

# Chapter 1

## Introduction and Overview

This book studies geometry methodically from an *analytical*, i.e., coordinate-based, viewpoint. In many settings this approach simplifies the computer representation of geometric data. We shall not confine ourselves to linear problems. This is not only appealing from a theoretical viewpoint, it is also practically motivated by advances in computer algebra and the availability of fast computer hardware.

In Chapter 2 we will lay some mathematical foundations. First, we will introduce the language of *projective geometry*, which is very well suited for many geometric applications. Since this is not usually covered in standard introductory courses in mathematics, we briefly discuss the central concepts of projective spaces and projective transformations. We will also introduce the notion of *convexity* in this chapter.

Our analytical approach motivates the structure of this book. It is centered around questions about algorithms which solve systems of equations and their increasingly complex variations with regard to the required mathematical tools.

### 1.1 Linear Computational Geometry

Most algorithms described in this book are based on *Gaussian elimination*, a core topic in any linear algebra course. In geometric language Gaussian elimination is a procedure which takes a set of affine hyperplanes,  $H_1, \dots, H_k$ , in the vector space  $K^n$  as input, where  $K$  is an arbitrary field. If

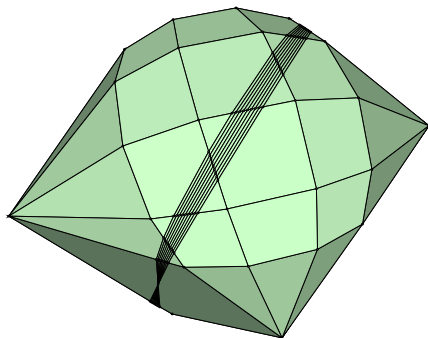
$$A = H_1 \cap \dots \cap H_k \tag{1.1}$$

the output can be an (affine) basis for  $A$ , or simply its dimension.

Our foray through computational geometry begins with the real numbers and the transition from equalities to inequalities. Consider for every hyperplane

$$H_i = \left\{ x \in \mathbb{R}^n : \sum_{j=1}^n a_{ij} x_j = b_i \right\}$$

**Fig. 1.1** An example of a bounded polyhedron in  $\mathbb{R}^3$ . This particular polyhedron is a polytope which is dual to a zonotope. The belt-like strip in the middle has several very thin facets



the closed half-space

$$H_i^+ = \left\{ x \in \mathbb{R}^n : \sum_{j=1}^n a_{ij}x_j \geq b_i \right\}.$$

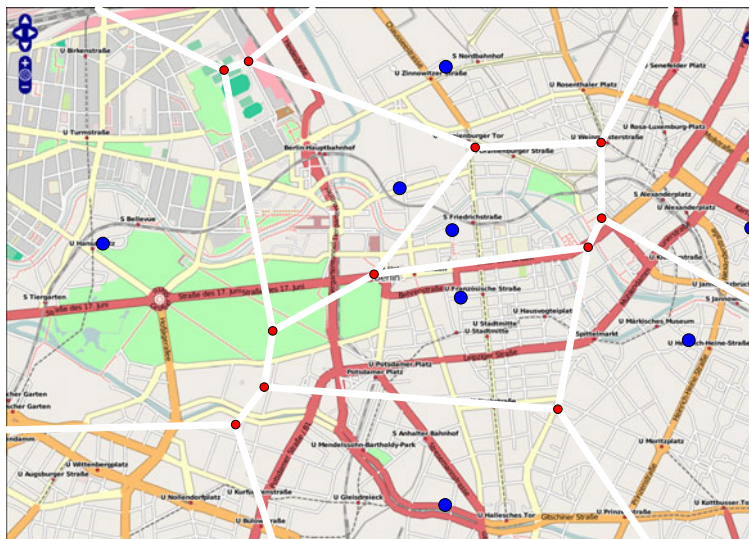
The intersection  $P = \bigcap_{i=1}^k H_i^+$  defines a (*convex*) *polyhedron* (see Fig. 1.1 for an example in  $\mathbb{R}^3$ ).

Polyhedra are fundamental to computational geometry and linear optimization. In higher dimensions, the combinatorial variety of polyhedra is considerably larger than that suggested by lower dimensional images, such as Fig. 1.1. One of the fundamental questions when determining the complexity of many algorithms is, what is the maximum number of vertices that a polyhedron defined by  $k$  linear inequalities can have? This question was first answered in 1970 by the *Upper-bound Theorem*. The proof (in a somewhat weaker formulation, see Theorem 3.46) and the explanation of the underlying geometric structure is the first goal of this book. This result is particularly important for computational geometry because we can use it to obtain complexity estimates for several algorithms.

In Chapter 3 we systematically study the properties of polytopes (face lattice, polarity, combinatorics of polytopes) up to *Euler's formula* and the *Dehn–Sommerville equations*. At the end of the chapter we illustrate some of the concepts with the geometric software `polymake`. We will also use this and other software as an aid to understanding the algorithms presented in later chapters.

The core of many mathematical applications is linear optimization, which addresses the problem of computing the minimum or maximum of a linear objective function on a polyhedron  $P$  (given by linear inequalities). For computational solutions it is important to note that the polyhedron can be empty, or the objective function can be unbounded on  $P$ . In Chapter 4 we give a brief introduction to the relevant aspects of linear optimization. In particular, we discuss the theoretically and practically important *simplex algorithm*. Our main focus (as throughout this text) will be from the geometric perspective.

An interesting computational problem of polytope theory is determining the *entire* set of vertices and rays of a polyhedron defined by a given set of inequalities. Using the duality theory described in Section 3.3, this is equivalent to determining



**Fig. 1.2** The solution to the post office problem for ten branches of the Deutsche Post AG in Berlin (two of which are not in the visible part of the openstreetmap.org map). OpenStreetMap is open data, licensed under the Open Data Commons Open Database License (ODbL). © OpenStreetMap contributors

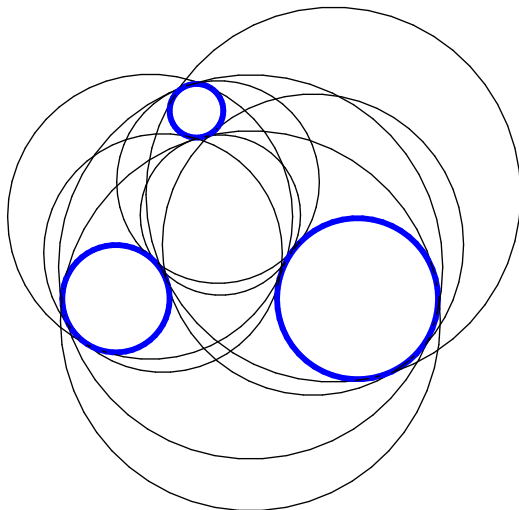
a minimal system of inequalities which define the convex hull of a point set. We devote Chapter 5 to the *convex hull problem*. For applications it is important to note that actually computing solutions to this problem becomes difficult in higher dimensions (simply because of the large output predicted by the Upper-bound Theorem). A general approach for efficient algorithms is the *divide-and-conquer* principle. We illustrate this by applying it to the computation of convex hulls in the plane.

Next, we examine *Voronoi diagrams* and the corresponding dual *Delone subdivisions*. Given an arbitrary point set  $S = \{s^{(1)}, \dots, s^{(m)}\}$  in the  $n$ -dimensional space  $\mathbb{R}^n$ , the *Voronoi region* corresponding to a point  $s^{(i)}$  comprises those points of  $\mathbb{R}^n$  which are no further from  $s^{(i)}$  (with respect to Euclidean distance) than from any other point of  $S$ .

In Chapter 6 we first show how convex hull algorithms can be used to compute Voronoi diagrams in arbitrary dimensions. Afterwards, we concentrate again on the planar case and present the *beach line algorithm*. Knowledge of abstract data types is an advantage for this, so the most important principles will be explained. For a more in depth discussion of common data structures the reader can refer to the recommended literature.

Voronoi diagrams can be used to solve the so-called *post office problem*; a classical application of computational geometry. Given a finite set of points  $S \subseteq \mathbb{R}^2$ , we efficiently compute for each point  $p \in \mathbb{R}^2$  the point  $s \in S$  which minimizes the Euclidean distance  $\|p - s\|$ . The points of  $S$  can be interpreted as post offices and the points  $p$  as customers. See Fig. 1.2. Of course, one can naively examine every point combination (which is efficient if there is only one customer). However, one

**Fig. 1.3** Eight (Apollonius-)circles which touch the three given circles



should interpret the problem as if the postal service wants to create an information system which quickly provides answers for a large group of customers, assuming that the positions of the post offices do not change.

In many applications, the Voronoi diagrams appear in the dual form. Therefore, in Chapter 7 we examine Delone subdivisions and triangulations. A Delone triangulation of the convex hull of a given point set  $S$  defined in this manner is in several respects optimal in comparison to all other triangulations of  $S$ . We show that in arbitrary dimensions the maximal radius of the circumsphere is minimized. Again, we will examine the planar case in greater detail.

## 1.2 Non-linear Computational Geometry

The second part of this book is dedicated to non-linear problems. In Chapter 8 we advance from systems of linear equations and inequalities to systems of polynomial equations, and thus into basic algebraic geometry. After this it would be natural to discuss systems of polynomial inequalities, i.e., semi-algebraic geometry, but this would be beyond the scope of this book. At relevant points, we content ourselves with some remarks on polynomial inequalities.

As a good example of a non-linear problem, consider *Apollonius' problem* (Apollonius of Perga ca. 260–190 BC): Given three circles  $C_1$ ,  $C_2$  and  $C_3$  in the plane, compute another circle that touches each of the previous ones (see Fig. 1.3). If the circles  $C_1$ ,  $C_2$  and  $C_3$  are in general position there exist eight (possibly complex) solutions. As a possible application, the circles could be interpreted as distance requirements for a set of given points. We will come back to this in later chapters.

In the second part of this book the algorithmic focus is on *Gröbner bases* (Chapter 9). These allow us to solve arbitrary systems of polynomial equations exactly (Chapter 10).

In Chapter 8 we give an introduction to resultants, planar affine and projective algebraic curves and to Bézout's Theorem. We conclude this chapter by illustrating some of these results using `Maple`.

A fundamental algorithmic problem, which is covered in Chapter 9 and will later be the basis of the method we use to solve systems of polynomial equations, is the *Ideal membership problem*. Given polynomials  $f$  and  $g_1, \dots, g_r$  in the polynomial ring  $K[x_1, \dots, x_n]$  over the field  $K$ , is  $f$  in the ideal generated by  $g_1, \dots, g_r$  or not? In general this question cannot be directly answered. This motivates the study of ideal bases with special properties, called Gröbner bases, for which the algorithmic decision problem becomes very simple. Therefore, given a polynomial ideal, the main task is to compute a Gröbner basis for this ideal. We will also develop the relevant theoretical background in computational algebra.

In Chapter 10 we discuss how Gröbner bases are used in the computational solution of systems of polynomial equations. To do this, we first give a brief introduction to the computer algebra system `Singular`. From a theoretical viewpoint, Hilbert's Nullstellensatz plays a fundamental role; it establishes a connection between *geometry* (in the sense of polynomial roots) and *algebra* (in the sense of polynomial ideals). Solutions to systems of polynomial equations may then be obtained from roots of univariate polynomials using elimination ideals. To conclude this chapter we present the most simple case of the *Conti-Traverso algorithm*, which illustrates how to use Gröbner basis techniques in the study of integer linear programs.

## 1.3 Applications

In the third part of this book we discuss some selected applications of the theoretical results presented earlier.

In Chapter 11 we approach the problem of reconstructing a curve from a given set of points lying on it. We use the concepts of the medial axis and the "local feature size" to evaluate the relationship between the (unknown) curve and the (given) points. The theoretical background from the first part of this book is sufficient for this application.

In Chapter 12 we treat lines in 3- and  $n$ -dimensional space. Lines in 3-dimensional space often occur in computational geometry and computer graphics, e.g., in visible surface determination. Although (affine) lines in  $\mathbb{R}^3$  are polyhedral objects, questions regarding intersections of lines are intrinsically non-linear. We study these geometric problems by looking at the algebraic characteristics of the *Plücker coordinates* (also known as Grassmann coordinates) of a line. We close this chapter with an example that illustrates the role played by 3-dimensional lines in computer graphics.

Finally, in Chapter 13 we give small insights into applications concerning *Global Positioning Systems (GPS)* and robotics. The functionality of GPS relies on several

satellites continuously orbiting the earth so that at least four of them are always accessible from (almost) any position on the Earth's surface. Determining positions using GPS is closely related to a 3-dimensional version of the Apollonius problem, as we will see in Chapter 13. Furthermore, we discuss, sometimes via computer, some fundamental problems of kinematics.

## Appendix

In three out of four parts of the appendix we provide foundations for algebraic structures, convex analysis as well as algorithms and complexity. These sections also standardize our notation. The fourth part of the appendix introduces software packages that are used throughout the book: `polymake`, `Maple` and `Singular`. We also mention `CGAL` and `Sage`.

**The Structure of This Text** This book consists of more material than a standard one semester course can cover. Hence, this text may be used in several different ways as a basis for a series of lectures. The following compilations are meant as a suggestion:

- “Linear Computational Geometry”: Chapters 2 to 7, Chapters 11 and 12. Please note that Chapter 12 uses elimination techniques from Part II of this book. However, the use of `Maple` or `Singular` allows us to treat examples without having a detailed knowledge of the theoretical concepts.
- “Non-linear Computational Geometry”: This is complementary to the selection above, hence consisting of Chapters 8 to 10 of the second part of this book and Chapter 13 from the applications part. The amount of material is suitable for a compact course as a follow-up to the course “Linear Computational Geometry”.
- “Cross-section of Polyhedral and Algebraic Methods”: Chapters 2, 3, 5 or 6, 8 until 10, 12, 13. Sections 9.5 and 10.6 may be left out in this.

Every chapter ends with a small section “Remarks” which references further suggested reading and historical remarks. All figures in this book were produced using the mentioned software and using `METAPOST` [62].