\sigma_{[0,1]}(\mathbf{x}) = \{\sigma_t(\mathbf{x}) | 0 \leq t \leq 1\}$. If each trajectory of σ is closed, or, equivalently, $S_0 = S_1$, the sweep and the corresponding swept volume are called *periodic*; otherwise, they are called *non-periodic*.

While mathematically appealing, the above definition of a swept volume does not provide an easy way of performing actual computations. Fortunately, if a sweep is smooth, it is possible to derive another, computationally attractive representation of a swept volume. Indeed, σ is just a curve in $\text{Diff}_c(\mathbb{R}^n)$, and if it is smooth, the Lie group structure of $\text{Diff}_c(\mathbb{R}^n)$ leads to the sweep differential equation (SDE),

$$\frac{d\mathbf{x}}{dt} = \mathbf{X}_\sigma(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^n, \quad t \in \mathbb{R},$$

² There is no loss of generality, since any finite closed interval can be reduced to $[0, 1]$ by a simple reparametrization

whose trajectories generate the sweep σ ([19]). The right-hand side of the SDE, $\mathbf{X}_\sigma(\mathbf{x}, t)$, is called the *sweep vector field* (SVF).

The SDE based representation of swept volumes turned out to be very effective and was used as the basis for several rendering algorithms ([13, 19]). Moreover, it allows us to utilize the powerful machinery of dynamical systems and obtain more insight into the structure of swept volumes. The only drawback, from a dynamical systems viewpoint, could be the fact that the sweep vector field is, generally, time dependent. This can be rectified by introducing an extension of the SDE to space-time, which, naturally, is called the *extended sweep differential equation (ESDE)* ([18]). It is defined by

$$\frac{d\mathbf{u}}{ds} = \mathbf{X}_\sigma^*(\mathbf{u}),$$

where $\mathbf{u} = (\mathbf{x}, t)$, and $\mathbf{X}_\sigma^*(\mathbf{u}) = (\mathbf{X}_\sigma(\mathbf{u}), 1)$. Integration readily yields $t = s + \text{const}$, so the parameter s is just a translation of time. The flow of the ESDE, $\sigma_s^*(\mathbf{x}, t) = (\sigma_s(\mathbf{x}), t + s)$, is called the extended sweep of σ , and the corresponding *extended sweep map*, $\Sigma^* : M \times I \rightarrow \mathbb{R}^{n+1}$, is defined by $\Sigma^*(\mathbf{x}, s) = \sigma_s^*(\mathbf{x}, 0)$, where M is an object. Finally, the *extended swept volume* of the object M is $S_\sigma^*(M) = \Sigma^*(M \times I)$.

The following result is easily obtained by employing basic properties of flows of autonomous dynamical systems.

Theorem 3.1.4 *An extended swept volume is a piecewise smooth submanifold of \mathbb{R}^{n+1} .*

Unfortunately, a generic swept volume is not a manifold, since there may be self-intersections. It turns out that swept volumes belong to a wider, but still manageable class of s -varieties. The proof of this fact is based on the following lemma, and can be found in [19].

Lemma 3.1.5 *Let $\pi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ denote the canonical projection, i.e. $\pi(\mathbf{x}, t) = \mathbf{x}$. Then given an object M and sweep σ , $\pi(S_\sigma^*(M)) = S_\sigma(M)$. That is, a swept volume is the image of the extended swept volume under the canonical projection.*

Proof. It suffices to show that the projection of the trajectory of the extended sweep, σ_s^* , starting at a point $(\mathbf{x}_0, 0)$, is equal to the trajectory of the sweep σ_t starting at the point \mathbf{x}_0 . The latter fact is easy to verify. The first trajectory is given by $\sigma_{[0,1]}^*(\mathbf{x}_0, 0)$, which by definition is equal to $\sigma_{[0,1]}(\mathbf{x}_0) \times [0, 1]$. Applying the projection, π , we obtain $\pi(\sigma_{[0,1]}^*(\mathbf{x}_0, 0)) = \pi(\sigma_{[0,1]}(\mathbf{x}_0) \times [0, 1]) = \sigma_{[0,1]}(\mathbf{x}_0)$, which is exactly the second trajectory. \square

Theorem 3.1.6 *A swept volume of a codimension- k object, $k \geq 0$, is a codimension- m s-subvariety, $m = \max\{0, k - 1\}$.*

In most practical applications, an object being swept is either of codimension-0 or codimension-1. Hence, its swept volume is a compact, connected, oriented, codimension-0 s-subvariety, and, consequently, it is completely defined by its boundary. Computation of a swept volume boundary has been a topic of great importance in the field for many years, and several effective methods have been developed ([40]). Notice, though, that in many cases the boundary of a swept volume is to a large extent determined by a codimension-1 submanifold of the initial object M – the set of initial grazing points, that is, points where the SVF is tangent to M (if M is of codimension-1 and with a boundary, then it is exactly the boundary, ∂M). In fact, the boundary of a swept volume can often be found by sweeping the set of initial grazing points and evaluating self-intersections. Thus, it seems reasonable to focus our attention on sweeps of codimension-2 objects and try to develop methods for finding self-intersections of the corresponding swept volumes. Moreover, sweeps of objects of higher codimension may also be of use.

Definition 3.1.7 *A generalized swept manifold is a swept volume of a codimension- k object, $k \geq 2$.*

Questions related to generalized swept manifolds are thoroughly investigated in the sequel.

3.2 Introduction to Swept Manifolds

The foregoing discussion showed the importance of studying generalized swept manifolds. In this section, we start building the corresponding mathematical framework. Except for a few classical results, the findings presented in this section are new, and some of them will be extensively employed later, in the course of developing intersection algorithms for swept manifolds.

The first important observation regarding generalized swept manifolds is that they are not manifolds. According to Theorem 3.1.6, they are s -subvarieties. But using Lemma 3.1.5, one can easily infer that all non-manifold points of generalized swept manifolds are produced by self-intersections. Moreover, there will be only a finite number of such self-intersections. Thus, we can subdivide a generalized swept manifold into several pieces such that each of them is a manifold, and the original object is obtained by putting these pieces together, and then evaluating the corresponding intersections (see Figure 3.2). This simple fact shows that we can safely restrict our attention to generalized swept manifolds that actually are manifolds.

Definition 3.2.1 *A codimension- k submanifold of \mathbb{R}^n , $k \geq 1$, is a swept manifold of type 1 if it is a swept volume of a codimension- $(k + 1)$ object.*

One of the main tools in the analysis of swept volumes was the SDE, and since a swept manifold is a specific case of a swept volume, we conclude that the type 1 swept manifold of an object M with a smooth sweep σ is also given by the set of trajectories of the SDE

$$\frac{d\mathbf{x}}{dt} = X_{\sigma}(\mathbf{x}, t)$$

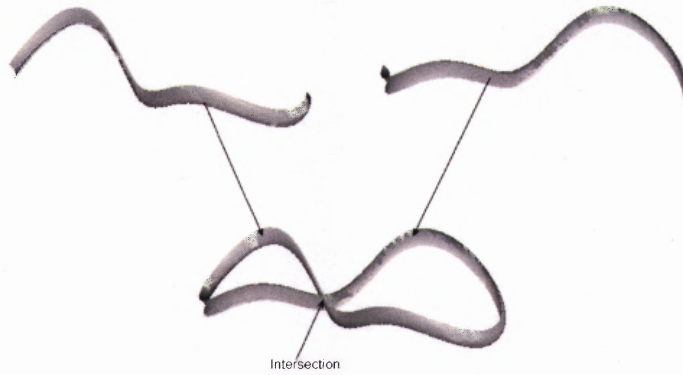


Figure 3.2 Subdivision of a generalized swept manifold into two intersecting manifolds.

originating from M and continuing until time $t = 1$. To take advantage of the SDE representation, we further consider only smooth sweeps and shall not indicate this assumption explicitly.

Focusing solely on the SDE representation of swept manifolds allows us also to broaden the class of objects under consideration. Indeed, our analysis will remain the same if we change the SVF $X_\sigma(\mathbf{x}, t)$ to an arbitrary C^r vector field, $r \geq 1$.

Definition 3.2.2 *A codimension- k submanifold of \mathbb{R}^n , S , is a swept manifold (of type 2), if there is a codimension- $(k + 1)$ object M and a C^r , $r \geq 1$, vector field $X(\mathbf{x}, t)$ on \mathbb{R}^n that generates the flow $\varphi_t(\mathbf{x})$ such that $S = \varphi_{[0,1]}(M) = \{\varphi_t(x) : (x, t) \in M \times [0, 1]\}$. The object M is called an initial manifold (object) of S , and the pair (M, X) is called a generator of S .*

Most of our work will be done for swept manifolds of type 2, which will be usually referred to simply as swept manifolds. Obviously, results obtained for swept manifolds (of type 2) are also true for swept manifolds of type 1. The reverse statement is, generally, not true, which broaches the subject of equivalence of the two types. The main question is: What condition should a vector field satisfy to be the SVF. The answer is provided by the following theorem, which is an adaptation of a standard result in differential topology.

Theorem 3.2.3 *Every smooth vector field with compact support is a SVF.*

Proof. Let $F : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n$ be a smooth vector field with compact support, and let $\varphi_t(\mathbf{x})$ be the flow generated by F . Then for each $t \in [0, 1]$, $\varphi_t : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a diffeomorphism of \mathbb{R}^n fixing points in the complement of a compact subset. In addition, φ_0 is the identity map, so $\varphi_t \in \text{Diff}_c(\mathbb{R}^n)$. Thus, we can define a map $\sigma : [0, 1] \rightarrow \text{Diff}_c(\mathbb{R}^n)$ by $\sigma(t) = \varphi_t$. Properties of a flow of a smooth vector field guarantee that σ is smooth, and therefore is a sweep generating the vector field F . \square

Since smooth vector fields with compact support are dense in the space of C^r vector fields, $r \geq 0$, and since practical cases are covered by the SFV, the difference between the two types of swept manifolds is negligible. It also allows us to restrict our attention to swept manifolds of type 2 that are generated by smooth vector fields.

3.2.1 Basic Properties of Swept Manifolds

Having laid out the basic definitions, we can now proceed to investigating the class of swept manifolds. The main topic on our agenda is the structure of *swept surfaces*, that is, codimension-1 swept manifolds in \mathbb{R}^3 . This case is particularly important because of its applicability – most geometric models are 3-dimensional. First, though, we explore general properties of swept manifolds, laying down a basis for our later analysis.

Look at the two surfaces shown in Figure 3.3. The first one is simply a sphere, and the second, which looks more complicated, is a diffeomorph of a sphere. If asked whether these surfaces are swept surfaces, one would promptly say that the sphere definitely is, because it is the surface of revolution of a semicircle. The sweep for the other surface is not obvious, though one might venture to say that it also is a swept surface, because, intuitively, diffeomorphisms should preserve the property of being swept. Such a guess would be correct.

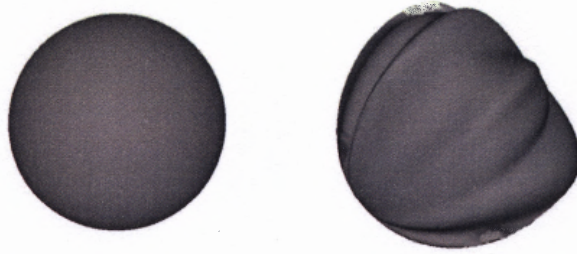


Figure 3.3 A sphere and its diffeomorph.

Theorem 3.2.4 *Let $M \subset \mathbb{R}^n$ be a piecewise smooth submanifold with a smooth interior, and let $S \subset \mathbb{R}^n$ be a swept manifold. Suppose there is a diffeomorphism $f : U_S \rightarrow U_M$, where U_M and U_S are open neighborhoods of M and S , respectively. Then M is a swept manifold.*

Proof. Let N be the initial manifold of S , and $F(\mathbf{x}, t)$ be its generating vector field. Define a smooth vector field on U_M by $G(\mathbf{y}, t) = D_{f^{-1}(\mathbf{y})}f \circ F(f^{-1}(\mathbf{y}), t)$, $\mathbf{y} \in U_M$. It is easy to check that if $c_{\mathbf{x}}(t)$ is a trajectory of F emanating from $\mathbf{x} \in N$, that is, $c_{\mathbf{x}}(t) = \phi_t(\mathbf{x})$, $\mathbf{x} \in N, t \in [0, 1]$, then $f \circ c_{\mathbf{x}}(t)$ is a trajectory of G emanating from $y = f^{-1}(\mathbf{x}) \in f^{-1}(N)$. Since $M = f(\cup_{\mathbf{x} \in N} c_{\mathbf{x}}([0, 1]))$ and G can be extended to \mathbb{R}^n by employing partition of unity, we see that M is a swept manifold generated by G and with the initial manifold $f^{-1}(N)$. \square

Remark 3.2.5 *If two piecewise smooth compact orientable submanifold of \mathbb{R}^n with smooth interiors are piecewise diffeomorphic, the corresponding diffeomorphism is extendable to some neighborhoods of the submanifolds. Therefore, the above theorem holds if M and S are piecewise diffeomorphic.*

Consider a line segment moving in space. If a vector field governing the motion is transverse to the line, the resulting surface is obviously homeomorphic to a square (Figure 3.4), and that is what one would usually expect. It is possible, though, that at some time during the motion the vector field becomes tangent to the line segment (see



Figure 3.4 Regular swept surface of a line segment.

Figure 3.5). Moreover, the motion can be just a translation along the corresponding

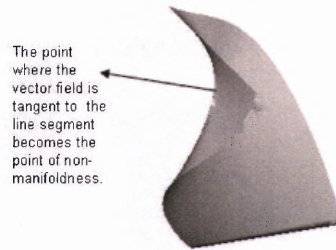


Figure 3.5 Non-regular swept surface of a line segment.

line, thus making the swept volume just another line segment. The later two case are not what one usually anticipates. Indeed, the resulting generalized swept manifold in the first case is not even a manifold. In the second case, the line segment is a swept manifold, but using another line segment to sweep it does not seem reasonable. This brings us to the following definition.

Definition 3.2.6 *A swept volume S with a generator (M, X) is called a regular swept volume if $X(\mathbf{x}, t)$ is transverse to $\varphi_t(M)$ for all $t \in [0, 1]$, where φ_t is the flow generated by X .*

Our definition of regularity is closely related to the one given in [18]. The main advantage of regular swept manifolds is that they greatly simplify the underlying analysis by eliminating some complicated cases. Unfortunately, there are important examples not covered by regular sweeps. The most crucial one is, probably, a sphere. A 2-sphere, S^2 , is obtained by rotating a semicircle around the corresponding axis. More precisely, we can choose the semicircle $M = \{(x, y, z) \in \mathbb{R}^3 : x = \sqrt{1 - z^2}, z \in$

$[-1, 1], y = 0\}$ and vector field $X(x, y, z) = (2\pi y, -2\pi x, 0)$ as a generator of the sphere $S^2 = \{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1\}$. Notice that there are two points, $(0, 0, \pm 1)$, where X is not transverse to $\varphi_t(M)$. In fact, $X(0, 0, \pm 1) = 0$, so these points are critical points of X . Another important observation is that the two critical points lie on the boundary of the initial manifold M . As we show later, augmenting regular swept manifolds by those having critical points on the boundary of an initial object produces a class of manifolds virtually as broad as the whole class of swept manifolds.

Topological simplicity of regular swept manifolds can also be seen from the following important result, which can be proved using the techniques employed to prove the two previous theorems.

Theorem 3.2.7 *Let S be a regular swept manifold with a generator (M, X) . Then S is piecewise diffeomorphic to $M \times [0, 1]$ if X is non-periodic, and to $M \times S^1$ if X is periodic. The corresponding interiors are always diffeomorphic.*

The above theorem also yields a nice description of the boundary of a regular swept manifold, which follows directly.

Corollary 3.2.8 *Let S be a regular swept manifold with a boundary, and let (M, X) be its generator. Denote by φ_t the flow generated by X . Then $\partial S = M \cup \varphi_{[0,1]}(\partial M) \cup \varphi_1(M)$.*

Looking at the proof of Theorem 3.2.7, one might notice that it is highly plausible that the reverse statement is also true. Indeed, manifolds of the type $M \times [0, 1]$ or $M \times S^1$, where M is an object, are formed by a simple translations or rotation of M , and a suitable homeomorphism should not change this drastically. A rigorous formulation of these ideas can be expressed as

Theorem 3.2.9 *Let S be a codimension- k , $k > 0$, submanifold of \mathbb{R}^n with a piecewise smooth boundary and smooth interior. Suppose that S is piecewise diffeomorphic to either $M \times [0, 1]$ or $M \times S^1$, where M is an object. Then S is a regular swept manifold.*

Proof. Since both $M \times [0, 1]$ and $M \times S^1$ are clearly swept manifold, the result follows from Theorem 3.2.4. \square

Theorems 3.2.7 and 3.2.9 represent criteria that lead to a nice topological description of regular swept manifolds. This description involves a reduction in dimensionality, which provides a great level of simplification. Of course, higher dimensional regular swept manifolds can still be extremely complicated, but, as we shall see, the most applicable class of regular swept surfaces possesses a simple topological classification.

We now list several interesting corollaries which focus on the fundamental group of a regular swept manifold, and can be proved readily from the definitions using standard techniques from algebraic topology.

Corollary 3.2.10 *Let S be a regular swept manifold with a non-periodic generator (M, X) . Then $\pi_1(S) \approx \pi_1(M)$.*

Corollary 3.2.11 *Let S be a regular swept manifold with a periodic generator (M, X) . Then $\pi_1(S) \approx \pi_1(M) * \mathbb{Z}$.*

3.2.2 Critical Swept Manifolds

Though the simplicity of regular swept manifolds is enticing, one should remember that several important swept manifolds are not regular. In this section, we investigate the effects of non-regularity on the structure of swept manifolds. Particularly, we focus on the type of non-regularity resulting from the presence of critical points of a generating vector field in an initial manifold. It turns out that manifoldness condition limits the variety of complications that would ensue otherwise.

Definition 3.2.12 *Let S be a swept manifold with a generator (M, X) , and suppose that X vanishes on some $C \subset M$. Then S is called a critical swept manifold.*

While there are no restrictions on the critical set C in this definition, one might suspect that it cannot be completely arbitrary. Indeed, consider a line segment J moving according to a non-periodic vector field that has a critical point $\mathbf{x} \in \text{Int } J$ (see Figure 3.6). Then the resulting generalized swept manifold is not a manifold, since a neighborhood of \mathbf{x} is not homeomorphic to an open ball. Surprisingly, this simple example has a very important generalization.

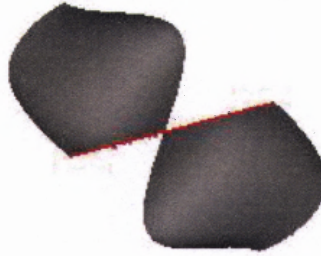


Figure 3.6 Generalized critical swept surface of a line segment.

Theorem 3.2.13 *Let $M \subset \mathbb{R}^n$ be a codimension- k object, $k \geq 2$, and let X be a smooth vector field on \mathbb{R}^n . Suppose also that $\mathbf{x} \in M$ is an isolated critical point of X . Then each of the following conditions implies that the generalized swept manifold of M obtained with X is not a manifold.*

1. $\dim M > 1$,
2. X is non-periodic, and \mathbf{x} is not one of the components of ∂M

Proof. Let $\dim M = k > 1$. Since M is a manifold, it suffices to show that for any $\epsilon > 0$ the generalized swept manifold of a k -dimensional ball of radius ϵ centered at \mathbf{x} , $B_\epsilon^k(\mathbf{x})$, is not a manifold if \mathbf{x} is a critical point of the vector field X . First, suppose that X is non-periodic. Since X is continuous and $B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\}$ is connected, we can assume that $\langle X, \mathbf{n} \rangle > (<)0$ for $\mathbf{x} \in \varphi_{[0, \delta]}(B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\})$, where $\langle \cdot, \cdot \rangle$ denotes the inner product, \mathbf{n} is the unit normal vector to the corresponding section, $\varphi_t(B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\})$, $t \in [0, \delta]$, and $\delta > 0$ is a sufficiently small number. Therefore, $\varphi_{[0, \delta]}(B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\})$

is homeomorphic to $(B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\}) \times [0, \delta]$, and since $\varphi_t(\mathbf{x} = \mathbf{x})$ for all t , we obtain that $\varphi_{[0, \delta]}(B_\epsilon^k(\mathbf{x}))$ is homeomorphic to $(B_\epsilon^k(\mathbf{x}) \setminus \{\mathbf{x}\}) \times [0, \delta] \cup \{\mathbf{x}\}$ and, hence, is not a manifold. The case when X is periodic is proved in the same way, the only difference is that the time span becomes $[-\delta, \delta]$ instead of $[0, \delta]$.

Now let $\dim M = 1$, $\mathbf{x} \in \text{Int } M$. This case can be reduced to the case of sweeping a line segment, $I = [\mathbf{x} - \epsilon, \mathbf{x} + \epsilon]$. Notice that $I \setminus \{\mathbf{x}\}$ has two connected components, and therefore (X, \mathbf{n}) can have different signs on different components. Nevertheless, if X is non-periodic, the same argument as above shows that $\varphi_{[0, \delta]}(I)$ is not a manifold. \square

One of the consequences of this result is that the critical subset C of a critical swept manifold cannot contain isolated points, unless the swept manifold is a surface and the critical points are exactly the boundary point, or the vector field is periodic. To see that the latter two cases are indeed possible, consider a rotation of a line segment around one of its end points (Figure 3.7). The resulting shape is a surface, and one of the boundary points is a critical point. The second case happens when one takes a circle and the corresponding rotational vector field to generate a 2-sphere (see Figure 3.8). Notice that periodicity is crucial in this case.



Figure 3.7 Non-periodical critical swept surface of a line segment.

Theorem 3.2.13 also gives more insight to other possible (and impossible) configurations of a critical set. The second condition in the theorem suggests that the only choice for a critical set of a non-periodic vector field can be a union of path connected components of the boundary of an initial object. The following result,



Figure 3.8 Periodic critical swept surface of a circle.

which can be proved readily by employing the same techniques as in the preceding theorem, just confirms this.

Theorem 3.2.14 *Let S be a critical swept manifold with a non-periodic generator (M, X) and a critical set $C \subset M$. Then C is a union of path connected components of ∂M .*

The next question we address is smoothness of non-periodic critical swept manifolds. The simple examples we considered so far always involved loss of smoothness at critical points. This is not a coincidence, but rather a result of the behavior of a vector field around a critical point. One might notice, though, that a critical subset should remain at the boundary of a critical manifold, thus yielding a smooth interior. In particular, one can easily prove the following result by using arguments analogous to those which were used to verify the statement of Theorem 3.2.13.

Theorem 3.2.15 *Let S be a critical swept manifold with a non-periodic smooth generator (M, X) and a critical set $C \subset \partial M$. Then $\text{Int } S$ is smooth, but S is only C^0 at any $\mathbf{x} \in C$. In addition, $\partial S = M \cup \varphi_{[0,1]}(\partial M) \cup \varphi_1(M)$.*

Unfortunately, analysis of general periodic critical swept manifolds is much more complicated, and basic properties resulting from non-periodicity are no longer valid: a critical set can belong to the interior of an initial object, C^∞ critical swept manifolds are quite common, and the boundary of an initial object does not have to remain a part of the boundary of the resulting swept manifold. In low dimensions,

though, these properties are not crucial, and a complete classification of critical swept manifolds is possible.

3.2.3 Swept Surfaces

Establishing the practical value of swept surfaces requires that additional effort should be put into their analysis. Moreover, the usefulness of some of the results obtained earlier for general swept manifolds may be a little obscure due to the abstract and unifying approach. Reemphasizing the main properties for swept surfaces may cast a new light on their possible applications, and utilizing additional mathematical tools provided by low dimensionality can help detect new important features that could elucidate the general structure of swept surfaces and facilitate design of efficient intersection algorithms.

Let us start by considering possible initial manifolds for swept surfaces. These are, in fact, just piecewise smooth, compact curves in \mathbb{R}^3 . It is natural to distinguish two cases right away: curves with a boundary, and curves without a boundary, that is, closed curves. In \mathbb{R}^3 , though, one should be careful about an additional possibility – knotted curves. Our notion of a knotted curve is somewhat different from the classical concept of a knot (and most like that of a braid), since we do not require a curve to be closed ([21]). Hence, both curves in Figure 3.9 are knotted, but only the second one is a knot. Studying swept manifolds with knotted initial curves seems to be a challenging and interesting task; in this work, however, we concentrate on more simple and, perhaps, more practical cases when an initial curve is not knotted, in which case we shall call it simple.

As in the general case, the structure of a swept surface simplifies significantly when the regularity condition is imposed. A low dimensionality for the initial manifold simplifies things even further, as illustrated by the following theorem.



Figure 3.9 Examples of knotted curves.

Theorem 3.2.16 *Let S be a regular swept surface with a generator (M, X) , and suppose that M not knotted. Then S is piecewise diffeomorphic to one of the following: (i) a square $Q = [0, 1] \times [0, 1]$, (ii) a cylinder $Y = S^1 \times [0, 1]$, (iii) a torus $T^2 = S^1 \times S^1$.*

Proof. The assumptions on M imply that M is piecewise diffeomorphic to either a line segment, $[0, 1]$, or a circle, S^1 . Therefore, $M \times [0, 1]$ is piecewise diffeomorphic to one of the three surfaces indicated in the statement of the theorem. Now the result follows from Theorem 3.2.7. \square

Remark 3.2.17 *In fact, this theorem also holds for the case of knotted initial curves.*

Notice that if an initial curve is smooth and closed, the resulting regular swept surface is also smooth. Similarly, regular periodic sweeps of smooth curves (closed or open) create smooth swept surfaces. Unfortunately, regular non-periodic sweeps of open curves always have points where smoothness is lost; indeed, it is easy to prove the following result using arguments in the proof of the above theorem.

Corollary 3.2.18 *Let S be a regular swept surface with a smooth generator (M, X) . Then S is smooth if and only if X is periodic or M is closed. If S is piecewise smooth, it is piecewise diffeomorphic to a square.*

Intuitively, one can expect that practical handling of sweeps of piecewise smooth curves should boil down to managing a finite collection of smooth curves due to the obvious subdivision of the initial curve into smooth parts. As we shall see

later, this is exactly the case. Therefore, it is reasonable to focus on swept surfaces with smooth initial curves and try to exploit the smoothness condition to discover additional properties. The classical result provided by the following easily proved lemma is the first step in this direction.

Lemma 3.2.19 *Let M is a smooth curve in \mathbb{R}^3 . Then there is a point $\mathbf{x}_0 \in M$ and a smooth vector field X on \mathbb{R}^3 such that M is the solution curve to the following initial value problem*

$$\frac{d\mathbf{x}}{ds} = X(\mathbf{x}, s), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad s \in [0, 1]$$

In other words, this lemma states that every smooth curve is, in fact, a swept curve. The reverse statement is, obviously, also true. Hence, a swept surfaces can be efficiently represented by an initial point, \mathbf{x}_0 , and two vector fields, X and Y . We shall regard such a triple as a generator of a swept surface and denote it by (\mathbf{x}_0, X, Y) ; the old notation for a generator will indicate the possibility of a piecewise smooth initial curve. The flows generated by X and Y will be usually denoted by φ_s and ψ_t , respectively. Notice, that the first vector field, X , provides the tangent space of the initial curve, $M = \varphi_{[0,1]}(\mathbf{x}_0)$, and the second is used to sweep this curve. Considering sections of the corresponding swept surface, $\psi_t(M)$, it is natural to assume that their tangent spaces are somehow related to the tangent space of M . Such a relation is provided by the following lemma.

Lemma 3.2.20 *Let S be a swept surface with a generator (\mathbf{x}_0, X, Y) , $S_t = \psi_t(M)$, where $M = \varphi_{[0,1]}(\mathbf{x}_0)$, and $\mathbf{y} = \psi_t(\mathbf{x})$, where $\mathbf{x} = \varphi_s(\mathbf{x}_0)$. Then the vector $X_Y = D_{\mathbf{x}}\psi_t \cdot X(\mathbf{x}, s)$ is tangent to the section S_t at \mathbf{y} .*

Proof. Since ψ_t is a diffeomorphism between M and S_t , and $X(\mathbf{x}, s)$ is tangent to M at \mathbf{x} , the derivative of ψ_t at \mathbf{x} , $D_{\mathbf{x}}\psi_t$, maps $X(\mathbf{x}, s)$ to a vector tangent to S_t at $\mathbf{y} = \psi_t(\mathbf{x})$. \square

Having the tangent space for each of the sections of a swept surface should surely be helpful in determining the tangent space of the whole surface. In fact, consider a swept surface S with a generator (\mathbf{x}_0, X, Y) . As the lemma shows, a tangent vector to S at a point $\mathbf{x} \in S$ can be found by sweeping the vector field X . Another tangent vector is given directly by the vector field Y , and if these two vectors are linearly independent at each point of S , they generate the tangent space of S . In general, linear independence of the two vectors is not guaranteed, but the problem can be easily rectified by imposing the regularity condition, thereby obtaining

Lemma 3.2.21 *Let S be a regular swept surface with a generator (\mathbf{x}_0, X, Y) , and let M, \mathbf{y} and X_Y be defined as in Lemma 3.2.20. Then X_Y and $Y(\mathbf{y}, t)$ are linearly independent vectors.*

Observe that it follows from these lemmas that each point on a swept surface seems to be uniquely determined by the two corresponding parameters, s and t . This correspondence defines a map from $[0, 1] \times [0, 1]$ to \mathbb{R}^3 which makes a swept surface a parametric surface. For example, it is easy to verify the following result.

Lemma 3.2.22 *Let S be a swept surface with a generator (\mathbf{x}_0, X, Y) . Then the map $f : [0, 1] \times [0, 1] \rightarrow \mathbb{R}^3$ defined by $f(s, t; \mathbf{x}_0) = \psi_t(\varphi_s(\mathbf{x}_0))$ is a parametric representation of S .*

Unfortunately, a closed form of the flow of a differential equation is, generally, not obtainable, but the few cases when an exact expression of the map f can be found represent a fairly broad class of swept surfaces and deserve some attention. This question will be considered in more detail in Section 4.2.5, when the intersection problem for the corresponding class of swept surfaces is considered. Meanwhile, we employ the regular swept surface classification to derive a very important result.

A careful reader has probably observed that though the tangent space at a point of a swept surface is generated by time dependent vector fields, it is, in itself,

not time dependent. Thus, one would expect that at each point of a swept surface the two vectors obtained using the time dependent fields should correspond to some time independent vectors. Constructing such vectors at each point of a swept surface would produce two time independent vector fields, which should behave very similar to the original ones, that is, they should generate the same swept surface.

Theorem 3.2.23 *Let S be a regular swept surface with a generator $(\mathbf{x}_0, X(\mathbf{x}, t), Y(\mathbf{x}, t))$. Then there are two time independent vector fields $X_1(\mathbf{x})$ and $Y_1(\mathbf{x})$ such that (\mathbf{x}_0, X_1, Y_1) is also a generator of S .*

Proof. Using Theorem 3.2.16, it is enough to show that a cylinder, a torus, and a square can be generated with time independent vector fields. Indeed, then a time independent vector fields for the original surface are obtained through a composition with the corresponding diffeomorphism. It is easy to see, though, that a square is generated by a pair of translations, a cylinder – by a rotation and translation, and a torus is generated by a pair of rotations. \square

Let us focus again on the roles played by generating vector fields, X and Y , which, from now on, are assumed to be time independent. Suppose these vector fields generate a swept surface S , then the initial manifold, M , is produced by the first vector field, X , sweeping the initial point \mathbf{x}_0 . Consider now a cross section $S_t = \psi_t(M)$. It follows from Theorem 3.2.20 that S_t can be obtained by sweeping the point $\psi_t(\mathbf{x}_0)$ using the vector field X_Y . This suggests that one may take $N = \psi_{[0,1]}(\mathbf{x}_0)$ to be an initial manifold and then generate S sweeping N with X_Y . Thus, the roles of Y and X , or at least X_Y , can be swapped. There is a little problem though. While the vector fields X and Y are defined on the whole \mathbb{R}^3 , X_Y is initially defined only at points of the swept surface, S . This raises a question: Is it possible to extend X_Y to \mathbb{R}^3 ? The answer to this question is positive, although such an extension may be quite sophisticated. Consider, for example, a torus, T^2 . As we know, it is a regular

C^∞ swept surface obtained by (rigidly) sweeping a circle, C_1 , along another circle, C_2 (see Figure 3.10). Exchanging the roles of C_1 and C_2 , we see that the sweep

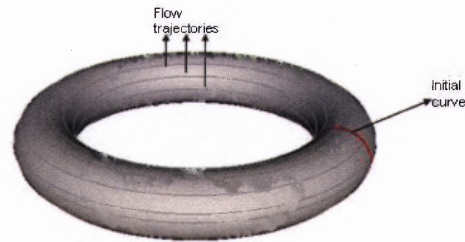


Figure 3.10 A torus as a swept surface.

becomes much more complicated; for instance, it is no longer rigid. Also, while it is clear how to sweep each point of C_2 to obtain T^2 , the extension of the corresponding vector field to the whole \mathbb{R}^3 is by no means obvious (at least to a less experienced reader). One possible suggestion could be the vector field with the flow depicted in Figure 3.11, which, incidentally, is related to a foliation of S^3 . Indeed, such a choice



Figure 3.11 Some trajectories of Hopf foliation of S^3 .

is the needed extension, and it is not very hard to show this rigorously. Hence, the standard generator of a torus has its *dual* generator, obtained by swapping the roles of the vector fields. Construction of dual generators for a square and a cylinder can be done in a similar way, and the following general result can be easily proved by employing a standard procedure of differential topology which is based on tubular neighborhoods, collars, and partition of unity.

Theorem 3.2.24 *Let S be a regular swept surface with a generator (\mathbf{x}_0, X, Y) . Then there is a vector field X^* defined on \mathbb{R}^3 such that $X^*|_S = X_Y$ and (\mathbf{x}_0, Y, X^*) is also a generator of S , called a dual generator.*

Existence of a dual generator has an obvious, but important consequence – every regular swept surface, S , has two associated vector fields, X^* and Y , defined on \mathbb{R}^3 , such that $X^*|_S$ and $Y|_S$ is a global frame for S . Having a global frame for a swept surface provides a big advantage when dealing with the intersection problem, and due to Theorem 3.2.24 we can safely assume that X^* and Y are given initially.

We conclude our discussion by making several observations regarding critical swept surfaces. First, consider a non-periodic swept surface with an open initial curve. In this case, there are only two possible critical points – boundary points. It is not difficult to show that if both boundary points are critical, the resulting swept surface is piecewise diffeomorphic to a half-disk; if only one boundary point is critical, the swept surface is piecewise diffeomorphic to a triangle (Figure 3.12). The case of periodic swept surfaces is more complicated, but it seems that most of such surfaces are piecewise diffeomorphic to one of the following: a sphere, a sphere with a triangle cut out (which is piecewise diffeomorphic to a triangle), and a disk (Figure 3.13).



Figure 3.12 Possible topological configurations of non-periodic critical swept surfaces.

3.2.4 Recursive Swept Manifolds

A large number of important properties of swept surfaces were obtained through the representation involving an initial point and two vector fields. One of the obvious advantages of such a representation is its uniformity – an initial curve is also a swept



Figure 3.13 Some of the possible topological configurations of periodic critical swept surfaces.

curve. Furthermore, specifying an explicit description of an initial curve provides additional information about it, thus giving us more insight to the structure of the entire swept surface. Not surprisingly, the question of a possible generalization of this approach to higher dimensions is coming next on our agenda.

It is well known that the complexity of a manifold grows 'exponentially' with its dimension. For example, there is a nice topological classification of surfaces ([43]), but no such classification is known for 3-manifolds. This, obviously, complicates our analysis of higher dimensional swept manifolds – an initial object can have a very intricate structure, creating a substantial obstacle to fully understanding the structure of the resulting swept manifold. Our study of swept surfaces was quite successful partly because an initial manifold was assumed to be a simple smooth curve. Note, that this assumption also makes an initial curve a swept curve – the fact that ultimately turned out to be the most useful. Although smooth manifolds are generally not swept manifolds, it is reasonable to ask what happens if one assumes that an initial manifold of some swept manifold is a swept manifold in itself. Does it simplify things significantly? There is an easy answer to this question – no, because though the initial manifold is a swept manifold, it can still be quite complicated due to its high dimension. The key here is to go further and demand that an initial manifold at every

level be a swept manifold; to simplify things even more, the regularity condition can be imposed. This bring us to the following definition.

Definition 3.2.25 *A recursive swept manifold of dimension 0 is a point. A recursive swept manifold of dimension $k > 0$ is a regular swept manifold that has a recursive swept manifold as its initial manifold. A generator of a recursive swept manifold of dimension k is denoted by $(\mathbf{x}_0, X_1, \dots, X_k)$.*

This definition nicely incorporates regular swept surface with smooth initial curves and provides a logical generalization to higher dimensions. Moreover, one would hope that a convenient mathematical representation of recursive swept manifolds will allow us to generalize some of the most important properties of swept surfaces. Indeed, let us denote the flows generated by the vector fields of a generator $(\mathbf{x}_0, X_1, \dots, X_k)$ by $\varphi_{t_1}^1, \dots, \varphi_{t_k}^k$. Then each point \mathbf{x} of the corresponding recursive swept manifold, M , is uniquely determined by the values of t_1, \dots, t_k , since $\mathbf{x} = \varphi_{t_k}^k \circ \varphi_{t_{k-1}}^{k-1} \circ \dots \circ \varphi_{t_1}^1(\mathbf{x}_0)$. Therefore, there is a simple parametrization of M , as shown by the following easily proved result.

Theorem 3.2.26 *Every k -dimensional recursive swept manifold $M \subset \mathbb{R}^n$ with a generator $(\mathbf{x}_0, X_1, \dots, X_k)$ has a smooth map $f: I^k \rightarrow \mathbb{R}^n, I = [0, 1]$ associated to it such that $M = f(I^k)$. The map f is defined by $f(t_1, t_2, \dots, t_k) = \varphi_{t_k}^k \circ \varphi_{t_{k-1}}^{k-1} \circ \dots \circ \varphi_{t_1}^1(\mathbf{x}_0)$.*

Thus, every recursive swept manifold is a parametric submanifold, although, as we mentioned before, such parametric representation may not be effectively computable.

Let us continue in the path trodden for swept surfaces and see if the generating vector fields of a recursive swept manifold can be assumed time independent. Recall that our proof of this result for swept surfaces was based on Theorem 3.2.16, which provides a topological classification of swept surfaces. It turns out that a similar classification can be obtained for recursive swept manifolds.

Theorem 3.2.27 *Let M be a recursive swept manifold of dimension $k \geq 2$. Then M is piecewise diffeomorphic to $I_{j_1} \times I_{j_2} \times \cdots \times I_{j_k}$, where $j_l \in \{0, 1\}$, $1 \leq l \leq k$, $I_0 = [0, 1]$, and $I_1 = S^1$.*

Proof. We have already proved the result for $k \leq 2$. Now assume that it holds for some k . Let $\dim M = k + 1$, then by Theorem 3.2.7 S is piecewise diffeomorphic to $N \times I_{j_{k+1}}$, $j_{k+1} \in \{0, 1\}$, where N is an initial manifold. Since N is a recursive swept manifold of dimension k , it is piecewise diffeomorphic to $I_{j_1} \times \cdots \times I_{j_k}$ by our assumption, which implies that S is piecewise diffeomorphic with $I_{j_1} \times \cdots \times I_{j_k} \times I_{j_{k+1}}$. By induction, the result holds for all $k \in \mathbb{N}$. \square

The statement about time independent generating vector fields readily follows from the above theorem.

Corollary 3.2.28 *Let M be a recursive swept manifold with a generator $(\mathbf{x}_0, X_1(\mathbf{x}, t_1), \dots, X_k(\mathbf{x}, t_k))$. Then there are time independent vector fields $Y_1(\mathbf{x}), \dots, Y_k(\mathbf{x})$ such that $(\mathbf{x}_0, Y_1(\mathbf{x}), \dots, Y_k(\mathbf{x}))$ is also a generator for M .*

Another very useful feature of swept surfaces is the description of their tangent spaces in terms of generating vector field. Fortunately, this property also can be readily generalized for recursive swept manifolds.

Theorem 3.2.29 *Let M be a recursive swept manifold with a generator $(\mathbf{x}_0, X_1, \dots, X_k)$. Then for every $\mathbf{x} \in \text{Int } M$ vectors $X_1^*(\mathbf{x}), \dots, X_{k-1}^*(\mathbf{x}), X_k(\mathbf{x})$, where*

$$X_i^*(\mathbf{x}) = D_{\mathbf{x}_{k-1}} \varphi_{t_k}^k D_{\mathbf{x}_{k-2}} \varphi_{t_{k-1}}^{k-1} \cdots D_{\mathbf{x}_i} \varphi_{t_{i+1}}^{i+1} \cdot X_i(\mathbf{x}_i), \quad 1 \leq i \leq k-1$$

$$\mathbf{x}_j = \varphi_{t_j}^j \circ \cdots \circ \varphi_{t_1}^1(\mathbf{x}_0), \quad 1 \leq j \leq k, \quad \text{and } \mathbf{x} = \mathbf{x}_k,$$

are linearly independent and span $T_x M$. Hence, $X_1^(\mathbf{x}), \dots, X_{k-1}^*(\mathbf{x}), X_k(\mathbf{x})$ is a global frame for TM .*

Finally, it would be worthwhile to invest some time in defining ‘dual’ generators for recursive swept manifolds. The main difficulty here is the large number

of possibilities in changing to the sweeping process. Indeed, there are $k!$ ways to generate the same recursive swept k -manifold, which corresponds to the number of permutations of $\{1, 2, \dots, k\}$. Therefore, it is more convenient to introduce the notion of *equivalent* generators. The following theorem is easy to prove, and it provides the details.

Theorem 3.2.30 *Let M be a recursive swept k -manifold with a generator $(\mathbf{x}_0, X_1, \dots, X_k)$. Then there is a family of $k!$ vector fields, $X_1^{\sigma(1)}, \dots, X_k^{\sigma(k)}$, where $\sigma \in S(\{1, \dots, k\})$ – a set of permutations of $\{1, \dots, k\}$ – such that each $(\mathbf{x}_0, X_1^{\sigma(1)}, \dots, X_k^{\sigma(k)})$ is a generator for M and $X_j^{\sigma(j)}|_M$, $1 \leq j \leq k$, can be obtained by sweeping one of the initial vector fields with the other vector fields. The generators of the form $\mathbf{x}_0, X_1^{\sigma(1)}, \dots, X_k^{\sigma(k)}$ are called *equivalent with respect to M* .*

An easy way to understand equivalent generators is to assume that each generating vector field creates only one particular dimension of M . This assumption becomes more natural if we recall that a collection of the generating vector fields is a frame for TM . Then choosing one of the equivalent generators corresponds to selecting a particular order of dimensions in which M is swept. For example, considering a cube (embedded in \mathbb{R}^4) we see that any of its three dimensions can be swept first, second, or third. In general, it is not very important which generator is used to sweep the manifold, but it can be crucial to have all equivalent generators available when performing a local (or global) analysis of the manifold during an intersection procedure.

The foregoing basic properties of swept manifolds provide a basis for our later development of efficient intersection algorithms. There are, however, several additional important issues concerning swept manifolds that are not explicitly related to algorithm development. One of the main examples, which is discussed in the next section, is the relation between general smooth manifolds and swept manifolds.

3.3 Smooth Manifolds vs. Swept Manifolds

So far regular swept surfaces and recursive swept manifolds were singled out for their simplicity, but there also is a point of view where simplicity may be far from what is needed. When designing a surface, one does not think about it in terms of vector fields, but rather envisions a whole surface as such, and only after mentally establishing the desired shape one would resort to mathematics in a search for a suitable representation. This is where the simplicity of swept surfaces may play against them – if the envisioned surface is quite complicated it may not be possible to find its generator. Thus, the ‘swept approach’ implicitly imposes restrictions on the complexity of a surface. If formidable, such restrictions can impede the modeling process and, consequently, diminish the potential of the approach. In what follows, we restate which smooth manifolds can be represented as swept manifolds and prove several new results that provide a useful tool for determining which smooth manifolds are essentially non-swept.

Before engaging in a deep discussion of sweeping criteria, let us make a simple observation which follows directly from the definition of a manifold: every manifold is locally flat. This immediately leads to the following result.

Theorem 3.3.1 *Every smooth submanifold $M \subset \mathbb{R}^n$ is locally swept, that is, for every point $\mathbf{x} \in M$ there is a neighborhood V with a compact closure such that \bar{V} is a swept manifold.*

Although essentially trivial, this theorem has a nice and important consequence: no matter how complicated a manifold is, it is always possible to subdivide it into *swept patches*. Therefore, it seems reasonable to assume that if we managed to develop an algorithmic procedure for constructing such a subdivision, all the topological restrictions would be lifted and the swept technique would be ready for practical applications. Unfortunately, the following statement shows that this approach would still not lead to a complete resolution of the problem.

Theorem 3.3.2 *For any $\varepsilon > 0$ there is a smooth surface S such that every open subset of S of the form $S \cap B_\varepsilon(\mathbf{x})$, where B_ε is a ball of radius ε centered at $\mathbf{x} \in S$, is not regularly swept.*

We postpone the proof of this theorem until basic criteria concerning non-swept manifolds are considered. It is now evident, though, that even the best procedure for subdividing a manifold into swept patches can take an enormous amount of time, particularly if the manifold has a very complicated structure. Hence, being locally swept is not enough for practical purposes, but it does not mean that the *swept patch subdivision* procedure would not be useful – many of manifolds are comprised of only a few swept patches and therefore admit an applicable *differential equation representation*.

One of the most important results obtained for regular swept manifolds was the following global sweeping criterion, as provided by Theorems 3.2.7 and 3.2.9: *regular swept manifolds are piecewise diffeomorphic to $M \times [0, 1]$ or $M \times S^1$, where M is an object; and the converse is also true.* Unfortunately, the practical value of this statement is questionable. Indeed, how can one efficiently determine if two manifolds are diffeomorphic? Well, if the two manifolds are compact smooth two-dimensional surfaces, it suffices to compare their Euler characteristics ([43]); however, no such result is available in the general case. Despite this setback, we might still obtain some insight into possible sweeping criteria by employing the common tool used for analyzing and classifying smooth manifolds – Morse theory ([67, 43]). So far, elements of this theory were used only in proofs, but it is quite easy to reformulate some of the previous results using Morse functions as the primary focus.

Theorem 3.3.3 *Let M be a smooth submanifold of \mathbb{R}^n of codimension-2, and let $f : M \rightarrow [0, 1]$ be an admissible Morse function without critical points. Then M is a*

regular swept manifold diffeomorphic to $f^{-1}(a) \times [0, 1]$ with the generator

$$\left(f^{-1}(a), \frac{\text{grad } f}{\|\text{grad } f\|^2} \right)$$

The statement of the theorem, which is an adaptation of a standard result in differential topology, is essentially a more specific and extended version of Theorem 3.2.9 – not only does it state that a manifold is a regular swept manifold, but also provides a corresponding generator. Unfortunately, there still is an obstacle – it is not clear how to find the appropriate Morse function. The usual choice of one of the coordinate functions, $(x_1, x_2, \dots, x_n) \rightarrow x_k$, may often be unsatisfactory. Indeed, consider a surface, S , depicted in Figure 3.14. All three coordinate functions, $f_k(x_1, x_2, x_3) = x_k$, have critical points. Hence, Theorem 3.3.3 is not applicable to them. It is obvious, though, that the surface is a swept surface formed by sweeping a circle.



Figure 3.14 A regular swept surface that does not admit the coordinate functions as Morse functions without critical points.

Periodic swept manifolds constitute another difficulty for Morse functions – even the simplest regular periodic swept surface, a torus, does not admit a Morse function without critical points. The impossibility of finding the right function could be a disincentive for employing Morse theory, but we should notice that a non-existence

of the appropriate Morse function may serve as a criterion for being a non-swept manifold.

Lemma 3.3.4 *Let M be a smooth submanifold of \mathbb{R}^n such that every admissible Morse function $f : M \rightarrow \mathbb{R}$ has critical points. Then M is not a regular non-periodic swept manifold.*

The assumption of non-periodicity may seem too restrictive, but analyzing it more carefully one notices that it excludes only periodic swept manifolds without a boundary. This follows from the existence of equivalent generators (see Theorem 3.2.30). As an example, consider a cylinder. On one hand, it is generated by a periodic sweep of a line segment, but on the other hand, it is generated by a non-periodic sweep of a circle. Still, it would be nice to extend the above result to all regular swept manifolds, and we proceed to the construction of such an extension by making the following trivial observation: for any $\mathbf{x} \in S^1$ and any neighborhood $N_{\mathbf{x}} \subset S^1$ of \mathbf{x} , the set $S \setminus N_{\mathbf{x}}$ is diffeomorphic to $[0, 1]$. Recalling that every smooth periodic swept manifold is diffeomorphic to $M \times S^1$, where M is an object, we arrive at the following result.

Lemma 3.3.5 *Let S be a smooth periodic swept manifold without a boundary. Then there is a smooth submanifold, $M \subset S$, and a tubular neighborhood of M , $N \subset S$, such that the manifold $S \setminus N$ is a smooth non-periodic swept manifold.*

Proof. Let (Q, X) be a generator of S , and let M be a t_0 -section of S , that is, $M = \varphi_{t_0}(Q)$, where φ_t is the flow generated by X . We can choose t_0 and $\varepsilon > 0$ such that $N = \varphi_{(1-\varepsilon, 1)}(\varphi_{1-\varepsilon}(Q))$ is a tubular neighborhood of M . Then $S \setminus N = \varphi_{[0, 1-\varepsilon]}(Q)$. A simple reparametrization yields the required result. \square

Reversing the statement of this lemma and combining it with Lemma 3.3.4 yields the following non-swept criterion.

Theorem 3.3.6 *Let S be a smooth connected submanifold of \mathbb{R}^n . Then S is a regular swept manifold if and only if one of the following two conditions holds.*

1. *There is an admissible Morse function on S without critical points. In this case S is a non-periodic swept manifold.*
2. *There is a codimension-1 submanifold of S , $M \subset S$, such that for any tubular neighborhood $N \supset M$ in S the manifold $S \setminus N$ admits a Morse function without critical points. In this case S is a periodic swept manifold without a boundary.*

A straightforward application of this criterion may still be quite problematic, since proving non-existence of a specific Morse function is by no means a trivial task. Therefore, development of additional lemmas that would facilitate practical applications of the criterion would provide a valuable tool. Recognizing the importance of the issue, the author is planning to address it in his future research.

We finish this chapter by proving Theorem 3.3.2. The proof is based on the following easily proved corollary of Theorem 3.3.6.

Corollary 3.3.7 *Every smooth surface of genus $k > 1$ is not a regular swept surface.*

Proof of Theorem 3.3.2 A trivial example is given by a surface of genus greater than two with diameter less than ε . A surface of an arbitrary diameter is constructed by attaching handles to a sphere of the corresponding radius. Indeed, consider a sphere, S^2 . Given $\varepsilon > 0$, choose a collection of points $\{\mathbf{p}_i\}_{i=1}^n \subset S^2$ such that each subset of the form $S \cap B_\varepsilon(\mathbf{x})$, $\mathbf{x} \in S^2$, contains at least one \mathbf{p}_k that satisfies $d(\mathbf{p}_k, \mathbf{x}) < \delta_1$, where $\delta_1 = \varepsilon/8$. Let $\delta_2 = \min\{d(\mathbf{p}_i, \mathbf{p}_j) | 1 \leq i, j \leq n, i \neq j\}$, and define $\delta = \min\{\delta_1, \delta_2/8\}$. For each \mathbf{p}_i , $1 \leq i \leq n$, attach a handle to S^2 that lies completely inside $S^2 \cap B_\delta(\mathbf{p}_i)$. The resulting surface satisfies all the requirements of the theorem. \square

CHAPTER 4

INTERSECTION PROBLEM

When dealing with intersections of objects, it is impossible not to notice a drastic difference between the theoretical and practical viewpoints. For instance, defining a curve as the intersection set of two surfaces is a common practice, but being a neat and easy-to-use representation for theoretical purposes, it turns into a formidable problem rife with unexpected difficulties when it comes to practical applications. The inherent difficulty of the intersection problem, from the practical point of view, can be perceived by noticing that after the many years it has been scrutinized by numerous scholars there is still no general solution that can encompass the large spectrum of objects used in various applications. Moreover, it has become a ‘tradition’ to focus one’s attention on the intersection problem for a specific class of objects that possess some nice properties. Such properties are then used to get more insight into the problem and, possibly, obtain a better solution.

In this chapter, the main class of objects under consideration is the class of regular swept surfaces, and the main goal is to develop efficient algorithms for evaluating the intersection set of two such surfaces. To achieve this goal, we shall employ the properties of regular swept surfaces that were discussed and analyzed in the foregoing theoretical part. Before plunging into details, though, we review several existing methods for finding intersections. Besides providing a better understanding of the problem, such a review may, actually, furnish several useful techniques. Furthermore, we should not disregard the possibility that some of the extant intersection methods might be directly applicable to swept surfaces and therefore could save us from much of the thankless work involved in reinventing the wheel. At the end, we pro-

vide a novel unconventional approach to the intersection detection problem based on homology theory.

4.1 Overview of the Surface-to-Surface Intersection Problem

The surface-to-surface intersection (SSI) problem has a long and fascinating history, and can be traced as far back as 1960s (see [32]). It evolved from a minor task for designers to one of the most fundamental problems in computer aided design, computer animation, simulation and control of manufacturing processes, and many other important areas. The growth of popularity and significance of the SSI problem resulted in numerous algorithms and methods for finding its solution. Such methods often emphasized different aspects of the problem (see e.g. [7, 61, 90, 46]), and by now the number of different approaches is almost comparable to the total number of successful procedures. Not surprisingly, classifying all the intersection methods is a far from easy task, nevertheless, there are a few valuable surveys on the intersection methods where several possible classifications are discussed ([83, 45, 76]). Our description here follows [76]. First, we present basic concepts related to the SSI problem.

While the meaning of the SSI problem is intuitively clear, a proper mathematical treatment of the issue requires more than just intuition. The following definition provides some of the necessary rigor.

Problem 1 *Let $M \subset \mathbb{R}^3$ and $N \subset \mathbb{R}^3$ be two compact surfaces with smooth interior and either with a piecewise smooth boundary or without a boundary. Find the set $I = M \cap N = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} \in M \& \mathbf{x} \in N\}$.*

A key word in this definition is 'find'. Its meaning can be quite different depending on many aspects of the problem, such as representations of the surfaces, or later usage of the intersection set. Moreover, it is usually impossible to 'find' the intersection set exactly and therefore some approximating procedure is implicitly assumed. From a practical point of view, it is important to distinguish the following two cases: 1) the

intersection set, I , has to be computed in a closed form, that is, represented by some formula(e); 2) the intersection set has to be computed numerically, that is, represented by a discrete set of values. In recent years, due to an extensive growth of computational power, often cumbersome analytical methods lost their appeal to more simple and elegant numerical algorithms, which generally work fairly well. Unfortunately, the latter methods have their own shortcomings.

1. An intersection set usually consists of several components. Making sure that all such components are found is a formidable problem.
2. Tangential intersections may result in numerical instability. Therefore, such cases require a special treatment, which is a heavy burden for intersection algorithms.

Both of these problems have received a lot of attention in recent years; the first one even has its own name – loop detection ([90]). It is fair to say, though, that tangential intersections are a little less popular, mainly because they represent a non-generic situation¹. Therefore, it very tempting, and also quite reasonable, to start developing an intersection algorithm with the transversality condition imposed and handle tangential intersection later by adding additional steps and necessary amendments. Following the path of simplification, one also can make additional assumptions about the intersecting surfaces. Such assumptions usually narrow the class of surfaces under consideration. Currently, most methods are aimed at parametric surfaces, a fairly broad class, but performance of such algorithms is generally satisfactory for a much smaller family of surfaces. A good example is provided by Non Uniform Rational B-Splines (NURBS) ([80]); they have been, and still are, very popular in computer aided geometric design, and many algorithms have been developed to specifically handle NURBS patch intersections, having an ability to compute intersections of parametric surfaces as a byproduct. Nevertheless, many existing intersection methods are very

¹This follows from the standard result in differential topology stating that transverse maps are dense in the space of all smooth maps between two manifolds

successful, and it is imperative that before designing a new algorithm, features that enhanced the efficiency of the existing methods be identified and analyzed.

4.1.1 Intersection Methods

The goal of writing a survey, even a small one, of existing intersection methods is not as easily achievable as it may seem at the first glance. The mere number of the algorithms and procedures is astounding, and trying to figure out how to describe all the differences and similarities, advantages and drawbacks, and properties crucial for performance makes one quickly realize that a scale of the project is surprisingly large. Fortunately, many scholars have attempted such a venture over years, and now, as we have already mentioned, there are several good sources of valuable information. All of them provide some sort of a classification of intersection methods, and while such classifications may vary depending on a particular source, the basic ideas are usually quite similar. Here, intersection algorithms are subdivided into four major groups, as described in [76]: analytic methods, lattice evaluation methods, marching methods, and subdivision methods.

Analytic Methods The basic idea of this type of intersection method is, just as the name implies, to find an analytic expression of the intersection curve, usually in terms of an equation. Obviously, it is not feasible for arbitrary surfaces, but restricting ourselves to the class of algebraic surfaces rectifies the problem. The following is the standard setup, when the intersection of a rational parametric surface and an implicitly represented algebraic surface is sought. Let $\mathbf{f}(u, v)$ and $g(x, y, z) = 0$ represent a rational surface and an algebraic surface, respectively. Then

$$\mathbf{f}(u, v) = \left(\frac{P_1(u, v)}{Q_1(u, v)}, \frac{P_2(u, v)}{Q_2(u, v)}, \frac{P_3(u, v)}{Q_3(u, v)} \right), \quad g(x, y, z) = \sum_{i,j,k} a_{ijk} x^i y^j z^k$$

Substituting the expression for \mathbf{f} into the expression for g , we obtain

$$\sum_{i,j,k} a_{ijk} \frac{P_1^i(u,v)P_2^j(u,v)P_3^k(u,v)}{Q_1^i(u,v)Q_2^j(u,v)Q_3^k(u,v)} = 0$$

Multiplying by the common denominator and expanding all the polynomials yields

$$\sum_{p,q} b_{pq} u^p v^q = 0,$$

which is an implicit representation of the intersection curve.

A big drawback of this approach is that the resulting algebraic curve is usually of a very high degree. Resolving the topology of such a curve is a formidable task with no easy solution. Several methods that address this problem (see e.g. [3, 89]) produce accurate results ², but are too time consuming to be used in practice. Therefore, analytic methods are viable only when the intersection of low-degree algebraic surfaces has to be computed.

Lattice Evaluation Methods It is well known that reducing dimensionality of a geometric or topological problem very often leads to considerable simplifications, and this simple idea lies at the core of lattice evaluation methods, which transform the surface-to-surface intersection problem into a family of curve-to-surface intersection problems. Such a transformation is usually done by first discretizing one surface, obtaining a collection of isoparametric curves, and then computing intersections of these curves with the other surface. The final step is a merging procedure that connects the intersection points into curves.

It is not difficult to describe this type of methods in more detail if we assume that both intersecting surfaces are represented parametrically. Indeed, let $\mathbf{f}(u,v)$ and $\mathbf{g}(s,t)$ be two parametric surfaces with the domain $[0,1] \times [0,1]$. Choosing a

²Rational arithmetic is assumed.

discretization step, $h = 1/N$, a family of isoparametric curves of \mathbf{f} is defined by

$$\mathbf{c}_i(u) = \mathbf{f}(u, ih), \quad 0 \leq i \leq N.$$

To find the intersection of the curve \mathbf{c}_k and the surface $\mathbf{g}(s, t)$, the following system of nonlinear equations has to be solved

$$\mathbf{c}_k(u) = \mathbf{g}(s, t).$$

Thus, the intersection of the \mathbf{f} and \mathbf{g} surfaces can be approximately computed by solving N independent nonlinear systems of equations. Merging of the intersection points is usually performed in an ad-hoc manner and generally uses distances to determine different connected components of the intersection curve.

Remark 4.1.1 *If the second surface is represented implicitly instead of parametrically, say $g(x, y, z) = 0$, then one will obtain N independent nonlinear equations*

$$g(c_k^1(u), c_k^2(u), c_k^3(u)) = 0,$$

where $c_k^i(u)$, $i = 1, 2, 3$, are vector components of the curve $\mathbf{c}_k(u)$.

The main drawback of the lattice evaluation methods is that it is not clear how to choose the initial discretization step. If the step is too big, small loops and tangential points in the intersection may be lost, while making it too small considerably slows down performance of the algorithm. In addition, solving nonlinear systems of equations can be a formidable problem in itself. Fortunately, if intersecting surfaces are algebraic and of a relatively low degree, quite a few efficient solvers are available ([60, 84, 97]).

Marching Methods Consider the following restricted intersection problem: given a point in the intersection set of two surfaces, compute the corresponding connected

component. Such a setup obviously simplifies a lot of things, and it is no surprise that quite a few very efficient algorithms for solving this problem have been developed. The basic idea of such algorithms is to move from one intersection point to another by analyzing the local geometry and topology of the surfaces. A sequence of intersection points thus obtained is then used to approximate the intersection component.

Customarily, a sequence of intersection point is generated by solving a system of differential, or mixed algebraic differential equations. Such a system is particularly easy to derive when the intersecting surfaces are smooth, or at least C^1 , because a general expression for a tangent vector to the intersection curve at a given point is available. To see this, consider two C^1 parametric surfaces $\mathbf{f}(u, v)$ and $\mathbf{g}(s, t)$. If $\mathbf{x} = \mathbf{f}(u, v)$, the the unit normal vector to the surface \mathbf{f} at \mathbf{x} is given by

$$\mathbf{n}_1(\mathbf{x}) = \frac{\mathbf{f}_u(u, v) \times \mathbf{f}_v(u, v)}{\|\mathbf{f}_u(u, v) \times \mathbf{f}_v(u, v)\|}.$$

Similarly,

$$\mathbf{n}_2(\mathbf{y}) = \frac{\mathbf{g}_s(s, t) \times \mathbf{g}_t(s, t)}{\|\mathbf{g}_s(s, t) \times \mathbf{g}_t(s, t)\|}$$

is the unit normal vector to the surface \mathbf{g} at $\mathbf{y} = \mathbf{g}(s, t)$. Thus, if \mathbf{x} is a transverse intersection point of \mathbf{f} and \mathbf{g} , then a tangent vector to the intersection curve at \mathbf{x} is given by

$$\tau(\mathbf{x}) = \mathbf{n}_1(\mathbf{x}) \times \mathbf{n}_2(\mathbf{x}).$$

These results can be summarized as follows. If \mathbf{x}_0 is a point in the intersection set of two transverse parametric surfaces, \mathbf{f} and \mathbf{g} , then the the corresponding connected component is a solution curve of the equation

$$\frac{d\mathbf{x}}{dt} = \tau(\mathbf{x}),$$

where $\tau(\mathbf{x})$ is defined as above.

Thus, it appears that an intersection component can be approximated quite efficiently if a starting point is given. Unfortunately, finding such a starting point is a non-trivial problem. Commonly, it is tackled by employing a loop detection technique, or a lattice evaluation or subdivision method; in either case it takes a considerable amount of time. Furthermore, to ensure an accurate result in tracing an intersection component may need a variable step size, which generally depends on the curvature of the surfaces at the intersection point. Finally, marching methods may exhibit bad behavior around tangential intersection points, as illustrated in Figure 4.1. One remedy to this problem can be computing all significant points³ of the intersection curve in advance, although no general procedure is available.

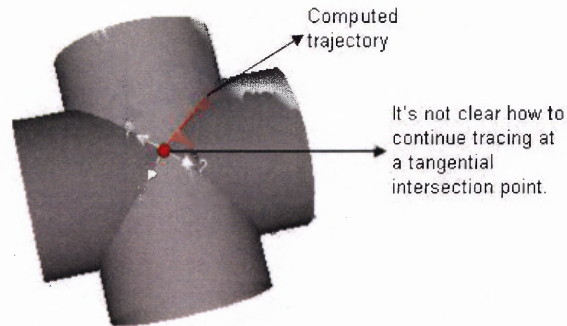


Figure 4.1 Tracing an intersection curve around a tangential intersection point.

Subdivision Methods Algorithms in this group employ one of the oldest techniques to achieve a suitable level of simplicity – divide and conquer. The idea is based on the fact that surfaces are locally flat, which means that on a very small scale they can be treated just like planes. Finding the intersection of two planes is a trivial problem, and reducing the general surface-to-surface intersection problem to a series of linear intersection problems may be quite practical. This is exactly the viewpoint of subdivision methods – they recursively decompose intersecting surfaces

³Significant points are singular, boundary, and turning points.

into smaller patches until the desired degree of flatness is achieved. During decomposition, trivially disjoint patches are identified and discarded, and at the last level, the actual intersection is computed; the later computation is almost trivial because of the flatness of the intersecting patches. The concluding step of a subdivision method is to merge all computed intersection elements into complete intersection curves.

There are two crucial steps in the above description: recursive decomposition and filtering, that is, identifying and discarding non-intersecting patches. The first one, besides a proper technique, requires a stopping criteria; an inappropriate choice of such criteria may lead to a significant decline in performance. The second step presents a whole new problem in itself – the intersection detection problem.

Among several possible approaches to recursive decomposition, the following two have become the most popular: a uniform decomposition, and an adaptive decomposition. To illustrate how they work, let us consider two parametric surfaces, \mathbf{f} and \mathbf{g} , with the parameter domain $[0, 1] \times [0, 1]$. The uniform decomposition subdivides each of the intervals $[0, 1]$ into N equal parts, thus producing N^2 smaller squares with the corresponding surface patches. After discarding disjoint patches, each of the remaining squares is again subdivided into N^2 parts, and so on. The adaptive decomposition, on the other hand, starts by estimating the curvature of the surfaces. Very often such estimates can be quite crude, although it generally depends on the context. Once the curvature is computed, a subdivision is performed according to a specified rule which usually asserts that a finer decomposition is needed around points with a large curvature. For example, one may proceed in the following manner: perform a uniform decomposition, then analyze the curvature around each of the nodes and, if necessary, perform additional subdivision of adjacent squares into k parts, where k is proportional to the ratio between the value of the curvature at the node and some specified threshold. In this procedure a value of k less than one indicates that the squares around the node should be merged. The final part of a

recursive decomposition, a stopping criterion, is usually the same for both uniform and adaptive decompositions – variation of some standard measure of the curvature in each surface patch should be less than some threshold value.

The intersection detection problem is generally quite complicated, but in the context of subdivision methods it is far less severe due to the fact that even a very crude estimate may suffice. Indeed, it turns out that the simple concept of a bounding box solves the problem quite nicely. A bounding box can be of several possible shapes, such as a sphere, a convex hull, or a regular axis-oriented bounding box with each shape providing a different compromise between efficiency and accuracy. An actual computation of a bounding box is done by estimating minimum and maximum values of the surfaces over the corresponding part of the domain, and then increasing the the size by a predefined factor to ensure a full enclosure.

4.1.2 Geometric and Topological Consistency of Intersection Algorithms

Let us begin with a simple example. Consider two curves, P and Q , shown in Figure 4.2. There is no question that these curves are very similar to each other. In fact,

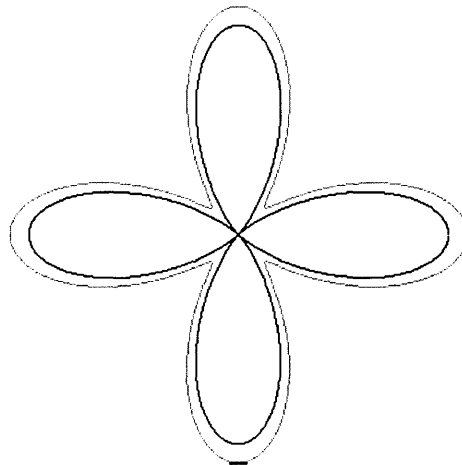


Figure 4.2 Example of geometrically close but topologically different curves.

any reasonable metric defined on subsets of \mathbb{R}^2 , for instance, the Hausdorff metric,

would justify this similarity. There is one significant difference, though – the inner curve, P , is self-intersecting. Such a feature cannot be measured by any geometric tools because it is topological in its essence. Therefore, topological methods should be employed to capture the difference between the two curves. In this case, it suffices to compute the corresponding fundamental groups: $\pi_1(P) = \mathbb{Z} * \mathbb{Z} * \mathbb{Z} * \mathbb{Z}$, $\pi_1(Q) = \mathbb{Z}$. Thus, the curves are not similar at all from the topological perspective.

An example opposite, in some sense, to the previous one is illustrated in Figure 4.3. Here, the two curves, M and N , are topologically the same, but their geometric difference is striking.

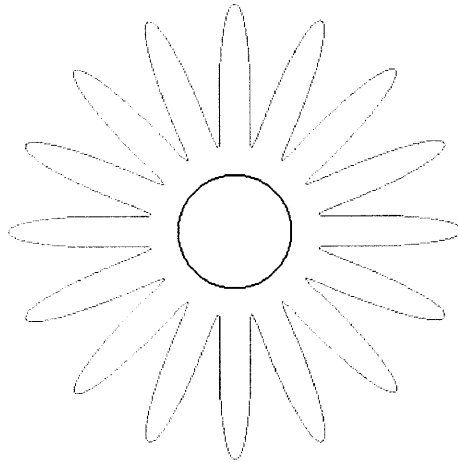


Figure 4.3 Example of topologically equivalent but geometrically different curves.

Consider now an intersection algorithm whose output is an approximation of the exact intersection curve of two surfaces. Obviously, it is desired that this approximation be as close to the exact solution as possible. But how should we measure ‘closeness’? The previous examples show that metric measurements are not sufficient and at least a combination of geometric and topological tools is needed. Thus, assessment of the quality of an algorithm should include two distinct concepts, which we shall call geometric and topological consistency. To provide rigorous definitions, let us denote an intersection algorithm by $\mathcal{A}(\sigma)$, where σ is a parameter vector; this

vector can include, for example, a discretization step and/or flatness tolerance. The exact intersection of surfaces S_1 and S_2 and the corresponding output of an algorithm $\mathcal{A}(\sigma)$ shall be denoted by $\mathcal{O}(S_1, S_2)$ and $\mathcal{O}_{\mathcal{A}(\sigma)}(S_1, S_2)$, respectively. Finally, let m be the Hausdorff metric ⁴ defined on subsets of \mathbb{R}^3 , that is,

$$\begin{aligned} m(A, B) &= \max \{m_*(A, B), m^*(A, B)\}, \text{ where} \\ m^*(A, B) &= \inf \{r > 0 | B \subset \cup_{\mathbf{x} \in A} B(\mathbf{x}, r)\}, \\ m_*(A, B) &= \inf \{r > 0 | A \subset \cup_{\mathbf{x} \in B} B(\mathbf{x}, r)\}, \text{ and} \\ B(\mathbf{x}, r) &= \{\mathbf{y} | \|\mathbf{x} - \mathbf{y}\| < r\}. \end{aligned}$$

Definition 4.1.2 *A surface intersection algorithm, $\mathcal{A}(\sigma)$, is called geometrically consistent if for any admissible pair of surfaces, S_1 and S_2 , and any $\epsilon > 0$ there exist a parameter vector σ such that $m(\mathcal{O}_{\mathcal{A}(\sigma)}(S_1, S_2), \mathcal{O}(S_1, S_2)) < \epsilon$.*

Topological consistency is defined as follows.

Definition 4.1.3 *A surface intersection algorithm, $\mathcal{A}(\sigma)$, is called topologically consistent if for any admissible pair of surfaces, S_1 and S_2 , there exist a parameter vector σ such that $\mathcal{O}_{\mathcal{A}(\sigma)}(S_1, S_2)$ and $\mathcal{O}(S_1, S_2)$ are homeomorphic.*

It is worth mentioning that defining topological consistency via homeomorphism works only for surface-to-surface intersections, and a different approach should be taken in the case of higher dimensional manifolds.

Let us now look back at the different types of intersection methods and see which of them are geometrically and/or topologically consistent.

Analytic methods are both geometrically and topologically consistent. Unfortunately, the price for this is efficiency.

Lattice evaluation methods are geometrically and topologically consistent in the case of transverse intersections. For some methods, though, geometric consistency requires only the absence of one-point tangential intersections.

⁴Any other metric also can be used.

Marching methods also guarantee topological and geometric consistency for transverse intersections. Although, there are modifications that are geometrically and topologically consistent for some tangential intersections.

Subdivision methods have the same consistency characteristics as lattice evaluation methods, but may be more efficient in some cases.

While the foregoing description gives some insight into the quality and performance of intersection algorithms, there is a subtle property that should be emphasized separately – geometric and topological consistency of an intersection algorithm may substantially depend on the class of admissible surfaces and the way the surfaces are represented. For example, if an intersection algorithm is based on solving systems of nonlinear equations, having different classes of functions, such as algebraic or transcendental, may lead to different consistency properties. Our goal here is to develop intersection algorithms for the entirely new class of swept surfaces, and therefore additional analysis may be required to investigate consistency characteristics of the new intersection algorithms.

4.2 Swept Manifold Intersections

It is now time that the many properties of swept surfaces that have been investigated in the previous chapter be combined with the basic ideas of extant intersection methods to generate a family of new intersection algorithms that can be successfully applied to swept surfaces. As we have mentioned, the novelty of the class of surfaces may create some obstacles for a direct application of existing intersection ideas. Indeed, the absence of an exact analytic representation for swept surfaces not only complicates a possible algorithmic procedure needed to solve the problem, but also affects geometric and topological consistency characteristics of any such algorithm. As an example, consider solving a system of differential equations in order to compute some points of a swept surface. Since it generally has to be done numerically, an approximation error is inevitable. An intersection algorithm may need to perform

millions of such computations, and there is a possibility that error accumulation will distort its output to an unacceptable degree. Therefore, it is crucial that swept surfaces are rendered with a precision that makes any potential error accumulation quite small.

4.2.1 Rendering Swept Surfaces

Most computer applications that deal with geometric objects have to do some type of rendering before displaying the objects on a screen, that is, a possibly continuous representation of an object maintained internally has to be transformed into a piecewise linear approximation, or triangulation, that can be directly fed into a graphics processor. For most objects represented analytically, such a procedure has become standard and can be performed accurately and efficiently. Unfortunately, swept surfaces possess a different kind of representation which is based on an initial value and two vector fields, (\mathbf{x}_0, X, Y) , the triple that is called a generator of the swept surface. Although it may seem trivial to generate a piecewise linear approximation of a swept surface having its generator, some care has to be taken to ensure a correct output. Consider, for instance, a generalized swept surface shown in Figure 4.4, where a self-intersection is clearly seen. Surprisingly, Figure 4.5 shows the same surface, but a bad rendering resulted in loss of the self-intersection, making the output topologically invalid.

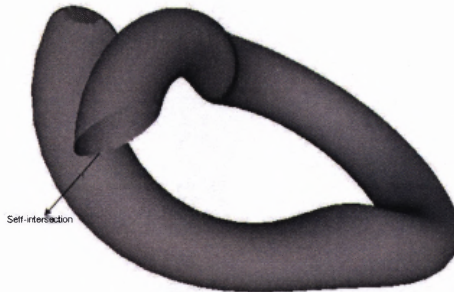


Figure 4.4 Well rendered self-intersecting generalized swept surface.



Figure 4.5 Badly rendered self-intersecting generalized swept surface.

Self-intersections of generalized swept surfaces can be trivially reduced to swept surface intersections, and the above example, as well as the foregoing comment about error accumulation, shows the importance of a proper rendering. Incidentally, the example also raises another curious question: Is it always possible to obtain a topologically and geometrically valid piecewise linear approximation of a swept surface? It clearly may not be true for arbitrary *generalized* swept surfaces, since if there is a point of tangential self-intersection, the rendering problem becomes as intricate as the intersection problem. Fortunately, the simplicity of regular swept surfaces leads to a positive answer. Let us denote by S_{PL} a piecewise linear approximation of a swept surface S ; we may also use the notation $S_{PL}(h)$ or $S_{PL}(h_1, h_2)$ if explicit dependence on discretization steps for the differential equations has to be indicated. The following theorem shows that regular non-periodic swept surfaces can always be rendered nicely.

Theorem 4.2.1 *Let S be a regular non-periodic swept surface with a generator (\mathbf{x}_0, X, Y) . Then for any $\epsilon > 0$ there is a choice of discretization steps, h_1 and h_2 , for differential equations*

$$\frac{d\mathbf{x}}{dt} = X(\mathbf{x}) \quad \text{and} \quad \frac{d\mathbf{y}}{dt} = Y(\mathbf{y}),$$

respectively, such that we may produce a triangulation $S_{PL}(h_1, h_2)$ which is homeomorphic to S and $m(S, S_{PL}(h_1, h_2)) < \epsilon$, where $m(\cdot, \cdot)$ is the Hausdorff metric.

Proof. Since the swept surface, S , is a compact oriented surface, there exists a finite open cover, $\mathcal{U}_\epsilon = \{B(\mathbf{x}_i, \epsilon/2)\}_{i=1}^n$, where $B(\mathbf{x}_i, \epsilon) = \{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) < \epsilon/2\}$ and $\mathbf{x}_i \in S$. Now let δ be the minimal diameter of the sets $S_i = S \cap B(\mathbf{x}_i, \epsilon/2)$, $1 \leq i \leq n$, that is

$$\delta = \min \{diam(S_i) | 1 \leq i \leq n\}, \quad diam(A) = \sup_{\mathbf{x}, \mathbf{y} \in A} d(\mathbf{x}, \mathbf{y}).$$

Choose h_1 and h_2 smaller than $\delta/3$, and suppose we employ a numerical scheme of order $k > 1$. It is easy to see that the approximation, C_{PL} , of the initial curve, C , computed with the step h_1 , and all the approximations, S_{PL}^m , $1 \leq m \leq [1/h_1]$, of trajectories emanating from C lie within $U_\epsilon = \cup_{\mathbf{x} \in S} B(\mathbf{x}, \epsilon/2)$. Construct the triangulation, $S_{PL} = S_{PL}(h_1, h_2)$, by the following rule: if $\mathbf{x}_{i,j}, \mathbf{x}_{i,j+1}, \mathbf{x}_{i+1,j}, \mathbf{x}_{i+1,j+1}$ are the j -th and $(j+1)$ -th computed points of the trajectories S_{PL}^i and S_{PL}^{i+1} , add the triangles defined by $(\mathbf{x}_{i,j}, \mathbf{x}_{i,j+1}, \mathbf{x}_{i+1,j})$ and $(\mathbf{x}_{i,j}, \mathbf{x}_{i+1,j}, \mathbf{x}_{i+1,j+1})$ to the triangulation. The choice of δ guarantees that each of these triangles lies within U_ϵ . Moreover, it also follows that S lies within $\cup_{\mathbf{x} \in S_{PL}} B(\mathbf{x}, \epsilon/2)$, which implies that $m(S, S_{PL}) < \epsilon$.

If ϵ is sufficiently small, a homeomorphism between S and S_{PL} is given by the projection $\pi : S_{PL} \rightarrow S$ along the normal vectors to S . \square

It may be a little surprising and disappointing to see regular periodic swept surfaces excluded from the theorem. Unfortunately, the straightforward approach employed in the proof of the theorem does not allow us to obtain a topologically correct triangulation in the case of periodicity. To see this, consider computation of a periodic orbit of some differential equation. Even the best numerical procedure can guarantee only that the end point of the computed curve will fall within some small neighborhood of the initial point, and exact coincidence cannot be ensured. Therefore, a numerically computed image of the initial curve under a periodic sweep may not match with it at the end, resulting in a topological flaw. The problem can be remedied, though, if it

is known a priori that the swept surface is periodic. Indeed, topological discrepancies can be fixed in this case by adding the initial curve as the last numerically computed section. Even better accuracy can be achieved by performing computation in both directions with respect to the periodic vector field and ‘gluing’ the end images together. Both approaches are illustrated in Figure 4.6, and the following theorem provides a rigorous proof.

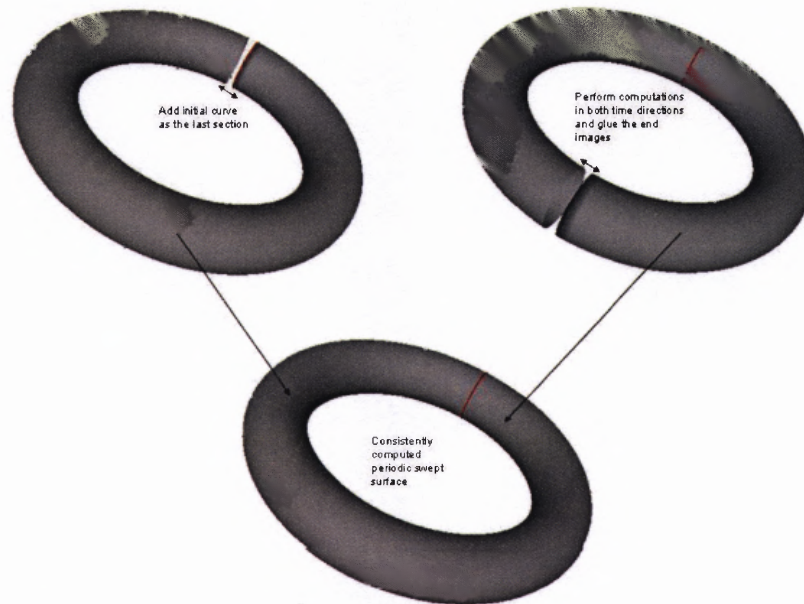


Figure 4.6 Two approaches to rendering periodic swept surfaces.

Theorem 4.2.2 *Let S be a regular periodic swept surface with a generator (\mathbf{x}_0, X, Y) . If the fact of periodicity is known a priori, then for any $\epsilon > 0$ there is a choice of discretization steps, h_1 and h_2 , for the differential equations*

$$\frac{d\mathbf{x}}{dt} = X(\mathbf{x}) \quad \text{and} \quad \frac{d\mathbf{y}}{dt} = Y(\mathbf{y}),$$

respectively, such that we may produce a triangulation $S_{PL}(h_1, h_2)$ which is homeomorphic to S and $m(S, S_{PL}(h_1, h_2)) < \epsilon$, where $m(\cdot, \cdot)$ is the Hausdorff metric.

Proof. Repeating the procedure described in the proof of Theorem 4.2.1, we obtain a triangulation $S_{PL} = S_{PL}(h_1, h_2)$ such that $m(S, S_{PL}) < \epsilon$. To make S_{PL} periodic, we

augment it by adding the triangles defined by $(\mathbf{x}_{n,j}, \mathbf{x}_{n,j+1}, \mathbf{x}_{1,j})$, $(\mathbf{x}_{1,j}, \mathbf{x}_{1,j+1}, \mathbf{x}_{n,j})$, $(\mathbf{x}_{n,m}, \mathbf{x}_{n,1}, \mathbf{x}_{1,m})$, and $(\mathbf{x}_{1,m}, \mathbf{x}_{1,1}, \mathbf{x}_{n,m})$, where $1 \leq j \leq m$. Here, $n = \lceil 1/h_2 \rceil$, $m = \lceil 1/h_1 \rceil$. It is clear that the resulting triangulation will still satisfy $m(S, S_{PL}) < \epsilon$ and will be homeomorphic with S via the projection $\pi : S_{PL} \rightarrow S$ along the normal vectors to S . \square

Continuing in this vein, we may also try to extend our findings to critical swept surfaces. In fact, the proofs of the previous theorems need little modification to become quite suitable for critical swept surfaces with only one critical point – the initial point \mathbf{x}_0 . Notice, though, that the difficulty arising when the other boundary point of an initial curve is critical is very similar to the one that occurs for periodic swept surfaces – numerical computation of the initial curve may perturb the second endpoint making it non-critical, thus leading to an erroneous result. But again, the problem can be easily rectified if the fact that the second boundary point is critical is known a priori. For example, we have the following result that can be readily proved using the methods in the proofs of Theorems 4.2.1 and 4.2.2.

Theorem 4.2.3 *Let S be a critical swept surface with a generator (\mathbf{x}_0, X, Y) and an open initial curve. If the only critical point of Y is \mathbf{x}_0 , or if it is known a priori that $\varphi_1(\mathbf{x}_0)$ is critical, then for any $\epsilon > 0$ there is a choice of discretization steps, h_1 and h_2 , for differential equations*

$$\frac{d\mathbf{x}}{dt} = X(\mathbf{x}) \quad \text{and} \quad \frac{d\mathbf{y}}{dt} = Y(\mathbf{y}),$$

respectively, such that we may produce a triangulation $S_{PL}(h_1, h_2)$ which is homeomorphic to S and $m(S, S_{PL}(h_1, h_2)) < \epsilon$, where $m(\cdot, \cdot)$ is the Hausdorff metric.

It is worth mentioning that sometimes an initial curve may be represented parametrically. In this case, all the previous result about rendering remain valid

and even can be improved. Indeed, for a parametrically represented curve it can be algorithmically determined whether or not its endpoints are critical. Therefore, there is no need for a priori knowledge.

4.2.2 Simple Discretization Algorithm

As the name suggests, this method is very straightforward. The basic idea is to triangulate given swept surfaces and then use computed triangulations to approximate the intersection. If S_1 and S_2 are given swept surfaces, the algorithm proceeds as follows.

1. Choose discretization steps, $\Delta s_1, \Delta t_1$ for the first surface, S_1 , and $\Delta s_2, \Delta t_2$ for the second surface, S_2 ;
2. Compute triangulations, $S_{PL}(\Delta s_1, \Delta t_1)$ and $S_{PL}(\Delta s_2, \Delta t_2)$;
3. Find the intersection of the computed triangulations, $S_{PL}(\Delta s_1, \Delta t_1) \cap S_{PL}(\Delta s_2, \Delta t_2)$, and report it as an approximation of the intersection $S_1 \cap S_2$.

Justification for this algorithm is essentially based on the results obtained in the previous section and is presented as the following result that is easy to prove using the methods developed above.

Theorem 4.2.4 *Let swept surfaces S_1 and S_2 satisfy the conditions of one of the Theorems 4.2.1- 4.2.3 and intersect transversally. Then for any $\epsilon > 0$ there exist discretization steps, $\Delta s_1, \Delta t_1$ and $\Delta s_2, \Delta t_2$, and triangulations, $S_{PL}^1(\Delta s_1, \Delta t_1)$ and $S_{PL}^2(\Delta s_2, \Delta t_2)$, such that $S_{PL}^1(\Delta s_1, \Delta t_1) \cap S_{PL}^2(\Delta s_2, \Delta t_2)$ is homeomorphic to $S_1 \cap S_2$ and $m(S_{PL}^1(\Delta s_1, \Delta t_1) \cap S_{PL}^2(\Delta s_2, \Delta t_2), S_1 \cap S_2) < \epsilon$, where $m(\cdot, \cdot)$ is the Hausdorff metric.*

It is now evident that despite its simplicity the algorithm is applicable, at least in theory, to virtually any pair of transverse swept surfaces. There are, however, several practical issues that require more thorough analysis. Consider, for instance,

the first step of the algorithm. While Theorem 4.2.4 guarantees that there is a suitable choice of discretization steps, it does not provide any constructive ideas. Therefore, some empirical procedure should be applied in this case, which may lead to a noticeable reduction in the efficiency of the algorithm. Also, as we shall see, the third step poses a serious efficiency problem.

Complexity of the Algorithm Let us start with the second step of the algorithm. Triangulations of the surfaces S_1 and S_2 is constructed by solving ordinary differential equations, and it is well known that a trajectory of an ODE can be easily computed in a time proportional to the reciprocal of the corresponding discretization step ([84]). The surfaces S_1 and S_2 require a computation of the initial curves with the steps Δs_1 and Δs_2 , respectively, followed by a computation of trajectories emanating from the initial curves. The number of such trajectories is Δs_1^{-1} , for the first initial curve, and Δs_2^{-1} , for the second initial curve. The trajectories themselves are computed with the steps Δt_1 and Δt_2 , respectively. Therefore, the total time needed to construct triangulations of S_1 and S_2 is $O(\Delta s_1^{-1} \Delta t_1^{-1} + \Delta s_2^{-1} \Delta t_2^{-1})$. Incidentally, the amount of space needed is also $O(\Delta s_1^{-1} \Delta t_1^{-1} + \Delta s_2^{-1} \Delta t_2^{-1})$, since the number of triangles in each triangulation is proportional to the number of points in the trajectories.

Analysis of the first step of the algorithm is a little harder, primarily due to the fact that no specific way of choosing the discretization steps is given. One possibility could be to crudely estimate the curvature of each surface and then make the choice based on the findings. Such an estimate can be obtained in linear time, that is, $O(\Delta \tilde{s}_1^{-1} + \Delta \tilde{t}_1^{-1})$ and $O(\Delta \tilde{s}_2^{-1} + \Delta \tilde{t}_2^{-1})$ for S_1 and S_2 , respectively. Indeed, approximations of the derivatives of the corresponding right-hand sides should be computed only at some fixed points that correspond to nodes of a grid in a parameter space. Such points can be easily found by computing a constant number of trajectories. Notice

that the choice of the steps $\Delta\tilde{s}_1$, $\Delta\tilde{t}_1$ and $\Delta\tilde{s}_2$, $\Delta\tilde{t}_2$ is not crucial and any reasonable values can be used.

The third step of the algorithm can be implemented using the following straightforward approach. For each triangle in one triangulation find all triangles in the other triangulation that intersect it and construct the corresponding part of the intersection. The number of triangles in triangulations for S_1 and S_2 is $O(\Delta s_1^{-1} \Delta t_1^{-1})$ and $O(\Delta s_2^{-1} \Delta t_2^{-1})$, respectively, which yields the time complexity of $O(\Delta s_1^{-1} \Delta t_1^{-1} \Delta s_2^{-1} \Delta t_2^{-1})$. The space complexity depends on the number of segments in the intersection curve and usually is linear. In any case, it is always less than quadratic, that is, $O(\Delta s_1^{-1} \Delta t_1^{-1} + \Delta s_2^{-1} \Delta t_2^{-1})$. Adding up the complexities of each step, the algorithm turns out to be $O(\Delta s_1^{-1} \Delta t_1^{-1} \Delta s_2^{-1} \Delta t_2^{-1})$ in time and $O(\Delta s_1^{-1} \Delta t_1^{-1} + \Delta s_2^{-1} \Delta t_2^{-1})$ in space.

Pros and Cons The algorithm has one big advantage – simplicity. Its implementation is fast and straightforward; no external libraries or modules have to be employed. Unfortunately, as it is usually the case in algorithm design, simplicity leads to loss of efficiency. Quartic order in time is very slow when there is a need for high accuracy. A speedup can be achieved by employing one of the existing methods for finding polygonal surfaces intersections ([74]). Not surprisingly, the procedure would become more complicated in this case, but it may be well worth it, especially if the intersecting surfaces are convex ([74, 38]).

Another drawback of the algorithm is its inability to deal with tangential intersections. This problem cannot be remedied, and the only consoling fact is that non-transverse intersections do not happen often; more precisely, transversality is a generic property.

4.2.3 Subdivision Algorithm

Consider two swept surfaces S_1 and S_2 shown in Figure 4.7. Suppose the simple discretization method is used to compute their intersection. It is clear that triangles

in the shaded regions do not intersect, but the algorithm will spend most of its time checking those very triangles for possible intersection. The question arises: Is it possible to discard the shaded parts of the surfaces before the intersection is computed? If we succeed in adopting ideas from subdivision methods the answer is definite ‘yes’. Indeed, constructing bounding boxes for the shaded regions, as illustrated in Figure 4.8, provides an easy test for possible intersections significantly improving the overall performance.

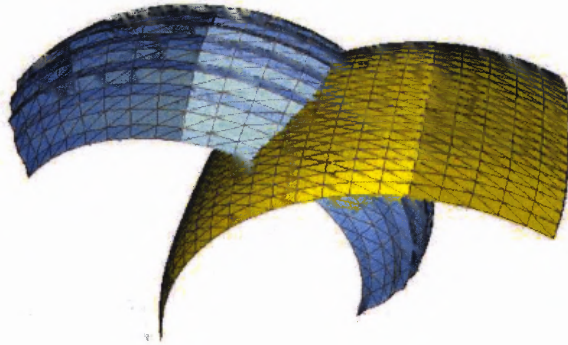


Figure 4.7 Intersecting surfaces.

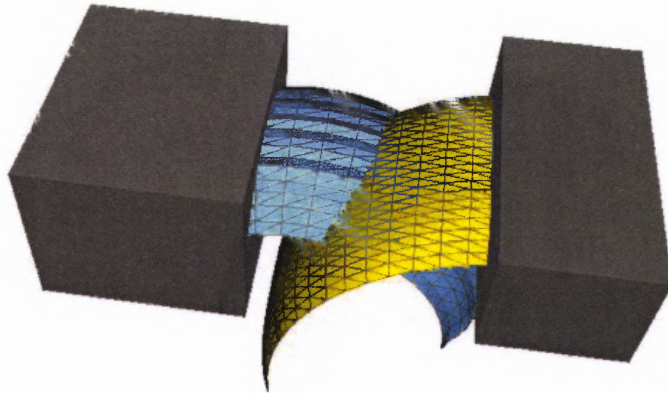


Figure 4.8 Example of employment of bounding boxes for discarding trivially disjoint parts of surfaces.

Unfortunately, fitting the subdivision methods procedure into the swept surface framework is not as easy as it might seem. Even the very first step – decomposing

a surface – poses a difficulty, since a simple decomposition of the domain is very inefficient in this case. A source of the obstacle lies in the underlying representation – computing a point on a swept surface corresponding to a node on the grid requires solving a differential equation (recall the first step of the simple discretization algorithm). While it may be acceptable if done only once, a recursive decomposition is not practically feasible if approached directly. A remedy to the problem can be found in an additional assumption, which is based on the concept of a dual vector field.

Recall from Section 3.2.3 that each sweep vector field, X , generating a swept surface, S , has its dual – the vector field, X^* , that is transverse to X and together with it defines the tangent space of S . Computation of X^* from X can be time consuming, but having the dual vector field can significantly improve a swept surface decomposition procedure. Therefore, the following assumption is made: swept surfaces for the subdivision algorithm are represented by two dual generators.

Let S_1 and S_2 be swept surfaces represented by pairs of dual generators, (\mathbf{x}_0, X_1, Y_1) and $(\mathbf{x}_0, Y_1, X_1^*)$, and (\mathbf{y}_0, X_2, Y_2) and $(\mathbf{y}_0, Y_2, X_2^*)$. The basic steps of the subdivision algorithm are as follows.

1. Decompose the surfaces S_1 and S_2 into patches S_1^{ij} , $1 \leq i \leq I$, $1 \leq j \leq J$, and S_2^{kl} , $1 \leq k \leq K$, $1 \leq l \leq L$.
2. Construct bounding boxes, B_1^{ij} and B_2^{kl} for the patches S_1^{ij} and S_2^{kl} , respectively.
3. Determine bounding boxes from each collection that do not meet any bounding box from the other collection and discard the corresponding patches.
4. Recursively apply the above steps to the remaining patches until the estimated curvature of all patches is less than some given threshold value.
5. Construct crude triangulations of the remaining patches and find their intersections.
6. Merge intermediate intersections pieces into one curve and report it as an approximation of the intersection set.

To fully understand how the algorithm works and properly assess it, additional explanation may be required for some of the steps. In particular, employment of the dual vector fields in the first step should be elaborated. The key observation here is that having dual generators of a swept patch and a good approximation of its boundary allows one to construct a subdivision of the patch where the boundaries of the sub-patches are approximated with the same precision. To see this, consider a swept patch in Figure 4.9a. Suppose its boundary has been computed using a fine discretization step(s). A subdivision can be done by choosing a set of points on the boundary and computing the corresponding trajectories using the same discretization step(s), as illustrated in Figure 4.9b. Notice that having two dual vector fields is crucial – in addition to providing efficient computations in both transverse directions they guarantee accurate approximation of the initial boundary.

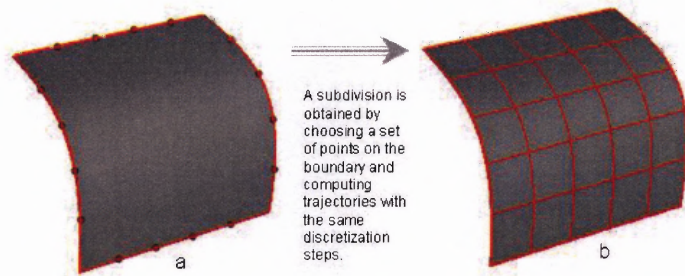


Figure 4.9 Subdivision of a swept patch.

The second step of the algorithm is fairly simple, especially when bounding boxes are chosen to be axis-oriented. Assuming that this is the case, pick a swept patch and estimate its maximum and minimum coordinates, x_{min} , x_{max} , y_{min} , y_{max} , and z_{min} , z_{max} . To improve accuracy, intermediate trajectories may be computed. Then the bounding box can be defined by the points $(x_{min} - \delta, y_{min} - \delta, z_{min} - \delta)$, and $(x_{max} + \delta, y_{max} + \delta, z_{max} + \delta)$, where δ is a small percentage of $\max\{x_{max} - x_{min}, y_{max} - y_{min}, z_{max} - z_{min}\}$ and is used to guarantee inclusion.

Identifying non-intersecting bounding boxes – the third step of the algorithm – is a problem that can be solved in many different ways. The simplest one is just to browse through bounding boxes in one collection and check whether they intersect bounding boxes in the other collection. If yes, the intersecting box from the second collection is marked, if no, the intersection box from the first collection is deleted. After checking all the boxes in the first collection, a final parsing through the second collection is done to remove unmarked boxes. Obviously, this naive approach is far from being fast and efficient, but it may be applicable in some simple cases. A substantial speedup can be achieved if a special algorithm for finding box intersections developed by Edelsbrunner and Zomorodian is employed ([98]).

A few details in the last step of the algorithm also need to be addressed. The difficulty here is in the absence of a good merging criterion. Commonly, two pieces of the intersection curve are joined if their ends fall within a small distance, perhaps, a machine epsilon. There is a drawback in such an approach, however, which will be discussed later. The process of finding two adjacent curve segments is also not trivial, since examining every pair for possible connection is very inefficient. An improvement is made by exploiting the recursive nature of the algorithm – every patch is stored as a node of a tree, where children of each node represent patches in the corresponding decomposition. Since most connecting curve segments will belong to the same subtree, such a structure provides a much faster way to perform merging. A tree where each node can have a different number of children is a standard data structure called polytree, and it is often used in subdivision methods for finding surface intersections ([88]).

Complexity of the Algorithm In general, complexity analysis of the subdivision algorithm is very complicated and strongly depends on the geometry of intersecting surfaces, which is well illustrated in Figure 4.10. Employing the subdivision algo-

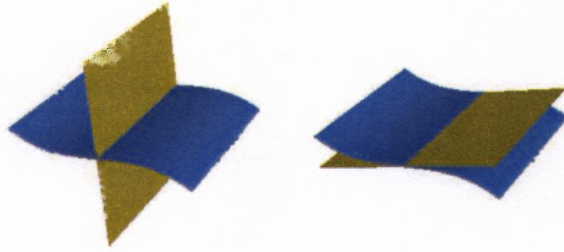


Figure 4.10 ‘Good’ and ‘bad’ intersections for the subdivision algorithm.

rithm, the intersection of the first pair of swept surfaces can be computed quite fast, because most bounding boxes would not intersect. The second pair, however, retains most of the bounding boxes at each decomposition step and, consequently, requires a considerable amount of time to find its intersection curve.

The dependency of the algorithm on the geometry of intersecting surfaces can be suppressed by making the following simplifying assumption: the number of bounding boxes that are eliminated at each decomposition step is a fraction of the total number of bounding boxes and is determined by a constant $1 - \alpha$, $0 < \alpha < 1$. Thus, if B is the number of bounding boxes for one of the surfaces at some decomposition step, only αB of them are not disjoint from the bounding boxes for the other surface.

In addition to α , we need to introduce a few more important parameters that are assumed to be the same for both intersecting surfaces. Let $h = 1/N$ be a discretization step of the algorithm, and assume that at each decomposition step we subdivide every swept patch into k^2 sub-patches. Then it is fairly easy to see that at the i -th decomposition step we have to subdivide $\alpha^{i-1} k^{2(i-1)}$ patches, each into k^2 parts. Notice also that each swept patch has exactly N/k^i linear segments in each side of its boundary after the i -th decomposition step. Finally, it is convenient to assume that $k = \sqrt[m]{N}$, where m is a nonnegative integer.

The structure of the algorithm suggests that its complexity can be calculated as a sum of three components: complexity of decomposition, complexity of culling bounding boxes, and complexity of finding and merging intersection segments. We start with the first component. Subdividing a patch with N/k^i linear segments in each side of its boundary into k^2 parts requires $O(2(k-1)N/k^i)$ arithmetic operations, since $2(k-1)$ trajectories consisting of N/k^i line segments have to be computed. The number of decomposition steps is trivially bounded by $m-1$. Consequently, the total number of arithmetic operations needed to perform the decomposition is proportional to the following sum:

$$\begin{aligned} \sum_{i=0}^{m-1} 2(k-1)\alpha^i k^{2i} \frac{N}{k^i} &= 2(k-1)N \sum_{i=0}^{m-1} \alpha^i k^i = 2(k-1)N \frac{\alpha^m k^m - 1}{\alpha k - 1} = \\ &= O(N\alpha^{m-1}k^m) = O(\alpha^{m-1}N^2). \end{aligned}$$

Therefore, the decomposition procedure takes $O(\alpha^{m-1}N^2)$ time.

Analysis of the elimination of bounding boxes depends on what method is employed for detecting box intersections. The simple algorithm described earlier finds intersections between two collections of n boxes in $O(n^2)$ time. Recalling that the number of bounding boxes for each surfaces is $\alpha^i k^{2i}$ after the i -th decomposition step and summing up, we obtain

$$\sum_{i=1}^m (\alpha^{i-1} k^{2i})^2 = O\left(k^4 \frac{\alpha^{2m} k^{4m} - 1}{\alpha^2 k^4 - 1}\right) = O(\alpha^{2(m-1)} k^{4m}) = O(\alpha^{2(m-1)} N^4),$$

proving that culling of bounding boxes can be done in $O(\alpha^{2(m-1)} N^4)$ time. Alternatively, we may choose the fast box intersection algorithm mentioned earlier. In this case, intersections between two collections of n boxes can be computed in $O(n \log^2 n)$

time. Summing over all decomposition steps yields

$$\begin{aligned} \sum_{i=1}^m \alpha^{i-1} k^{2i} \log^2(\alpha^{i-1} k^{2i}) &= O\left(k^2 \log^2(k^2) \sum_{i=1}^m i^2 \alpha^{i-1} k^{2(i-1)}\right) = \\ &O\left(k^2 \log^2(k^2) \frac{\alpha^m k^{2m} (m^2 \alpha^2 k^4 - (2m^2 + 2m - 1)\alpha k^2 + (m+1)^2) - \alpha k^2 - 1}{(\alpha k^2 - 1)^3}\right) = \\ &O\left(k^2 \log^2(k^2) \frac{(m+1)^2 \alpha^m k^{2m}}{\alpha k^2 - 1}\right) = O(\alpha^{m-1} N^2 \log^2 N). \end{aligned}$$

Hence, using the fast box intersection algorithm reduces the time complexity to $O(\alpha^{m-1} N^2 \log^2 N)$. This is a considerable improvement, especially when N is very big and m is fairly small.

Notice that both algorithms for finding box intersections require that collections of boxes be represented as simple arrays. Such a requirement is satisfied by slightly modifying the aforementioned polytree data structure – nodes at each level of the tree should be stored in a simple array. This modified data structure is also well suited for computing intersections of patches and merging the resulting segments. Indeed, the number of swept patches after culling bounding boxes at the last decomposition step is bounded by $\alpha^{m-1} k^{2(m-1)}$, and a box intersection algorithm provides us with the lists of intersecting boxes. These lists correspond to the lists of possibly intersecting swept patches. Therefore, assuming that each intersection segment can be computed in constant time, all such segments can be found in time proportional to the number of swept patches, that is, $O(\alpha^{m-1} N^{2-\frac{2}{m}})$. It is important to understand that the proportionality follows from the fact that each swept patch intersecting a given swept patch can be accessed in a constant time, as provided by simple arrays. Since intersecting segments are generally stored in the leaves of the same polytree, merging can be done by analyzing siblings of each leaf and then storing the result at the next level nodes of the tree. A simple method would just check each sibling against the other to find those that have sufficiently close endpoints. In general, this approach takes $O(n^2)$ time, where n is the number of siblings, but it can be improved

to $O(n)$ by employing the adjacency information for swept patches. According to our assumption, the number of siblings of any node in the tree is $O(\alpha k^2)$. It is also reasonable to assume that the number of intersection segments decreases by the factor of $\lambda \alpha^{-1} k^{-2}$ after merging at a given level is done. The constant λ is usually greater than 1 and such that λ^{m-1} is equal to the number of connected components of the intersection curve. The total number of leaves in the tree, that is, nodes at the level $m - 1$, is $O(\alpha^{m-1} k^{2(m-1)})$. Consequently, merging at this level takes $O(\alpha^{m-1} k^{2(m-1)})$ time, producing $O(\lambda \alpha^{m-2} k^{2(m-2)})$ nodes at the next level, each with $O(\lambda \alpha k^2)$ siblings. Applying this procedure recursively, we find that the total merging time is

$$O \left(\sum_{i=0}^{m-1} \lambda^i \alpha^{m-1-i} k^{2(m-1-i)} \lambda^i \alpha k^2 \right).$$

Similar to the previous cases, the sum can be bounded by a simpler expression, yielding that the complexity of the finding and merging intersection segments is $O(\alpha^{m-1} N^{2-\frac{2}{m}})$. Adding up all three major steps, we obtain the total time complexity of the algorithm of either $O(\alpha^{m-1} N^2 \log^2 N)$ or $O(\alpha^{2(m-1)} N^4)$, depending on whether the fast box intersection algorithms is used, or the simple one.

To complete our analysis, we should also compute the memory requirements of the subdivision algorithm. Fortunately, it is fairly easy to do – the memory space used by the algorithm is proportional to the number of swept patches at all levels, and can be expressed as the following sum:

$$\sum_{i=0}^m \alpha^i k^{2i} = O \left(\frac{\alpha^{m+1} k^{2(m+1)}}{\alpha k^2 - 1} \right) = O(\alpha^m k^{2m}) = O(\alpha^m N^2).$$

Hence, the space complexity of the subdivision algorithm is $O(\alpha^m N^2)$.

Pros and Cons The first thing worth mentioning is a considerable improvement in complexity compared to the simple discretization algorithm, even if the simple method is used for culling bounding boxes. Notice that the parameter α affects the

efficiency of the algorithm quite substantially. In fact, if α is very small, which is usually the case when a ‘good’ surface intersection occurs, α^{m-1} can be of order N^{-1} , reducing the complexity of the algorithm to $O(N \log^2 N)$ or $O(N^2)$, depending on the choice of a box intersection algorithm.

There are several difficulties, though. First, there is no general procedure for choosing an initial discretization step. So far, the choice is based on estimates of the curvature of the intersecting surfaces and is usually made using some heuristic methods. Second, there is no rule governing the number of decomposition levels. Is it necessary to subdivide until every swept patch is formed by single linear segments of the corresponding trajectories, or is it better to stop earlier and save some time? Finally, the algorithm may experience problems around points of tangential intersection. Therefore, additional heuristic procedures are needed to remedy the situation.

4.2.4 Marching Algorithm

Evidently, this algorithm is based on the ideas from existing marching methods for finding surface intersections, and to keep in conformity with the standards, we first describe steps needed to compute an intersection component when one of its points is given. A procedure for finding such points in all components of the intersection set is discussed later.

As has been mentioned earlier, an intersection curve can usually be traced by setting up some differential or algebro-differential equation. If the intersecting surfaces are smooth (or at least C^1) such a differential equation is obtained by computing a tangent vector to the intersection curve. We have derived the corresponding formulae for the case of parametric surfaces, but is it possible to do the same for swept surfaces? As we shall see, the answer to this question is positive, although the procedure is computationally efficient only when each intersecting surface is represented with both dual generators.

Let S_1 and S_2 be two swept surfaces with generators (\mathbf{x}_0, X_1, Y_1) and (\mathbf{x}_0, X_2, Y_2) , respectively, and let $(\mathbf{x}_0, Y_1, X_1^*)$ and $(\mathbf{x}_0, Y_2, X_2^*)$ be the corresponding dual generators. Suppose $\mathbf{x} \in S_1$ and $\mathbf{y} \in S_2$, then the unit normal vectors, $n_1(\mathbf{x})$ and $n_2(\mathbf{y})$, to S_1 and S_2 , respectively, are defined by

$$n_1(\mathbf{x}) = \frac{Y_1(\mathbf{x}) \times X_1^*(\mathbf{x})}{\|Y_1(\mathbf{x}) \times X_1^*(\mathbf{x})\|} \text{ and } n_2(\mathbf{y}) = \frac{Y_2(\mathbf{y}) \times X_2^*(\mathbf{y})}{\|Y_2(\mathbf{y}) \times X_2^*(\mathbf{y})\|}.$$

Consequently, if $\mathbf{x} \in S_1 \cap S_2$ is a point of transverse intersection, the tangent unit vector to the intersection curve at \mathbf{x} is given by

$$\tau(\mathbf{x}) = \frac{n_1(\mathbf{x}) \times n_2(\mathbf{x})}{\|n_1(\mathbf{x}) \times n_2(\mathbf{x})\|},$$

yielding the following result: If the surfaces S_1 and S_2 intersect transversally, and $\bar{\mathbf{x}}$ is a given point on the intersection curve, then the corresponding connected component is a part of the solution curve of the following initial value problem:

$$\frac{d\mathbf{x}}{dt} = \tau(\mathbf{x}), \quad \mathbf{x}(0) = \bar{\mathbf{x}},$$

where τ is defined as above. It is now clear that having explicit expressions for dual vector fields is important for efficiency of the algorithm, inasmuch as solving the foregoing differential equation requires values of X_1^* and X_2^* at every point of the corresponding discrete trajectory, and computing these values by solving another differential equation, as given in Lemma 3.2.20, is computationally expensive.

Let us reemphasize that the solution curve of the above initial value problem contains the intersection component as one of its parts. Therefore, it is necessary to define rules that would allow us to stop computations at the right moment. First, let us figure out a stopping criterion when the intersection component is a closed curve. In this case, it is reasonable to finish computations when the current point of the trajectory is in a sufficiently small neighborhood of the initial point, $\bar{\mathbf{x}}$. The size of the neighborhood should take into account the discretization step and the error intro-

duced by the numerical scheme. It is also useful to compute a few more points of the trajectory and determine if they fall within the allowed distance from the corresponding starting points. Alternatively, we can repeat computations in both directions and check if the end points of the trajectories get sufficiently close. Incidentally, the later procedure also improves the accuracy.

If the intersection component is not closed, its end points will necessarily lie on the boundary of one of the intersection surfaces. It is possible to implement a method that would check if a point on the trajectory is a boundary point by computing the corresponding distance. Unfortunately, each instance of such testing would generally take $O(N \log N)$ time, where $h = 1/N$ is a discretization step used to compute the boundaries of the intersecting surfaces. This time can be improved to $O(k)$, where k is the number of convex components of the boundaries of the surfaces, but such convex decomposition is usually not available. The problem can be resolved by detecting all boundary intersection points while looking for a starting point in each intersection component. The computational cost of this additional procedure is negligible, and the gain is substantial. Notice, though, that the starting point search can produce several points in the same intersection component, potentially leading to multiple tracing of the same curve. This issue can be resolved by evaluating distances between trajectories being computed and all the starting points. If some point of a trajectory falls within a sufficiently small neighborhood of a different starting point, the two corresponding trajectories should be merged.

We are now ready to describe the steps of the marching algorithm.

1. Find a starting point in each intersection component and all boundary intersection points.
2. Choose a discretization step, h , and evaluate the error bound, ϵ_n , for the n -th step of the chosen numerical scheme.

3. Initialize an auxiliary list of trajectories, $Tr = \emptyset$, and begin tracing by solving the following initial value problems

$$\frac{d\mathbf{x}}{dt} = \tau(\mathbf{x}), \quad \mathbf{x}(0) = \bar{\mathbf{x}}_k,$$

where $\bar{\mathbf{x}}_k$, $1 \leq k \leq K$, are the points found in the previous step. Tracing of each trajectory consists of the following steps:

- (a) Compute the next point, \mathbf{x}_n , in the trajectory.
- (b) Check if \mathbf{x}_n falls within the ϵ_n neighborhood of some starting point, $\bar{\mathbf{x}}_k$. If yes, go to the next step, if no, go to step 3a.
- (c) If $\bar{\mathbf{x}}_k$ is the starting point of the current trajectory, then the corresponding intersection component is probably a closed loop. Make additional testing and amendments, if necessary, store the trajectory in the auxiliary list, Tr , and go to step 3a.
- (d) If $\bar{\mathbf{x}}_k$ is not a boundary point, remove it from the list of starting points and go to step 3a.
- (e) If the starting point of the trajectory is also a boundary point, store the trajectory in Tr and go to step 3a.
- (f) Mark $\bar{\mathbf{x}}_k$ as the starting point of the current trajectory, reverse the time, t , and go to step 3a.

4. Report trajectories in Tr as an approximation of the intersection set.

The only step that has not been described yet is the first one, and we now proceed to explaining the corresponding procedure.

Starting Point Search Generally, finding starting points for a marching method is a complicated problem, which is usually solved by employing a subdivision method or loop detection technique. Having already developed a subdivision algorithm that can handle swept surfaces, it is reasonable to investigate how suitable it is for finding starting points. Alternatively, we can try to modify the basic ideas of available loop detection methods to obtain a method for evaluating starting points. In what follows, we elaborate on the first approach.

Application of the subdivision algorithm to a starting point search is rather straightforward. The key idea here is to look at collections of adjacent swept patches at each decomposition level. Notice that after some number of decomposition steps, each collection of adjacent swept patches will determine exactly one intersection component. Therefore, a point in each intersection component can be computed by taking one swept patch from each collection and finding one of its intersection points. Recalling our analysis of the subdivision algorithm, it becomes evident that such a procedure for computing starting points is quite efficient. A possible obstacle could be in finding collections of adjacent swept patches, but, luckily, the modified poly-tree data structure allows us to perform this operation rather effectively. It is worth mentioning, though, that determining the optimal number of decomposition steps is a rather intricate question. Indeed, small intersection loops, if present, may be lost if our decomposition is crude. On the other hand, if there are no small loops, producing a fine decomposition is just a waste of resources. Currently we cannot suggest any definite solution to this issue, although employment of local loop detection techniques seems promising. To complete the search of starting points we should also identify all boundary intersection points. This step can be easily incorporated in the above procedure by computing intersections of boundary swept patches in the adjacency collections.

Complexity of the Algorithm Having a natural division of the marching algorithm into two parts – starting point search and tracing – it is convenient to perform the complexity analysis of each part separately. Let us consider tracing first. It is obvious that if K is the number of connected components of the intersection curve and $O(T)$ is the time needed to trace one intersection component, then the total tracing time is $O(KT)$. Choosing a discretization step so that $h = 1/N$, we find that it takes $O(N)$ time to compute a trajectory, and since all the auxiliary steps used when trac-

ing an intersection component take a constant time, we obtain $O(T)=O(N)$. Hence, the total tracing time is $O(KN)$. Incidentally, this is also the space complexity of this part of the algorithm.

Analysis of the starting point search involves practically the same steps as used when investigating the subdivision algorithm. Therefore, we shall omit tedious calculations and present the final result: If $h = 1/N$ is the discretization step, then the complexity of computing starting points in every intersection component is $O(\alpha^p N^{\frac{2p}{m}} \log^2 N)$ in time and $O(\alpha^p N^{\frac{2p}{m}})$ in space, where p is the number of decomposition steps, and α and m are the corresponding parameters from the subdivision algorithm.

Pros and Cons The marching algorithm shows a noticeable improvement in efficiency when comparing with the subdivision algorithm without any complications in its structure. In fact, it may be even simpler to implement. Furthermore, tracing trajectories provides some useful local information about the intersecting surfaces. For instance, if the norm of the cross product of the normal vectors $n_1(\mathbf{x})$ and $n_2(\mathbf{x})$ becomes small, it is highly probable that the trajectory approaches a point of tangential intersection. While the current version of the marching algorithm is not designed for handling tangential intersections, using the above test we may try to employ some additional tools around singularities to improve the behavior of the algorithm.

Arguably the most problematical part of the marching method is the choice of discretization steps. Due to a potentially different geometric structure of the intersecting surfaces around different intersection components, or even points, adaptive discretization would probably be the best solution. The problem is that there is no general rule governing the size of the discretization step. As usual, we can resort to curvature estimates, but such ad hoc methods are not always reliable.

4.2.5 Intersections of Restricted Swept Surfaces

The designed intersection algorithms do not assume any restrictions on sweeps that generate the intersecting surfaces. Oftentimes, though, a context in which swept surfaces are used provides additional information about the nature of the underlying sweeps, thus enabling us to derive more specific mathematical representations. For example, it may be known that all the sweeps are rigid. In fact, rigid sweeps are quite common in practice. Recalling Definition 3.1.2, we may derive the following *rigid sweep differential equation*:

$$\dot{\mathbf{x}} = \dot{\xi}(t) + \dot{A}(t)A^T(t)(\mathbf{x} - \xi(t)),$$

where $\mathbf{x}, \xi(t) \in \mathbb{R}^n$, $A(t) \in SO(n)$, $\xi(0) = 0$, and $A(0) = I$ – the identity matrix in \mathbb{R}^n . The closed form solution of this equation is given by the following sweep:

$$\sigma_t(\mathbf{x}) = A(t)\mathbf{x} + \xi(t).$$

An important corollary of this result is that rigid swept manifolds have explicit parametric representations if the initial manifold has one. Indeed, suppose the initial manifold is parametrized by

$$F : \underbrace{[0, 1] \times \dots \times [0, 1]}_{n-2 \text{ times}} \rightarrow \mathbb{R}^n,$$

that is, $F(s_1, s_2, \dots, s_{n-2}) = (f_1(s_1, s_2, \dots, s_{n-2}), \dots, f_n(s_1, s_2, \dots, s_{n-2}))$. Then the corresponding swept manifold is parametrized by the map

$$S : \underbrace{[0, 1] \times \dots \times [0, 1]}_{n-1 \text{ times}} \rightarrow \mathbb{R}^n, \quad S(s_1, \dots, s_{n-2}, t) = A(t)F(s_1, \dots, s_{n-2}) + \xi(t).$$

Since the initial manifold for a swept surface is just a curve, its parametrization is usually known or easily derivable. Consequently, it is reasonable to assume that a

rigid swept surface has an explicit parametric representation of the form

$$f(s, t) = A(t)c(s) + \xi(t),$$

where $c(t)$ is a parametric representation of the initial curve, and $A(t)$ and $\xi(t)$ are defined as above. Thus, intersections of rigid swept surfaces can be computed using any of the existing intersection methods for parametric surfaces.

It is clear that having a closed form expression for swept surfaces is greatly beneficial, which leads to the following question: What other classes of swept manifolds (surfaces) can be parametrized explicitly? A partial answer to this question can be obtained by making some modifications to a rigid sweep. For example, we can introduce a simple deformation by means of a scaling factor. In this case, the sweep differential equation is

$$\dot{\mathbf{x}} = \dot{\xi}(t) + \left[\left(\frac{\dot{\alpha}(t)}{\alpha} \right) A(t) + \dot{A}(t) \right] A^T(t) (\mathbf{x} - \xi(t)),$$

where $\alpha : [0, 1] \rightarrow \mathbb{R}$ is a smooth map such that $\alpha(0) = 1$ and $\alpha(t) > 0$ for all $t \in [0, 1]$; it represents the scaling factor. The corresponding sweep is given by

$$\sigma_t(\mathbf{x}) = \alpha(t)A(t)\mathbf{x} + \xi(t).$$

It is easy to see that the parametric representation of a swept surface generated with such a sweep is defined by

$$f(s, t) = \alpha(t)A(t)c(s) + \xi(t).$$

It is important to realize, however, that using existing intersection methods for swept surfaces with explicit parametrization may not necessarily be the best option. In some cases, our algorithms, which were specifically designed for swept surfaces, can be more effective. Moreover, it may also be possible to improve the performance

of these algorithms by taking into account the simplified structure of swept surfaces. All these issues constitute a significant part of our future research.

4.3 Homological Intersection Detection

There is a variety of approaches to analyzing topological spaces. The ones based on algebraic topology focuses on projecting the problem into the realm of algebra, mapping topological features to the corresponding features of some algebraic structures, e.g. groups or rings, thus enabling one to employ the powerful algebraic machinery. The diversity of algebraic topology methods is quite astonishing in itself, but homology theory often provides the best compromise between simplicity and generality. It is particularly well suited for analyzing SW complexes [42, 43], which represent a broad class of topological spaces and cover most practical cases. In particular, the intersection of two smooth submanifolds of \mathbb{R}^n is locally a SW complex.

In this sections, we present what is perhaps a slightly unconventional approach to the intersection problem, which is based on homology theory. Later, we shall discuss how the theoretical results may be employed in practice using available data structures. Before plunging into details of the general case, let us take a look at an example. Consider two surfaces in space that intersect transversally (Figure 4.11). Let us denote them by M and N . It is easy to see that if $x \in M \cap N$, then a neighborhood of x is homeomorphic to a union of two intersecting planes. If $x \notin M \cap N$, a neighborhood of x is homeomorphic to just one such plane (Figure 4.11). Unfortunately, checking if two spaces are homeomorphic is not an easy problem; in many cases, it can be reduced to comparing Betti numbers or homology groups of the spaces. It turns out that to discern intersection points in the above example it is enough to compute the local homology groups of $M \cup N$ at a point of interest. We can do it fairly easily, obtaining the following: If $x \in M \cup N$, then the only non-trivial homology group is $H_2(M \cup N, M \cup N \setminus \{x\}) = \mathbb{Z}^3$. If $x \notin M \cup N$, then the non-trivial

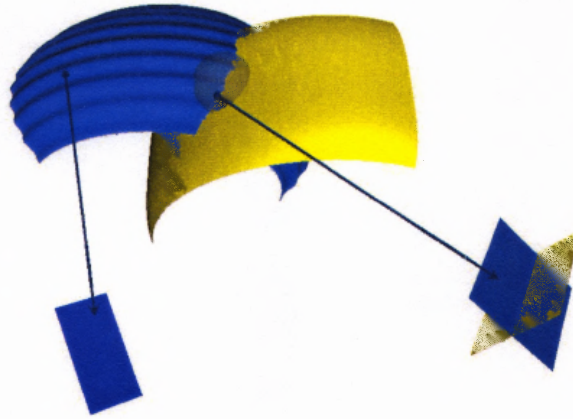


Figure 4.11 Transverse intersection two surfaces.

homology group is $H_2(M \cup N, M \cup N \setminus \{x\}) = \mathbb{Z}$. Similar homological characterization holds for transverse intersections of smooth, codimension-1 submanifolds of \mathbb{R}^n for any n ([17]). Things become more complicated though, when non-transverse intersections occur, especially in higher-dimensional spaces. Still, as we shall see, a description in terms of homology remains quite elegant, straightforward and useful.

We now proceed to analyzing the general case. Usually, objects under consideration are assumed to be smooth, compact submanifolds of \mathbb{R}^n without a boundary. But since homology is homotopy invariant, we start by considering *topological* submanifolds of \mathbb{R}^n . To simplify our analysis, we impose some restrictions on the intersection set – we assume that it is an s -subvariety of \mathbb{R}^n (as defined in [17]), which, for example, is always the case if the intersecting manifolds are analytic, or piecewise linear.

Theorem 4.3.1 *Let M and N be two topological submanifolds of \mathbb{R}^n without boundaries, and let $I = M \cap N$ be an s -subvariety. Denote by p, q and r dimensions of M, N and I , respectively, and let $n > p \geq q > 0$.*

(1) If $x \in (M \cup N) \setminus I$ then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p, x \in M, \text{ or } k = q, x \in N \\ 0, & \text{otherwise} \end{cases}$$

(2) If $x \in I$ the following hold:

(i) if $p > q > r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p, q \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

(ii) if $p > q = r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p \\ \mathbb{Z} \oplus H_{k-1}(I, I \setminus \{x\}), & \text{if } k = q = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

(iii) if $p = q > r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}^2, & \text{if } k = p = q \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

(iv) if $p = q = r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}^2 \oplus H_{k-1}(I, I \setminus \{x\}), & \text{if } k = p = q = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

Proof. Let $x \in M \setminus I$, and consider $H_k(M \cup N, M \cup N \setminus \{x\})$. By excision, $H_k(M \cup N, M \cup N \setminus \{x\}) = H_k(U, U \setminus \{x\})$, where $U \subset M$ is a neighborhood of x in M , and from the long exact sequence for the pair $(U, U \setminus \{x\})$ we infer that $H_k(U, U \setminus \{x\}) = \tilde{H}_{k-1}(U \setminus \{x\})$. Since M is a manifold of dimension p , the set $U \setminus \{x\}$ is homotopic to a $(p - 1)$ -sphere, S^{p-1} . Therefore, $\tilde{H}_{p-1}(U \setminus \{x\}) = \mathbb{Z}$ and $H_{k-1}(U \setminus \{x\}) = 0, k \neq p$. The case $x \in N \setminus I$ is proved in the same way.

Now let $x \in I$. Again, by excision we get $H_k(M \cup N, M \cup N \setminus \{x\}) = H_k(U, U \setminus \{x\})$, where $U \subset M \cup N$ is a neighborhood of x in $M \cup N$, and the long sequence for the pair $(U, U \setminus \{x\})$ implies that $H_k(U, U \setminus \{x\}) = \tilde{H}_{k-1}(U \setminus \{x\})$. Let us

denote $X = U \setminus \{x\}$. We can write $X = A \cup B$, where $A = U \cap M \setminus \{x\}$ and $B = U \cap N \setminus \{x\}$. Notice that from excision and the corresponding long exact sequence, we get $H_k(I, I \setminus \{x\}) = \tilde{H}_{k-1}(A \cap B)$. Consider the reduced Mayer-Vietoris sequence for A, B and X :

$$\begin{array}{ccccccc} \cdots & \longrightarrow & \tilde{H}_k(A \cap B) & \xrightarrow{\Phi} & \tilde{H}_k(A) \oplus \tilde{H}_k(B) & \xrightarrow{\Psi} & \tilde{H}_k(X) \xrightarrow{\partial} \\ & & & & & & \\ & & \xrightarrow{\partial} & \tilde{H}_{k-1}(A \cap B) & \longrightarrow & \cdots & \longrightarrow \tilde{H}_0(X) \longrightarrow 0 \end{array}$$

Since M and N are manifolds of dimensions p and q , respectively, A and B are homotopic to S^{p-1} and S^{q-1} , respectively. So, the only non-trivial reduced homology groups of A and B are \tilde{H}_{p-1} and \tilde{H}_{q-1} , respectively. Both of them are \mathbb{Z} . By assumption, $M \cap N$ is an s -subvariety of \mathbb{R}^n of dimension r , which implies that U can be chosen such that $M \cap N \cap U$ is homeomorphic to a finite union of r -dimensional balls B_j^r each of which contains x . Therefore, $M \cap N \cap U \setminus \{x\} = A \cap B$ is homotopic to a connected union of $(r-1)$ -dimensional spheres. Hence, the only non-trivial reduced homology group of $A \cap B$ is \tilde{H}_{r-1} .

Getting back to the Mayer-Vietoris sequence, we see that it gives rise to one, two or three short exact sequences, as follows.

Case 1: $p > q > r + 1$. We get three short exact sequences:

$$0 \longrightarrow \mathbb{Z} \longrightarrow \tilde{H}_{p-1}(X) \longrightarrow 0,$$

$$0 \longrightarrow \mathbb{Z} \longrightarrow \tilde{H}_{q-1}(X) \longrightarrow 0,$$

$$0 \longrightarrow \tilde{H}_r(X) \longrightarrow \tilde{H}_{r-1}(A \cap B) \longrightarrow 0,$$

from which the result follows.

Case 2: $p > q = r + 1$. We obtain two short exact sequences:

$$0 \longrightarrow \mathbb{Z} \longrightarrow \tilde{H}_{p-1}(X) \longrightarrow 0,$$

$$0 \longrightarrow \mathbb{Z} \longrightarrow \tilde{H}_r(X) \longrightarrow \tilde{H}_{r-1}(A \cap B) \longrightarrow 0.$$

The first sequence implies that $H_{p-1}(X) \approx \mathbb{Z}$. In the second case, using the fact that all the groups involved are free and finitely generated, we infer that the sequence is a split exact sequence. Hence, $H_r(X) \approx \mathbb{Z} \oplus H_{r-1}(A \cap B)$.

Case 3: $p = q > r + 1$. We have two short exact sequences:

$$0 \longrightarrow \mathbb{Z}^2 \longrightarrow \tilde{H}_{p-1}(X) \longrightarrow 0,$$

$$0 \longrightarrow \tilde{H}_r(X) \longrightarrow \tilde{H}_{r-1}(A \cap B) \longrightarrow 0,$$

which yield the desired result.

Case 4: $p = q = r + 1$. We get one short exact sequence:

$$0 \longrightarrow \mathbb{Z}^2 \longrightarrow \tilde{H}_r(X) \longrightarrow \tilde{H}_{r-1}(A \cap B) \longrightarrow 0.$$

Again, using the fact that all the groups involved are free and finitely generated, we conclude that $H_r(X) \approx \mathbb{Z}^2 \oplus H_{r-1}(A \cap B)$.

Notice that all non-trivial homology groups for A , B , or $A \cap B$ are considered in these four cases. Therefore, the short exact sequences for $\tilde{H}_k(X)$ that are not considered above are of the form

$$0 \longrightarrow \tilde{H}_k(X) \longrightarrow 0,$$

which implies that all such $\tilde{H}_k(X)$ are trivial. This completes the proof. \square

Remark 4.3.2 *It can be seen from the proof that the hypothesis that the intersection set is an s -subvariety of \mathbb{R}^n can probably be weakened, since we only need the sequence*

$$0 \longrightarrow \mathbb{Z} \longrightarrow \tilde{H}_r(X) \longrightarrow \tilde{H}_{r-1}(A \cap B) \longrightarrow 0,$$

to be a split exact sequence.

The above theorem has several important corollaries, the first of which generalizes Theorem 7.2 of [17].

Corollary 4.3.3 *Let M and N be C^r , $r \geq 1$, compact submanifolds of \mathbb{R}^n without boundaries of dimensions $p < n$ and $q < n$, respectively, and suppose $M \pitchfork N$. Then if $x \in M \cap N$, the local (relative) homology satisfies*

$$H_k(M \cup N, (M \cup N) \setminus \{x\}) = \begin{cases} \mathbb{Z}^3, & \text{if } k = p, p = q = n - 1 \\ \mathbb{Z}^2, & \text{if } k = p, p = q < n - 1 \\ \mathbb{Z}, & \text{if } k = p, q, p + q - n; p \neq q \\ 0, & \text{otherwise} \end{cases},$$

and if $x \in (M \cup N) \setminus (M \cap N)$,

$$H_k(M \cup N, (M \cup N) \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p \text{ and } x \in M \text{ or } k = q \text{ and } x \in N \\ 0, & \text{otherwise} \end{cases}$$

Proof. It follows from Theorem 4.3.1 and the fact that the transverse intersection of two C^r submanifolds of \mathbb{R}^n of dimensions p and q is a C^r submanifold of dimension $p + q - n$. \square

Corollary 4.3.4 *If a C^1 compact, connected, codimension-1 submanifold M of \mathbb{R}^n has points of transverse self-intersection, then there exists an $x \in M$ such that $H_{n-1}(M, M \setminus \{x\}) = \mathbb{Z}^3$.*

If C^r submanifolds M and N intersect tangentially, then $M \cap N$ can have a quite complicated structure, e.g. it may not even be a submanifold. The following easily proved lemma describes, to some extent, the local structure of a tangential intersection of two C^r submanifolds of \mathbb{R}^n .

Lemma 4.3.5 *Let M and N be two C^r submanifolds of \mathbb{R}^n without boundaries of dimensions p and q , respectively, and let $x \in M \cap N$ be a point where M and N intersect non-transversally. Then there is a neighborhood $U \subset \mathbb{R}^n$ of x and a C^r function $f : U \cap M \rightarrow \mathbb{R}^{n-q}$ such that $U \cap M \cap N = f^{-1}(0)$ and x is a critical point of f .*

A reader familiar with differential topology will notice that the statement of the lemma is equivalent to saying that $M \cap N$ is a C^r subvariety of \mathbb{R}^n . It may not be an s -subvariety though. Therefore, we still need the corresponding assumption for the statement of Theorem 4.3.1 to be true.

In many cases, the local homology groups of $M \cap N$ can be computed fairly easily, thereby yielding explicit formulas for the local homology at the intersection point. We demonstrate this in the following example. Consider the paraboloid M , given by $z = x^2 + y^2$, and the surface N , given in cylindrical coordinates by the following equations:

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \\ z &= r^2 \sin(8\theta) \end{aligned} \quad ,$$

where $\theta \in [0, 2\pi]$, $r \geq 0$. A neighborhood of the origin is shown in Figure 4.12. These two surfaces intersect tangentially, and the intersection set, I , is an s -subvariety shown in Figure 4.13. By excision, the local homology groups $H_k(I, I \setminus \{0\})$ are isomorphic to the corresponding reduced homology groups $\tilde{H}_{k-1}(I \setminus \{0\})$. Since the set $I \setminus \{0\}$ consists of eight contractible components, we obtain $H_k(I, I \setminus \{0\}) = \mathbb{Z}^7$

for $k = 1$ and $H_k(I, I \setminus \{0\}) = 0$ for $k \neq 1$. Thus, Theorem 4.3.1 implies that $H_k(M \cup N, M \cup N \setminus \{0\}) = \mathbb{Z}^9$ for $k = 2$ and $H_k(M \cup N, M \cup N \setminus \{0\}) = 0$ for $k \neq 2$.

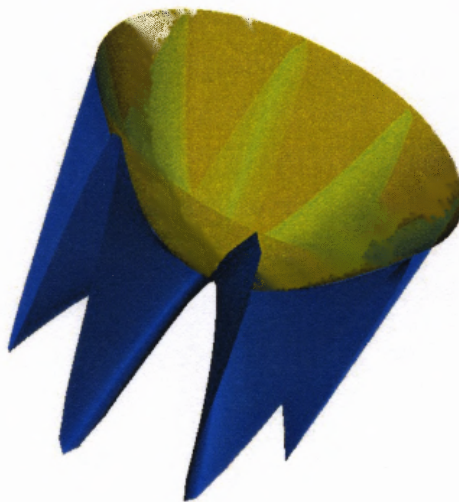


Figure 4.12 Tangentially intersecting surfaces.

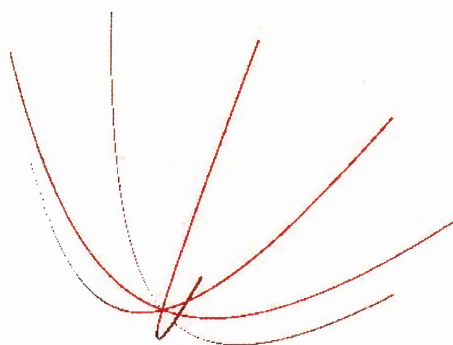


Figure 4.13 Intersection set of the surfaces

4.3.1 Manifolds with Boundary

Theorem 4.3.1 can be easily generalized to the case of topological submanifolds with boundaries. The proof remains virtually unchanged: we just have to consider cases when a point of interest belongs to the boundary of each of the submanifolds. If $x \in \partial M \cap N$ or $x \in M \cap \partial N$, then the set A or, respectively, B used in the proof is

contractible. Therefore, the corresponding elements in the Mayer-Vietoris sequence are zeros. Following the proof, we can conclude that if $x \in \partial M \cap (N \setminus \partial N)$ or $x \in (M \setminus \partial M) \cap \partial N$ then the local homology will be as in the items (i) – (iv) of Theorem 4.3.1 with \mathbb{Z} factored out at the corresponding places. If $x \in \partial M \cap \partial N$, then we should factor out \mathbb{Z}^2 . Thus, we obtain the following theorem.

Theorem 4.3.6 *Let M and N be two topological submanifolds of \mathbb{R}^n with boundaries, and let $I = M \cap N$ be an s -subvariety (with a boundary). Denote by p, q and r dimensions of M, N and I , respectively, and let $n > p \geq q > 0$. If $x \in I \cap (\partial M \cup \partial N)$ the following hold:*

(i) if $p > q > r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p, x \notin \partial M \\ & \text{or } k = q, x \notin \partial N \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

(ii) if $p > q = r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p, x \notin \partial M \\ \mathbb{Z} \oplus H_{k-1}(I, I \setminus \{x\}), & \text{if } k = q = r + 1, x \notin \partial N \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = q = r + 1, x \in \partial N \\ 0, & \text{otherwise} \end{cases}$$

(iii) if $p = q > r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p = q, x \notin \partial M \cap \partial N \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = r + 1 \\ 0, & \text{otherwise} \end{cases}$$

(iv) if $p = q = r + 1$, then

$$H_k(M \cup N, M \cup N \setminus \{x\}) = \begin{cases} \mathbb{Z} \oplus H_{k-1}(I, I \setminus \{x\}), & \text{if } k = p = q = r + 1, \\ & x \notin \partial M \cap \partial N \\ H_{k-1}(I, I \setminus \{x\}), & \text{if } k = p = q = r + 1, \\ & x \in \partial M \cap \partial N \\ 0, & \text{otherwise} \end{cases}$$

The corresponding generalization of Corollary 4.3.3 is as follows.

Corollary 4.3.7 *Let M and N be C^r , $r \geq 1$, compact submanifolds of \mathbb{R}^n with boundaries of dimensions $p < n$ and $q < n$, respectively, and suppose $M \pitchfork N$. Then if $x \in \partial M \cap (N \setminus \partial N)$ or $x \in (M \setminus \partial M) \cap \partial N$, the local (relative) homology satisfies*

$$H_k(M \cup N, (M \cup N) \setminus \{x\}) = \begin{cases} \mathbb{Z}^2, & \text{if } k = p, p = q = n - 1 \\ \mathbb{Z}, & \text{if } k = p, p = q < n - 1 \\ \mathbb{Z}, & \text{if } k = p + q - n; p \neq q \\ \mathbb{Z}, & \text{if } k = p, x \in (M \setminus \partial M) \cap \partial N; p \neq q \\ \mathbb{Z}, & \text{if } k = q, x \in \partial M \cap (N \setminus \partial N); p \neq q \\ 0, & \text{otherwise} \end{cases},$$

and if $x \in \partial M \cap \partial N$, then

$$H_k(M \cup N, (M \cup N) \setminus \{x\}) = \begin{cases} \mathbb{Z}, & \text{if } k = p, p = q = n - 1 \\ \mathbb{Z}, & \text{if } k = p + q - n; p \neq q \\ 0, & \text{otherwise} \end{cases}$$

These results show that the local homology allows us to distinguish not only intersection points from non-intersection points, but also boundary points from non-boundary points in the intersection set of two transverse, codimension-1, submanifolds of \mathbb{R}^n .

CHAPTER 5

APPLICATIONS

Because of the ubiquity of computer aided geometric modeling systems and the fact that all of them have to deal with intersections of surfaces, employment of intersection algorithms became so mundane that many current descriptions of applications of intersection methods resort to merely listing several areas of science that utilize geometric models. While such an approach is very laconic and may be quite informative, it might be interesting to notice that most descriptions of this kind pertain to intersection methods designed for highly popular classes of surfaces, for example, NURBS. Applicability of intersection algorithms for less popular types of surfaces has usually to be justified. Interestingly, it is commonly done by showing that the corresponding surface class is well suited for representing some geometric models, which, in its turn, implies that the algorithm is well fit for computing intersections that might occur in those models.

It would be nice to say that swept manifolds belong to the elite family of popular geometric representations. Unfortunately, a lot of things have still to be done to secure a place for swept surfaces and manifolds in the world of computer aided geometric design. Consequently, we shall assume the above mentioned approach and describe possible applications of our results by showing how swept surfaces can be employed in various geometric models and pinpointing occurrences of swept surface intersections. At the end, we also present possible applications of the homological intersection characterization.

5.1 Virtual Sculpting

When Michelangelo was carving his David, he probably rejected quite a few prototypes before achieving all his goals in the final version. Obviously, those discarded

models consumed a lot of his efforts and resources, and not every artist or designer can afford such a waste of time. Fortunately, in our technological time there is an easy solution to this problem – virtual sculpting. Its basic idea is quite naive: perform carving or similar operations on a computer model of an object and then use this model and some appropriate machinery to create an actual prototype. With such a setup, all actions of a designer take place in a virtual world and can be easily undone or redone. Therefore, there will be no waste of resources, and almost all efforts can be invested into the artistic part of the process.

To get a better idea of how virtual sculpting is done, consider Figure 5.1. You can see a person (perhaps a designer) furnished with a helmet and a pen, both

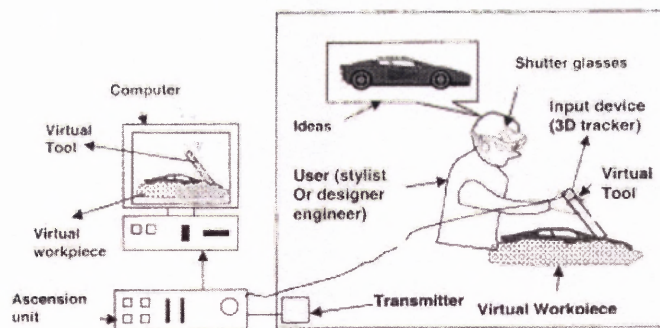


Figure 5.1 Virtual reality sculpting system.

connected to a computer. The pen represents a carving tool, which can be ‘virtually’ anything, and the helmet provides a visual feedback. When the designer chooses a tool and moves the pen, the software calculates the trajectory of the tool and computes its effect on the workpiece, thereby ‘carving out’ the corresponding part. Notice that the key step of the above procedure is computing the intersection of the trajectory of the tool with the workpiece. This step can be naturally modeled as the swept volume intersection problem. Furthermore, many tools effect a workpiece only with their sharp edges, as illustrated in Figure 5.2. Such an edge is generally modeled as a curve, and its swept volume is a swept surface. A workpiece is usually a solid, so

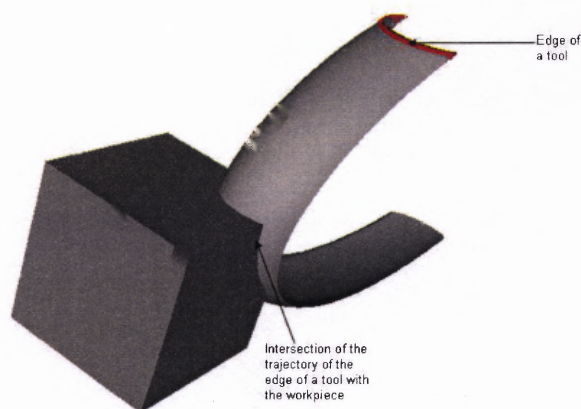


Figure 5.2 Intersection of a tool with a workpiece as the swept surface intersection problem.

it is completely determined by its boundary. Initially, this boundary is just a simple parametric or piecewise swept surface, and it is easy to see that carving preserves its piecewise swept structure. Therefore, one of the main parts of virtual sculpting can be effectively modeled as the swept surface intersection problem, thus creating a nice application niche for previously developed algorithms.

5.2 Tissue Engineering

The severe nature of many injuries sustained by humans often necessitates replacement of the damaged tissue by an artificial material. For example, each year around 200,000 people in the U.S. undergo a total hip replacement. The corresponding procedure is well developed, but there are still some shortcomings which may lead to serious complications. A big improvement could be achieved by designing a material that would imitate properties of the original tissue as closely as possible. Unfortunately, current possibilities are still quite limited, because the number of parameters that determine the structure of biological materials is too large to handle and, furthermore, not all such parameters are always known.

In the recent work by Regli and Sun and their collaborator, they have developed several innovative methods for representing properties of heterogeneous biomaterials ([86, 11]). The basic parameters taken into account were porosity, density and permeability. In addition, the seemingly random nature of biological materials was simulated by employing elements of stochastic geometry. While their algorithms perform sufficiently well and even have been used in manufacturing processes to create heterogeneous objects, such as bone replacement tissue, there is still room for improvement. It seems that capturing some of the physical processes that govern formation of the biological material would yield good results, and it turns out that swept surfaces could be a very useful tool in this case.

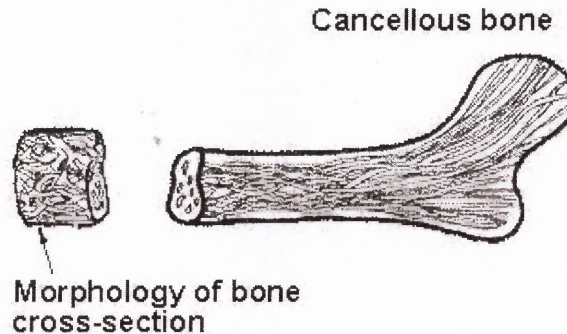


Figure 5.3 Simplified picture of internal bone morphology.

Consider a bone illustrated in Figure 5.3. A close inspection of its internal structure reveals that it can be regarded as being comprised of some biological material threads, perhaps channels, swept through the length of the bone. A cross section of each such thread can be viewed as a circle, suggesting that tissue strands in a bone might be modeled as swept surfaces of a family of circles. This approach was first proposed in [17], where the following simple model was presented.

Let a bone be aligned along the z -axis and suppose that at $z = 0$ the cross section of the bone consists of n circles, $\partial D_1, \dots, \partial D_n$, that bound disks D_1, \dots, D_n . We want to create channels inside the bone by flowing these disks along its length.

While we may allow the disks to intertwine, we shall assume that their density at each $z = \text{constant}$ is proportional to the initial density at $z = 0$. All these requirements can be easily expressed in mathematical terms. A sweep of the disks is defined by the following ODE:

$$\dot{\mathbf{x}} = X(\mathbf{x}),$$

where

$$\mathbf{x} = (x, y, z), \quad X(\mathbf{x}) = (U(\mathbf{x}), 1).$$

Denoting $\mathbf{u} = (x, y)$, the desired density property is obtained by imposing the following restriction:

$$\partial_{\mathbf{u}}U(\mathbf{u}, z) = \kappa(z)A(\mathbf{u}, z),$$

where $A \in SO(2)$. Here the positive quantity $\kappa(z)$ is the scale factor, so the density of the disks bounded by the channels walls at a cross section $z = z^*$ is just $\kappa(z) \times$ (density at $z = 0$).

As we can see, the above restrictions are very mild, which makes the model capable of generating a wide variety of very intricate configurations that may represent rather realistic models of bone morphology. There is an important deviation from reality though – branching or merging of channels was not allowed. One possible solution to this issue is to introduce several families of disks with different swept vector fields. Then branching and merging can be simulated via intersections of channels from different families – at a merging point, the part of one of the channels following the intersection is removed, while at a branching point, the new part following the intersection is added. The intersection problem that arises in this case is clearly the swept surface intersection problem. Thus, the level of applicability of previously developed intersection algorithms rises up another notch.

5.3 Homological Intersection Characterization

Throughout the years, models in computer aided geometric design have evolved from simple surfaces to complicated, non-manifold, higher-dimensional spaces. The way these models are represented can be quite different, but here we will focus on models that are simplicial complexes [73, 42]. Such complexes can be very general and may easily model rather intricate topological spaces, which makes them highly attractive to those who deal with non-manifolds. Non-manifoldness itself can be of many different types, but the type that most frequently occurs in geometric models is one resulting from the intersection of several (usually two) manifolds. The intersections can be both transverse and tangential, and their number can be quite large. When one performs some operations on such non-manifold models (e.g. smoothing), topological artifacts may appear in a neighborhood of a non-manifold point. Also, some topological defects at (or around) intersections in a model are often produced during its construction. Detecting such flaws is quite problematical, since some of them may not be visible to a human eye. There are several procedures aimed at reducing such deficiencies in a geometric model, but most of them are restricted to two- or three-dimensional cases. Results obtained in the previous section allow us to design a general framework for analyzing and improving geometric non-manifold models in any finite-dimensional space. The basic idea is rather naive – simply compute the local homology groups at a point of interest. Such a point is usually a non-manifold point, and the local homology is usually known to a user, since a designer knows the number of manifolds and in what way they should intersect at each point of the model. Therefore, as follows from Theorem 4.3.1 and the corresponding corollaries, comparing the computed local homology with the desired one should determine whether or not there is a defect in the model at this point.

Besides detecting flaws, the foregoing results can often be used to determine whether a given model is a manifold. Indeed, the local homology at a manifold point

has only one non trivial homology group, which is isomorphic to \mathbb{Z} . In most models, the non-trivial homology groups at a non-manifold point are different from \mathbb{Z} . In fact, many models turn out to be codimension-1 s-subvarieties. To detect non-manifoldness in such a model, we can just browse through all vertices of the model and check if the local homology group of the corresponding dimension is \mathbb{Z} . The browsing may not be very efficient though, and we will elaborate on this later.

We now provide more details regarding actual implementation of the above ideas. Obviously, any algorithm that does such computations depends in a fundamental way on the data structure used to describe the model. There are several non-manifold data structures, e.g. Radial Edge Data Structure [94], Tri-Cyclic Cusps Data Structure [41], SGC Data Structure [87]. Though there are some significant differences among them, all these data structures have an important common feature - they provide means for local analysis of the topology of the model. For example, it is always possible to compute the link of a point quite efficiently. This allows us to describe the basic steps of the intersection detection algorithm without getting involved in tedious details. Suppose that a model is represented by a simplicial complex S (described by one of the data structures). Then the primary steps are the following:

1. Choose (using some method) a vertex $v \in S$ and compute its link $L(v)$.
2. Compute the homology groups of $L(v)$. By excision, these are the local homology groups of S at v .
3. Using Theorem 4.3.1, try to determine what kind of intersection, if any, occurs at the point v . If needed, go to step 1.

These steps are very general, and many intermediate steps that were skipped require much work. For example, computation of homology groups is a big question that has been addressed in many papers [49, 51, 50, 24]. Fortunately, the link of a vertex in a simplicial complex does not usually contain a large number of simplices.

Therefore, even the most basic algorithms, say the reduction algorithm, can be used without any real difficulties. Also, it is worth noting that Theorem 4.3.1 may not provide an answer in some exotic cases. But, as we have already mentioned, this does not usually happen in practice. Another important question is how to choose a point in Step 1. Though existing data structures provide excellent means for analyzing local topological structure of a simplicial complex, they do not incorporate any information that would facilitate global analysis. As a result, it is impossible, for example, to figure out a priori in what direction it is best to move if we want to determine whether there is a non-manifold point in the model. Therefore, an exhaustive search has to be performed in such cases, and this is highly inefficient.

We also should point out that computing local homology groups (as indicated in Step 2) may be an overkill in some cases. In fact, if one looks for a transverse intersection in a simplicial complex that represents a codimension-1, s -subvariety, the following lemma [17] is much easier to apply than the homological criteria developed above.

Lemma 5.3.1 *Let M and N be two codimension-1 submanifolds of \mathbb{R}^n without boundaries, and suppose \mathcal{T} is a triangulation of $M \cup N$. If $v \in \mathcal{T}$ is a vertex that represents a point of transverse intersection of M and N , then each $(n-2)$ -simplex of \mathcal{T} incident to v is the face of exactly four $(n-1)$ -simplices.*

Thus, identifying transverse intersections is much less computationally expensive than locating non-transverse intersections, since adjacency information is readily available in any non-manifold data structure. It is also possible to obtain similar results for tangential intersections; but such results should be derived on a case-by-case basis and would not provide the kind of uniform classification that is available through the homology approach.

CHAPTER 6

CONCLUSIONS AND FUTURE RESEARCH

This work is primarily devoted to a study of a new class of geometric objects called swept manifolds, with a great emphasis on the problem of finding intersections of swept surfaces. Although swept manifolds can be regarded as special case of swept volumes, whose popularity and importance in geometric modeling has been constantly growing, we have shown that distinguishing them in a separate class provides many advantages. The very definition of a swept manifolds guarantees that its topology is significantly simpler than the topology of a swept volume. Using this simplification, we have been able to conduct a fairly deep analysis of topological and geometric properties of swept manifolds. In the course of our investigation, we have introduced several subclasses of swept manifolds, in particular, regular and critical swept manifolds. Assumptions governing these two types of swept manifolds are relatively mild and satisfied in almost all practical applications, while the additional simplification they provide allowed us to obtain a detailed topological description of the two types. In fact, one of our main results gives a complete topological characterization of a regular swept manifold if the topology of its initial manifold is known. Due to the magnitude of potential applications, special attention has been paid to 2-dimensional swept manifolds, that is, swept surfaces. Taking advantage of their low dimensionality, we have performed a more thorough analysis, elucidating several important features. For instance, using the previously mentioned result we have completely resolved the topology of regular swept surfaces. Obviously, this fact has a great practical value since it delimits the range of modeling possibilities of swept surfaces. In an endeavor to extend the nice structure of swept surfaces to higher dimensions, we have introduced yet another subclass of swept manifolds – recursive swept manifolds. Our brief

analysis of recursive swept manifolds has shown that they retain several important properties specific to swept surfaces.

In addition to a theoretical study of swept manifolds, this work contains an algorithmic part in which we focus on the important problem of computing intersections of swept surfaces. A brief description of existing intersection methods, provided at the beginning of the corresponding chapter, shows that they are not directly applicable to swept surfaces. Therefore, we have developed new intersection algorithms aimed specifically at swept surfaces. In the course of designing these algorithms, we made extensive use of general ideas that underlie the most popular types of extant intersection methods, modifying some of their features to satisfy conditions imposed by the structure of swept surfaces. As a result of such an approach, we have obtained three algorithms: the Simple Discretization Algorithm, the Subdivision Algorithm, and the Marching Algorithm; each of them is based on one of the existing types of intersection methods. We have provided detailed descriptions of our algorithms, including complexity analysis, and have shown that they can be quite efficient, although sometimes there may be a need for additional assumptions on swept surfaces. Unfortunately, we have not been able to provide a quantitative comparison between our intersection algorithms and existing methods, partly because of the lack of benchmark problems with a standard representation of the intersecting surfaces that would be applicable in this case. It is possible to look at qualitative behavior though. As we have mentioned, if objects under consideration are swept surfaces without explicit parametrization, extant intersection methods would require some sort of conversion to obtain an appropriate representation, which would considerably impair their performance compared to the swept surface intersection algorithms. If, on the other hand, intersecting surfaces are represented by sweeps in a closed form, some of the existing intersection methods, for example the Gordon-Klein intersector, may perform better than our algorithms, especially when the surfaces are algebraic. We hope that a more

rigorous comparison will become possible in the near future and plan on establishing several intersection problems that would allow us to obtain some quantitative data for our algorithms as well as for existing intersection methods.

As an extension of our work on the intersection problem, we have also presented a novel approach to local characterization of intersections of submanifolds and s -subvarieties of a Euclidean space. The method is based on homology theory and allows us to distinguish between intersection points of different kinds in an effectively computable way, that is, one can algorithmically detect whether a point is an intersection point and whether it belongs to a transversal or tangential intersection. In addition, it is often possible to determine when an intersection point is a boundary point.

To show the practical side of the developed theory and algorithms, we have indicated several possible applications of swept surfaces and their intersections. In particular, we have described how swept manifolds can be used in virtual sculpting and how the corresponding intersection problem becomes a crucial part of the ‘carving’ operations. Also, we have sketched a new method for modeling the heterogeneous structure of biomaterials such as a bone. We use swept surfaces to model strands such as canals that comprise the structure of a bone and employ intersections to allow for branching and merging. Finally, we have indicated how the homological intersection characterization can be applied to analyze and improve higher dimensional geometric models.

It is important to understand that the progress we have achieved in our study of swept manifolds and their intersections is only the first step in our continuing research on swept volumes and their place in computational topology. Much more work has to be done to obtain a complete topological classification of arbitrary swept manifolds and to develop efficient tools for evaluating general swept volumes. For example, finding possible topological structures of non-regular swept manifolds would

be very helpful for proper assessment of regularity conditions. A related important issue is a possibility of generalizing swept surfaces to cover a broader spectrum of important topological configurations. The simple topology of regular swept surfaces does not allow us to represent sophisticated geometric objects; for instance, we had to simulate branching and merging when modeling strands of a bone. So, is it possible to modify the definition of a (regular) swept surface to allow for forking or, more generally, to encompass surfaces of genus greater than one? From a theoretical point of view, this can be easily done by designing a sweep vector field with critical points of the corresponding index. Indeed, considering such a vector fields as a gradient vector field of some Morse function, the necessary topology of the generated swept surface is provided by Morse theory. Unfortunately, carrying out computations around critical points is a nontrivial task with serious obstacles, and a lot of additional analysis needs to be done before this idea can be implemented in practice.

The swept surface intersection algorithms also have a lot of room for improvement. For example, how can we guarantee that the correct topology of the intersection set is resolved? Currently, this kind of question is usually tackled by employing interval arithmetic ([2, 58]), and it seems reasonable to investigate how well the same approach works in the case of swept surfaces. Another issue that needs to be addressed, and which has been mentioned several times, is finding a universal (at least to some extent) criterion for choosing discretization steps. Ad hoc approaches may work quite well, but having a reliable, general method would clearly be beneficial. In addition, improving the performance of our algorithms and finding a way to drop some of the assumptions have a prominent place in our future research.

Obviously, answering all these questions and problems will take a lot of time and resources, but the gain could be quite beneficial. We have mentioned several times that the swept surface intersection problem is qualitatively different from the standard surface-to-surface intersection problem and preserving the original swept

representations is crucial for the efficiency of intersection algorithms. Perhaps an even more important reason for developing methods that handle swept surfaces directly is retaining the given information. Indeed, constructing algebraic approximations for swept surfaces, which is required for most existing intersection methods, completely disregards the underlying sweeps and therefore can lead to serious complications when a thorough intersection analysis is needed. The intersection algorithms we have developed provide a nice basis for further work on the swept surface intersection problem, and we hope that findings that we have presented will entice some scientists and inspire them to invest their efforts in the corresponding research directions.

REFERENCES

- [1] M. A. Akivis and V. V. Goldberg, *Differential Geometry of Varieties with Degenerate Gauss Maps*, ser. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 18. New York: Springer-Verlag, 2004.
- [2] N. Amenta, T. J. Peters, and A. C. Russell, “Computational topology: ambient isotopic approximation of 2-manifolds,” *Theoret. Comput. Sci.*, vol. 305, no. 1-3, pp. 3–15, 2003.
- [3] D. S. Arnon and S. McCallum, “A polynomial-time algorithm for the topological type of a real algebraic curve,” *J. Symb. Comput.*, vol. 5, no. 1-2, pp. 213–236, 1988.
- [4] D. K. Arrowsmith and C. M. Place, *An Introduction to Dynamical Systems*. Cambridge: Cambridge University Press, 1990.
- [5] N. M. Aziz, R. Bata, and S. Bhat, “Surfaces: Bezier surface/surface intersection,” *IEEE Comput. Graph. Appl.*, vol. 10, no. 1, pp. 50–58, 1990.
- [6] C. L. Bajaj, C. M. Hoffmann, R. E. Lynch, and J. E. H. Hopcroft, “Tracing surface intersections,” *Comput. Aided Geom. Design*, vol. 5, no. 4, pp. 285–307, 1988.
- [7] C. L. Bajaj and G. L. Xu, “Nurbs approximation of surface/surface intersection curves,” *Adv. Comput. Math.*, vol. 2, no. 1, pp. 1–21, 1994.
- [8] R. E. Barnhill, G. Farin, M. Jordan, and B. R. Piper, “Surface/surface intersection,” *Comput. Aided Geom. Design*, vol. 4, no. 1-2, pp. 3–16, 1987.
- [9] R. E. Barnhill, “Geometry processing: curvature analysis and surface-surface intersection,” in *Mathematical Methods in Computer Aided Geometric Design (Oslo, 1988)*. Boston, MA: Academic Press, 1989, pp. 51–60.
- [10] M. Bern and D. Eppstein, “Emerging challenges in computational topology,” 1999.
- [11] D. Bespalov, A. Shoukoufandeh, W. Regli, and W. Sun, “Scale-space representation and classification of 3d models,” *ASME J. Computers and Information Science in Engineering*, vol. 3, pp. 315–324, 2003.
- [12] G. D. Birkhoff, *Dynamical Systems*, ser. With an Addendum by Jurgen Moser. American Mathematical Society Colloquium Publications, Vol. IX. Providence, R.I.: American Mathematical Society, 1966.
- [13] D. Blackmore, H. Jiang, and M. Leu, “Modeling of swept solids using sweep differential techniques,” 1992.

- [14] D. Blackmore, M. C. Leu, L. P. Wang, and H. Jiang, "Swept volumes: a retrospective and prospective view," *Neural Parallel Sci. Comput.*, vol. 5, no. 1-2, pp. 81–101, 1997.
- [15] D. Blackmore, M. Leu, and F. Shih, "Analysis and modeling of deformed swept volumes," *CAD*, vol. 26, no. 4, pp. 315–326, 1994.
- [16] D. Blackmore, M. Leu, and L. P. Wang, "The sweep-envelope differential equation algorithm and its application to nc machining verification," *CAD*, vol. 29, no. 9, pp. 629–637, 1997.
- [17] D. Blackmore, Y. Mileyko, M. Leu, W. Regli, and W. Sun, "Computational topology and swept volumes," in *AMS-DIMACS Volume on Computer Aided Design and Manufacturing*, R. Janardin and M. Schmid, Eds. AMS, 2005.
- [18] D. Blackmore, R. Samulyak, and M. C. Leu, "Singularity theory approach to swept volumes," *International Journal of Shape Modeling*, vol. 6, no. 1, pp. 105–129, 2000.
- [19] D. Blackmore and M. C. Leu, "Analysis of swept volume via lie groups and differential equations," *Int. J. Rob. Res.*, vol. 11, no. 6, pp. 516–537, 1992.
- [20] Y. G. Borisovich, N. M. Bliznyakov, T. N. Fomenko, and Y. A. Izrailevich, *Introduction to Differential and Algebraic Topology*, ser. Kluwer Texts in the Mathematical Sciences. Dordrecht: Kluwer Academic Publishers Group, 1995, vol. 9.
- [21] R. H. Crowell and R. H. Fox, *Introduction to Knot Theory*. New York: Springer-Verlag, 1977.
- [22] M. Daniel and A. Nicolas, "A surface-surface intersection algorithm with a fast clipping technique," in *Curves and Surfaces in Geometric Design (Chamonix-Mont-Blanc, 1993)*. Wellesley, MA: A K Peters, 1994, pp. 105–112.
- [23] C. de Boor, *A Practical Guide to Splines*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 2001, vol. 27.
- [24] C. J. A. Delfinado and H. Edelsbrunner, "An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere," *Comput. Aided Geom. Design*, vol. 12, no. 7, pp. 771–784, 1995.
- [25] T. K. Dey, H. Edelsbrunner, and S. Guha, "Computational topology," in *Advances in Discrete and Computational Geometry (South Hadley, MA, 1996)*, ser. Contemp. Math. Providence, RI: Amer. Math. Soc., 1999, vol. 223, pp. 109–143.
- [26] E. DiBenedetto, *Real Analysis*, ser. Birkhäuser Advanced Texts: Basler Lehrbücher. [Birkhäuser Advanced Texts: Basel Textbooks]. Boston, MA: Birkhäuser Boston Inc., 2002.

- [27] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *Discrete Comput. Geom.*, vol. 28, no. 4, pp. 511–533, 2002.
- [28] G. Farin, "Rational B -splines," in *Geometric Modeling (Böblingen, 1990)*, ser. Comput. Graph. Systems Appl. Berlin: Springer, 1991, pp. 115–130.
- [29] G. Farin, "Splines in CAD/CAM," *Surveys Math. Indust.*, vol. 1, no. 1, pp. 39–73, 1991.
- [30] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design*, ser. Computer Science and Scientific Computing. San Diego, CA: Academic Press Inc., 1997.
- [31] G. Farin, *NURBS*. Natick, MA: A K Peters Ltd., 1999.
- [32] G. Farin, "A history of curves and surfaces in CAGD," in *Handbook of Computer Aided Geometric Design*. Amsterdam: North-Holland, 2002, pp. 1–21.
- [33] G. Farin and B. Hamann, "Current trends in geometric modeling and selected computational applications," *J. Comput. Phys.*, vol. 138, no. 1, pp. 1–15, 1997.
- [34] G. Farin and D. Hansford, "Discrete Coons patches," *Comput. Aided Geom. Design*, vol. 16, no. 7, pp. 691–700, 1999.
- [35] G. Farin, J. Hoschek, and M.-S. Kim, Eds., *Handbook of Computer Aided Geometric Design*. Amsterdam: North-Holland, 2002.
- [36] R. T. Farouki, C. Y. Han, J. Hass, and T. W. Sederberg, "Topologically consistent trimmed surface approximations based on triangular patches," *Comput. Aided Geom. Design*, vol. 21, no. 5, pp. 459–478, 2004.
- [37] D. B. Gauld, *Differential Topology*, ser. Monographs and Textbooks in Pure and Applied Mathematics. New York: Marcel Dekker Inc., 1982, vol. 72.
- [38] J. E. Goodman and J. O'Rourke, Eds., *Handbook of Discrete and Computational Geometry*. Boca Raton, FL, USA: CRC Press, Inc., 1997.
- [39] T. A. Grandine and F. W. Klein, IV, "A new approach to the surface intersection problem," *Comput. Aided Geom. Design*, vol. 14, no. 2, pp. 111–134, 1997.
- [40] W. Guoping, S. Jiaguang, and H. Xuanji, "The sweep-envelope differential equation algorithm for general deformed swept volumes," *Comput. Aided Geom. Design*, vol. 17, no. 5, pp. 399–418, 2000.
- [41] E. Gursoz, Y. Choi, and F. Prinz, "Vertex-based representation of non-manifold boundaries," in *Geometric Modeling for Product Engineering*, J. Turner, M. Wozny, and K. Preiss, Eds. North Holland: Elsevier Science, 1990, pp. 107–130.
- [42] A. Hatcher, *Algebraic Topology*. Cambridge: Cambridge University Press, 2002.

- [43] M. W. Hirsch, *Differential Topology*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1994, vol. 33.
- [44] M. W. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1974.
- [45] C. M. Hoffmann, *Geometric and Solid Modeling: An Introduction*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989.
- [46] M. E. Hohmeyer, "A surface intersection algorithm based on loop detection," in *SMA '91: Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*. New York, NY, USA: ACM Press, 1991, pp. 197–207.
- [47] P. M. Hubbard, "Collision detection for interactive graphics applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 3, pp. 218–230, 1995.
- [48] E. Huber and W. Barth, "Surface-to-surface intersection with complete and guaranteed results," in *Developments in Reliable Computing (Budapest, 1998)*. Dordrecht: Kluwer Acad. Publ., 1999, pp. 189–202.
- [49] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 2004, vol. 157.
- [50] T. Kaczyński, M. Mrozek, and M. Ślusarek, "Homology computation by reduction of chain complexes," *Comput. Math. Appl.*, vol. 35, no. 4, pp. 59–70, 1998.
- [51] T. Kaczynski, K. Mischaikow, and M. Mrozek, "Computing homology," *Homology Homotopy Appl.*, vol. 5, no. 2, pp. 233–256 (electronic), 2003.
- [52] M. Karoubi and C. Leruste, *Algebraic Topology via Differential Geometry*, ser. London Mathematical Society Lecture Note Series. Cambridge: Cambridge University Press, 1987, vol. 99.
- [53] G. A. Kreizis and N. M. Patrikalakis, "Rational polynomial surface intersections," in *Proc. 17th ASME Design Automation Conf.*, vol. 2. New York: ASME, 1991, pp. 43–53.
- [54] G. A. Kreizis, N. M. Patrikalakis, and F. E. Wolter, "Topological and differential-equation methods for surface intersections," *CAD*, vol. 24, no. 1, pp. 41–55, 1992.
- [55] S. Krishnan and D. Manocha, "An efficient surface intersection algorithm based on lower-dimensional formulation," *ACM Trans. Graph.*, vol. 16, no. 1, pp. 74–106, 1997.
- [56] T. Larsson and T. Akenine-Mller, "Collision detection for continuously deforming bodies," 2001.

- [57] D. Lasser, "Intersection of parametric surfaces in the Bernstein-Bezier representation," *Comput. Aided Des.*, vol. 18, no. 4, pp. 186–192, 1986.
- [58] H. Lopes and S. Pesco, "Single-valued tubular surface intersection using interval arithmetic," in *Mathematical Methods for Curves and Surfaces (Oslo, 2000)*, ser. Innov. Appl. Math. Nashville, TN: Vanderbilt Univ. Press, 2001, pp. 253–262.
- [59] G. Lukács, "The generalized inverse matrix and the surface-surface intersection problem," in *Theory and Practice of Geometric Modeling (Blaubeuren, 1988)*. Berlin: Springer, 1989, pp. 167–185.
- [60] M. A. H. MacCallum, "An ordinary differential equation solver for REDUCE," in *Symbolic and Algebraic Computation (Rome, 1988)*, ser. Lecture Notes in Comput. Sci. Berlin: Springer, 1989, vol. 358, pp. 196–205.
- [61] D. Manocha and S. Krishnan, "Algebraic pruning: a fast technique for curve and surface intersection," *Comput. Aided Geom. Design*, vol. 14, no. 9, pp. 823–845, 1997.
- [62] D. Manocha, A. Varshney, and H. Weber, "Evaluating surface intersections in lower dimensions," in *Curves and Surfaces in Geometric Design (Chamonix-Mont-Blanc, 1993)*. Wellesley, MA: A K Peters, 1994, pp. 327–334.
- [63] W. S. Massey, *A Basic Course in Algebraic Topology*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1991, vol. 127.
- [64] Y. Matsumoto, *An Introduction to Morse Theory*, ser. Translations of Mathematical Monographs. Providence, RI: American Mathematical Society, 2002, vol. 208.
- [65] J. N. McDonald and N. A. Weiss, *A Course in Real Analysis*. San Diego, CA: Academic Press Inc., 1999.
- [66] W. W. Meyer, "Algebraic parametrization for quadric surface intersections," in *Geometric Aspects of Industrial Design (Wright-Patterson Air Force Base, OH, 1990)*. Philadelphia, PA: SIAM, 1992, pp. 1–10.
- [67] J. Milnor, *Morse Theory*, ser. Based on lecture notes by M. Spivak and R. Wells. Annals of Mathematics Studies, No. 51. Princeton, N.J.: Princeton University Press, 1963.
- [68] J. Moser, Ed., *Dynamical Systems, Theory and Applications*. Berlin: Springer-Verlag, 1975.
- [69] C. Mow, T. J. Peters, and N. F. Stewart, "Specifying useful error bounds for geometry tools: an intersector exemplar," *Comput. Aided Geom. Design*, vol. 20, no. 5, pp. 247–251, 2003.
- [70] G. Müllenheim, "A procedure for determining starting points for a surface/surface intersection algorithm," in *Curves and Surfaces (Chamonix-Mont-Blanc, 1990)*. Boston, MA: Academic Press, 1991, pp. 337–340.

- [71] J. R. Munkres, *Elementary Differential Topology*, ser. Lectures given at Massachusetts Institute of Technology, Fall. Princeton, N.J.: Princeton University Press, 1966, vol. 1961.
- [72] J. R. Munkres, *Topology: A First Course*. Englewood Cliffs, N.J.: Prentice-Hall Inc., 1975.
- [73] J. R. Munkres, *Elements of Algebraic Topology*. Menlo Park, CA: Addison-Wesley Publishing Company, 1984.
- [74] J. O'Rourke, *Computational Geometry in C*. New York, NY, USA: Cambridge University Press, 1998.
- [75] R. S. Palais, *Real Algebraic Differential Topology. Part I*, ser. Mathematics Lecture Series. Wilmington, Del.: Publish or Perish Inc., 1981, vol. 10.
- [76] N. M. Patrikalakis, "Surface-to-surface intersections," *IEEE Comput. Graph. Appl.*, vol. 13, no. 1, pp. 89–95, 1993.
- [77] N. M. Patrikalakis, "Interrogation of surface intersections," in *Geometry Processing for Design and Manufacturing*. Philadelphia, PA: SIAM, 1992, pp. 161–185.
- [78] Q. S. Peng, "An algorithm for finding the intersection lines between two B -spline surfaces," *Zhejiang Daxue Xuebao*, no. Special issue on computational geometry, pp. 85–98, 1984.
- [79] L. Perko, *Differential Equations and Dynamical Systems*, ser. Texts in Applied Mathematics. New York: Springer-Verlag, 2001, vol. 7.
- [80] L. Piegl and W. Tiller, *The NURBS Book*. London, UK: Springer-Verlag, 1995.
- [81] L. A. Piegl, "Constructive geometric approach to surface-surface intersection," in *Geometry Processing for Design and Manufacturing*. Philadelphia, PA: SIAM, 1992, pp. 137–159.
- [82] H. Pottmann and G. Farin, "Developable rational Bézier and B -spline surfaces," *Comput. Aided Geom. Design*, vol. 12, no. 5, pp. 513–531, 1995.
- [83] M. Pratt and A. D. Geisow, "Surface/surface intersection problems," in *The Mathematics of Surfaces (Manchester 1984)*, vol. 6. New York: The Clarendon Press Oxford University Press, 1986, pp. 117–142.
- [84] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.
- [85] G. M. Rassias, Ed., *Differential Topology-Geometry and Related Fields, and Their Applications to the Physical Sciences and Engineering*, ser. Teubner-Texte zur Mathematik [Teubner Texts in Mathematics]. Leipzig: BSB B. G. Teubner Verlagsgesellschaft, 1985, vol. 76.

- [86] W. Regli, J. Han, and M. Pratt, "Algorithms for feature recognition from solid models: a status report," *IEEE Trans. Robotics and Automation*, vol. 16, pp. 982–996, 2000.
- [87] J. Rossignac and M. O'Connor, "Sgc: A dimension-independent model for pointsets with internal structures and incomplete boundaries," in *Geometric Modeling for Product Engineering*, J. Turner, M. Wozny, and K. Preiss, Eds. North Holland: Elsevier Science, 1990, pp. 145–180.
- [88] C. L. Sabharwal, "A fast implementation of surface/surface intersection algorithm," in *SAC '94: Proceedings of the 1994 ACM Symposium on Applied Computing*. New York, NY, USA: ACM Press, 1994, pp. 333–337.
- [89] T. Sakkalis, "The topological configuration of a real algebraic curve," *Bull. Australian Math. Soc.*, vol. 43, no. 1, pp. 37–50, 1991.
- [90] T. W. Sederberg and R. J. Meyers, "Loop detection in surface patch intersections," *Comput. Aided Geom. Design*, vol. 5, no. 2, pp. 161–171, 1988.
- [91] P. Selick, *Introduction to Homotopy Theory*, ser. Fields Institute Monographs. Providence, RI: American Mathematical Society, 1997, vol. 9.
- [92] T. E. Stoyanov, "Marching along surface/surface intersection curves with an adaptive step length," *Comput. Aided Geom. Design*, vol. 9, no. 6, pp. 485–489, 1992.
- [93] G. Vegter, "Computational topology," in *Handbook of Discrete and Computational Geometry*, ser. CRC Press Ser. Discrete Math. Appl. Boca Raton, FL: CRC, 1997, pp. 517–536.
- [94] K. Weiler, "The radial edge data structure: A topological representation for non-manifold geometric boundary modeling," in *Geometric Modeling for CAD Applications*, M. Wozny, H. McLaughlin, and J. Encarnacao, Eds. North Holland: Elsevier Science, 1988, pp. 3–36.
- [95] G. W. Whitehead, *Elements of Homotopy Theory*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1978, vol. 61.
- [96] S. Willard, *General Topology*. Mineola, NY: Dover Publications Inc., 2004.
- [97] T. Yamamoto, Ed., *Proceedings of the International Symposium on Computational Mathematics*. Amsterdam: Elsevier Science B.V., 1991.
- [98] A. Zomorodian and H. Edelsbrunner, "Fast software for box intersections," in *SCG '00: Proceedings of the Sixteenth Annual Symposium on Computational Geometry*. New York, NY, USA: ACM Press, 2000, pp. 129–138.