67 SOFTWARE

Michael Joswig and Benjamin Lorenz

INTRODUCTION

This chapter is intended as a guide through the ever-growing jungle of geometry software. Software comes in many guises. There are the fully fledged systems consisting of a hundred thousand lines of code that meet professional standards in software design and user support. But there are also the tiny code fragments scattered over the Internet that can nonetheless be valuable for research purposes. And, of course, the very many individual programs and packages in between.

Likewise today we find a wide group of users of geometry software. On the one hand, there are researchers in geometry, teachers, and their students. On the other hand, geometry software has found its way into numerous applications in the sciences as well as industry. Because it seems impossible to cover every possible aspect, we focus on software packages that are interesting from the researcher's point of view, and, to a lesser extent, from the student's.

This bias has a few implications. Most of the packages listed are designed to run on UNIX/Linux[1] machines. Moreover, the researcher's genuine desire to understand produces a natural inclination toward open source software. This is clearly reflected in the selections below. Major exceptions to these rules of thumb will be mentioned explicitly.

In order to keep the (already long) list of references as short as possible, in most cases only the Web address of each software package is listed rather than manuals and printed descriptions, which can be found elsewhere. At least for the freely available packages, this allows one to access the software directly. On the other hand, this may seem careless, since some Web addresses are short-lived. This disadvantage usually can be compensated by relying on modern search engines.

The chapter is organized as follows. We start with a discussion of some technicalities (independent of particular systems). Since, after all, a computer is a technical object, the successful use of geometry software may depend on such things. The main body of the text consists of two halves. First, we browse through the topics of this Handbook. Each major topic is linked to related software systems. Remarks on the algorithms are added mostly in areas where many implementations exist. Second, some of the software systems mentioned in the first part are listed in alphabetical order. We give a brief overview of some of their features. The libraries `CGAL` [F$^+$16] and `LEDA` [led] are discussed in depth in Chapter 68. In particular, the `CGAL` implementations for the fundamental geometric algorithms in dimensions

[1] No attempt is made to comment on differences between various UNIX platforms. According to `distrowatch.com`, today's default UNIX-like platform is probably close to Debian/Ubuntu Linux. Most other Linux distributions are not very different for our purposes. FreeBSD and its derivative MacOS X come quite close. Restricting to one kind of operating system is less of a problem nowadays. This is due to a multitude of virtualization concepts that provide the missing links among several platforms. We would like to mention `docker` [doc] as one project that is becoming increasingly popular.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

two and three mark the state of the art and should be considered as the standard reference.

This chapter is a snapshot as of early 2016. It cannot be complete in any sense. Even worse, the situation is changing so rapidly that the information given will be outdated soon. All this makes it almost impossible for the nonexpert to get any impression of what software is available. Therefore, this is an attempt to provide an overview in spite of the obvious complications. For historical reasons we kept pointers to some codes or web pages, which can be interesting, even if they are a bit outdated.

GLOSSARY

Software can have various forms from the technical point of view. In particular, the amount of technical knowledge required by the user varies considerably. The notions explained below are intended as guidelines.

Stand-alone software: This is a program that usually comes "as-is" and can be used immediately if properly installed. No programming skills are required.

Libraries: A collection of software components that can be accessed by writing a main program that calls functions implemented in the library. Good libraries come with example code that illustrates how (at least some of) the functions can be used. However, in order to exploit all the features, the user is expected to do some programming work. On the other hand, libraries have the advantage that they can be integrated into existing code. Some stand-alone programs can also be used as libraries; if they appear in this category, too, then there are substantial differences between the two versions, or the library has additional functionality.

General-purpose systems and their modules: Computer algebra systems like `Maple` [map15], `Mathematica` [mat15] and `SageMath` [S$^+$16] form integrated software environments with elaborate user interfaces that incorporate numerous algorithms from essentially all areas of mathematics. This chapter lists only functionality or extensions that the authors find particularly noteworthy.

Additional Web services: There are very many software overviews on the Web. A few of them that are focused on a specific topic are mentioned in the main text. Some webpages offer additional pieces of source code. Even better, nowadays websites may offer access to some compute server running in the background. This is one of the two most common ways for mathematical software to arrive at mobile devices (the other being `JavaScript`). A short list of particularly useful websites is given further below.

GENERAL SOURCES

For each of the major general-purpose computer algebra systems there exists a website with many additional packages and individual solutions. See the Web addresses of the respective products. In particular, `Sage` [S$^+$16] offers a free cloud service that is suitable for collaborative work. A particularly powerful general source is `Wolfram Alpha` [Wol16], which employs `Mathematica` [mat15] and calls itself a "computational knowledge engine." It is not even restricted to mathematics.

There are several major websites that are of general interest to the discrete and computational geometry community. Some of them also collect references to software, which are updated more or less frequently. One popular example is Eppstein's "Geometry Junkyard" [Epp12].

For those who are beginning to learn how to develop geometry software it will probably be too hard to do so by reading the source code of mature systems only. O'Rourke's book [O'R98] can help fill this gap. Its numerous example programs in C and Java are also electronically available [O'R00].

"The Stony Brook Algorithm Repository" maintained by Skiena [Ski01] is still useful to some extent. Section 1.6 is dedicated to computational geometry, and it contains links to implementations.

In addition to software for doing geometry it is sometimes equally important to get at interesting research data. A particularly comprehensive, and fast growing, site is `Imaginary` [ima]. The `DGD Gallery`, where "DGD" is short for "Discretization in Geometry and Dynamics," is a new site with geometric models of various kinds [dgd].

ARITHMETIC

Depending on the application, issues concerning the arithmetic used for implementing a geometric algorithm can be essential. Using any kind of exact arithmetic is expensive, but the overhead induced also strongly depends on the application. A principal choice for exact arithmetic is *unlimited precision integer* or *rational* arithmetic as implemented in the `GNU Multiprecision Library (GMP)` [gmp15]. Several projects also use the `Multiple Precision Integers and Rationals (MPIR)` [mpi15] that was forked from `GMP`. They are implemented as C-libraries, but they can be used from many common programming languages via corresponding bindings. However, some problems require nonrational constructions. To cover such instances, libraries like `Core` [YD10] and LEDA ([led], Chapter 68) offer special data types that allow for exact computation with certain radical expressions.

Geometric algorithms often rely on a few primitives like: Decide whether a point is on a hyperplane or, if not, tell which side it lies on. Thus exact coordinates for geometric objects are sometimes less important than their true relative position. It is therefore natural to use techniques like interval arithmetic. This may be based on packages for multiple precision floating-point numbers such as `GNU Multiple Precision Floating-Point Reliably (MPFR)` [mpf16]. *Floating-point filters* can be understood as an improved kind of interval arithmetic that employs higher precision or exact methods if needed. For more detailed information see Chapter 45.

Yet another arithmetic concept is the following: Compute with machine size integers but halt (or trigger an exception) if an overflow occurs. Typically such an implementation depends on the hardware and thus requires at least a few lines of assembler code. Useful applications for such an approach are situations where the overflow signals that the computation is expected to become too large to finish in any reasonable amount of time. For instance, `hull` [Cla] uses exact integer arithmetic for convex hull computation and signals an overflow. Going one step further, `Normaliz` [BSSR16] first computes with machine size integers but when an overflow occurs only the corresponding computation step, e.g., a vector or matrix multiplication, will be restarted with arbitrary precision. When the result still does not fit the machine size, the whole computation will be restarted with arbitrary precision integers.

Instead of using a form of exact arithmetic, some implementations perform combinatorial post-processing in order to repair flawed results coming from rounding errors. An example is the convex hull code `qhull` by Barber, Dobkin, and Huhdanpaa [BDH16]. Usually, this is only partially successful; see the discussion on the corresponding Web page [BDH01] in `qhull`'s documentation.

FURTHER TECHNICAL REMARKS

While the programming language in which a software package is implemented often does not affect the user, this can obviously become an issue for the administrator who does the installation. Many of the software systems listed below are distributed as source code written in C or C++. Additionally, some of the larger packages are offered as precompiled binaries for common platforms.

C is usually easy. If the source code complies with the ANSI standard, it should be possible to compile it with any C compiler. Some time ago the situation was different for C++. However, current compilers (e.g., `gcc` and `clang`) quickly converge to the C++14 and C++1z standards.

While it is by no means a new programming language, in recent years Python became increasingly popular. For basic mathematical computations there are some interesting libraries, including `numpy` and `scipy`. Further, `SageMath` uses (the old version 2 of) Python as its scripting language. Currently, there is a bit of a hype around the Julia language, which is interesting for doing mathematics and became popular for scientific computing. For computational geometry there is a GitHub site [jul16] with code for basic constructions.

JavaScript deserves to be mentioned, too. Currently, this is the only programming language supported by most web browsers without the need of plugins or extensions. This makes it particularly interesting for software to be deployed on mobile devices.

67.1 SOFTWARE SORTED BY TOPIC

This section should give a first indication of what software to use for solving a given problem. The subsections reflect the overall structure of the whole Handbook. References to `CGAL` [F$^+$16] and `LEDA` [led] usually are omitted, since these large projects are covered in detail in Chapter 68.

67.1.1 COMBINATORIAL AND DISCRETE GEOMETRY

This section deals with software handling the combinatorial aspects of finitely many objects, such as points, lines, or circles, in Euclidean space. Polytopes are described in Section 67.1.2.

STAND-ALONE SOFTWARE

The simplest geometric objects are clearly points. Therefore, essentially all geometry software can deal with them in one way or another. A key concept to many nontrivial properties of finite point sets in $\mathbb{R}^d$ is the notion of an *oriented matroid*.

For oriented matroid software and the computation of the set of all triangulations of a given point set, see TOPCOM by Rambau [Ram14]. In order to have correct combinatorial results, arbitrary precision arithmetic is essential.

Stephenson's CirclePack [Ste09] can create, manipulate, store, and display circle packings.

Lattice points in convex polytopes are related to combinatorial optimization and volume computations; see Beck and Robins [BR07]. Further there is a connection to commutative algebra, e.g., via monomial ideals and Gröbner bases; see Bruns and Gubeladze [BG09]. Software packages that specialize in lattice point computations include barvinok [Ver16], LattE [BBDL$^+$15], and Normaliz [BSSR16]. Moreover, polymake offers interfaces to several of the above and adds further functionality. Various volume computation algorithms for polytopes, using exact and floating-point arithmetic, are implemented in vinci by Büeler, Enge, and Fukuda [BEF03].

Dynamic geometry software allows the creation of geometrical constructions from points, lines, circles, and so on, which later can be rearranged interactively. Objects depending, e.g., on intersections, change accordingly. Among other features, such systems can be used for visualization purposes and, in particular, also for working with polygonal linkages. An open source system in this area is GeoGebra [Geo15]. Commercial products include Cabrilog's Cabri 3D [Cab07], the Geometer's Sketchpad [Geo13] as well as Cinderella [RGK16] by Kortenkamp and Richter-Gebert.

Graph theory certainly is a core topic in discrete mathematics and therefore naturally plays a role in discrete and computational geometry. There is an abundance of algorithms and software packages, but they are not especially well suited to geometry, and so they are skipped here. Often symmetry properties of geometric objects can be reduced to automorphisms of certain graphs. While the complexity status of the graph isomorphism problem remains open, bliss [JK15] by Junttila and Kaski or McKay's nauty [MP16] work quite well for many practical purposes.

LIBRARIES

Ehrhart polynomials and integer points in polytopes are also accessible via Loechner's PolyLib [Loe10] and the Integer Set Library [isl16].

ADDITIONAL WEB SERVICES

The new VaryLab web site [RSW] offers interactive functions for surface optimization. This includes, e.g., discrete conformal mappings of euclidean, spherical, and hyperbolic surfaces based on circle patterns and circle packings. For polyominoes, see Eppstein's Geometry Junkyard [Epp03] and Chapter 14.

67.1.2 POLYTOPES AND POLYHEDRA

In this section we discuss software related to the computational study of convex polytopes. The distinction between polytopes and unbounded polyhedra is not essential since, up to a projective transformation, each polyhedron is the Minkowski sum of an affine subspace and a polytope.

A key problem in the algorithmic treatment of polytopes is the convex hull problem, which is addressed in Section 67.1.4.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

STAND-ALONE SOFTWARE

polymake [GJ+16] is a comprehensive framework for dealing with polytopes in terms of vertex or facet coordinates as well as on the combinatorial level. The system offers a wide functionality, which is not restricted to polytopes and which is further augmented by interfacing to many other programs operating on polytopes. Among the combinatorial algorithms implemented is a fast method for enumerating all the faces of a polytope given in terms of the vertex-facet incidences by Kaibel and Pfetsch [KP02].

Normaliz by Bruns et al. [BSSR16] was initially designed for combinatorial computations in commutative algebra. However, it offers many functions for vector configurations, lattice polytopes, and rational cones.

Triangulations of polytopes can be rather large and intricate. Rambau's TOPCOM [Ram14] is primarily designed to examine the set of all triangulations of a given polytope (or arbitrary point configurations). Pfeifle and Rambau [PR03] combined TOPCOM with polymake to compute secondary fans and secondary polytopes; see also Section 16.6. Gfan by Jensen is software for dealing with Gröbner fans and derived objects, but it can also compute secondary fans [Jen].

The combinatorial equivalence of polytopes can be reduced to a graph isomorphism problem. As mentioned above, graph isomorphism can be checked by nauty [MP16] or bliss [JK15].

The Geometry Center's Geomview [Geo14] or jReality [WGB+09] can both be used for, among others, the visualization of 3-polytopes and (Schlegel diagrams of) 4-polytopes.

LIBRARIES

Normaliz [BSSR16] is, in fact, primarily a C++ library, but it comes with executables ready for use. The Parma Polyhedral Library (PPL) offers a fast implementation of the double description method for convex hull computations [BHZ16]. PolyLib [Loe10] is a library for working with rational polytopes; it is primarily designed for computing Ehrhart polynomials. polymake [GJ+16] comes with a C++ template library for container types that extend the Standard Template Library (STL). This allows one to access all the functionality, including the interfaced programs, from the programmer's own code. Further, the library offers a variety of container classes suitable for the manipulation of polytopes.

GENERAL PURPOSE SYSTEMS AND THEIR MODULES

convex by Franz [Fra09] is a package for the investigation of rational polytopes and polyhedral fans in Maple.

ADDITIONAL WEB SERVICES

The website Polyhedral.info [pol] contains a long list of codes that are related to polyhedral geometry and applications.

67.1.3 COMBINATORIAL AND COMPUTATIONAL TOPOLOGY

Recent years saw an increasing use of methods from computational topology in discrete and computational geometry. A basic operation is to compute the homology of a finite simplicial complex (or similar). Although polynomial time methods (in

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

the size of the boundary matrices) are known for most problems, the (worst case exponential) elimination methods seem to be superior in practice; see Dumas et al. [DHSW03]. Implementations with a focus on the combinatorics include the `GAP` package `simpcomp` by Effenberger and Spreer [ES16] as well as `polymake` [GJ+16].

STAND-ALONE SOFTWARE

The Computational Homology Project `CHomP` [cho] is specifically designed for the global analysis of nonlinear spaces and nonlinear dynamics.

`SnapPea` by Weeks [Wee00] is a program for creating and examining hyperbolic 3-manifolds. On top of this, Culler and Dunfield built `SnapPy` (which is pronounced [ˈsnæpaɪ]) [CDW]. A comprehensive system for normal surface theory of 3-manifolds is `Regina` by Burton and his co-authors [BBP+14].

`Geomview`'s [Geo14] extension package `Maniview` can be used to visualize 3-manifolds from within.

LIBRARIES

Persistent homology is a particularly powerful computational topology technique applicable to a wide range of applications; see Chapter 24. `PHAT` is a C++ library with methods for computing the persistence pairs of a filtered cell complex represented by an ordered boundary matrix with mod 2 coefficients [BKR]. Another C++ library with a similar purpose is `Dionysus` [Mor]. A somewhat broader software project for topological data analysis is `GUDHI` [BGJV], and it also does persistence. For those who prefer Java over C or C++ there is `JavaPlex` [TVJA14].

ADDITIONAL WEB SERVICES

The `CompuTop.org` Software Archive of Dunfield [Dun] collects software for low-dimensional topology.

67.1.4 ALGORITHMS FOR FUNDAMENTAL GEOMETRIC OBJECTS

The computation of convex hulls and Delaunay triangulations/Voronoi diagrams is of key importance. For correct combinatorial output it is crucial to rely on arbitrary-precision arithmetic. On the other hand, some applications, e.g., volume computation, are content with floating-point arithmetic for approximate results. Some algorithmically more advanced but theoretically yet basic topics in this section are related to topology and real algebraic geometry.

In our terminology the *convex hull problem* asks for enumerating the facets of the convex hull of finitely many points in $\mathbb{R}^d$. The dual problem of enumerating the vertices and extremal rays of the intersection of finitely many halfspaces is equivalent by means of cone polarity. There is the related problem of deciding which points among a given set are extremal, that is, vertices of the convex hull. This can be solved by means of linear optimization.

STAND-ALONE SOFTWARE

Many convex hull algorithms are known, and there are several implementations. However, there is currently no algorithm for computing the convex hull in time polynomial in the combined input and output size, unless the dimension is con-

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

sidered constant. The behavior of each known algorithm depends greatly on the specific combinatorial properties of the polytope on which it is working. One way of summarizing the computational results from Avis, Bremner, and Seidel [ABS97] and Assarf et al. [AGH$^+$15] is: Essentially for each known algorithm there is a family of polytopes for which the given algorithm is superior to any other, and there is a second family for which the same algorithm is inferior to any other. For these families of polytopes we do have a theoretical, asymptotic analysis that explains the empirical results; see Chapter 26. Moreover, there are families of polytopes for which none of the existing algorithms performs well. Which algorithm or implementation works best for certain purposes will thus depend on the class of polytopes that is typical in those applications. For an overview of general convex hull codes see Table 67.1.1.[2] Note that `cddlib` and `lrslib` are both listed under exact arithmetic but can also be used with floating-point arithmetic.

Additionally, there are specialized codes: `azove` by Behle [Beh07] is designed to compute the vertices of a polytope with 0/1-coordinates from an inequality description by iteratively solving linear programs.

TABLE 67.1.1 Overview of convex hull codes.

Exact arithmetic		
PROGRAM	ALGORITHM	REMARKS
`beneath_beyond`	beneath-beyond [Ede87, 8.3.1]	Part of `polymake` [GJ$^+$16]
`cddlib` [Fuk07]	double description [JT13, 5.2]	
`lrslib` [Avi15]	reverse search [AF92]	
`normaliz` [BSSR16]	pyramid decomposition [BIS16]	
`porta` [CL09]	double description	
`ppl` [BHZ16]	double description	
Non-exact arithmetic		
PROGRAM	ALGORITHM	REMARKS
`cddf+` [Fuk07]	double description	
`hull` [Cla]	randomized incremental [CMS93]	Assumes input in gen. pos.; Exact computation unless Overflow signaled
`qhull` [BDH16]	quickhull [BDH96]	

The computation of Delaunay triangulations in d dimensions can be reduced to a $(d{+}1)$-dimensional convex hull problem; see Section 26.1. Thus, in principle, each of the convex hull implementations can be used to generate Voronoi diagrams. Additionally, however, some codes directly support Voronoi diagrams, notably Clarkson's `hull` [Cla], `qhull` by Barber, Dobkin, and Huhdanpaa [BDH16], and, among the programs with exact rational arithmetic, `lrs` by Avis [Avi15].

The following codes are specialized for 2-dimensional Voronoi diagrams: Shewchuk's `Triangle` [She05] and Fortune's `voronoi` [For01]. See also `cdt` by Lischinski [Lis98] for incremental constrained 2-dimensional Delaunay triangulation. For 3-dimensional problems there is `Detri` by Mücke [Müc95]. Delaunay triangulations

[2] We call an implementation *exact* if it, intentionally (but there may be programming errors, of course), gives correct results for *all* possible inputs. The nonexact convex hull codes use floating-point arithmetic or more advanced methods, but for each of them some input is known that makes them fail. The quality of the output of the nonexact convex hull codes varies considerably.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry,
J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

and, in particular, constrained Delaunay triangulations, play a significant role in meshing. Therefore, several of the Voronoi/Delaunay packages also have features for meshing and vice versa.

For the special case of triangulating a simple polygon (Chapter 30), Seidel's randomized algorithm has almost linear running time. The implementation by Narkhede and Manocha is part of the `Graphics Gems` [KHP$^+$13, Part V]. This archive and also Skiena's collection of algorithms [Ski01] contain more specialized code and algorithms for polygons.

Mesh generation is a vast area with numerous applications; see Chapter 29. This is reflected by the fact that there is an abundance of commercial and noncommercial implementations. We mention only a few. From the theoretical point of view the main categories are formed by 2-dimensional triangle meshes, 2-dimensional quadrilateral meshes, 3-dimensional tetrahedral meshes, 3-dimensional cubical (also called hexahedral) meshes, and other structured meshes. A focus on the applications leads to entirely different categories, which is completely ignored here. `Triangle` produces triangle meshes. `QMG` is a program for quadtree/octree 2- and 3-dimensional finite element meshing written by Vavasis [Vav00]. `Trelis` [tre14] can do many different variants of 2- and 3-dimensional meshing; it is a commercial product that is free for scientific use. Note that, depending on the context, triangle or tetrahedra meshes are also called triangulations.

In applications geometric objects are sometimes given as point clouds meant to represent a curve or surface. With the introduction of 3D-scanners and similar devices, appropriate techniques and related software became increasingly important. Obviously, this problem is directly related to mesh generation. `Cocone` by Dey et al. [DGG$^+$02] and `Power Crust` by Amenta, Choi, and Kolluri [ACK02] are designed to produce "watertight" surfaces; see Chapter 35. Geomagic is a company that sells many software products related (not only) to meshing [geo]. For instance, `Geomagic Wrap` generates meshes from 3D-scans.

`VisPak` by Wismath et al. [W$^+$02] is built on top of `LEDA` and can be used for the generation of visibility graphs of line segments and several kinds of polygons.

Smallest enclosing balls of a point set in arbitrary dimension can be computed with Gärtner's `Miniball` [Gär13].

Software at the junction between convexity and real algebraic geometry is still scarce. A noticeable exception is `Axel` by Mourrain and Wintz [MW]. It is designed for dealing with semi-algebraic sets in a way that is common in geometric modeling.

The computer algebra system `Magma` by Cannon et al. [C$^+$16] has some basic support for real algebraic geometry. Visualization of curves and surfaces can be done with `surf` by Endrass [End10].

LIBRARIES

Most of the above-mentioned software systems for dealing with polytopes and convex hulls are available as libraries of various flavors. `cddlib` [Fuk15] and `lrslib` [Avi15] are the C library versions of Fukuda's `cdd` and Avis's `lrs`, respectively. The `Parma Polyhedra Library` [BHZ16] and `Normaliz` [BSSR16] primarily *are* libraries, written in C++. `polymake` can be used as a C++ library, too. All of the above offer exact convex hull computation and exact linear optimization.

There is a C++ library version of `qhull` [BDH16] that performs convex hulls and Voronoi diagrams in floating-point arithmetic. Moreover, `cddlib` and `polymake` also have a limited support for floating-point arithmetic.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

The computation of Voronoi diagrams, arrangements, and related information is a particular strength of `CGAL` [F+16] and `LEDA` [led]. See Chapter 68.

The Quickhull algorithm (in three dimensions) is implemented as a Java class library [Llo04], and this has been wrapped for JavaScript in the `CindyJS` project [cin].

There is a parallelized C++ library `MinkSum` to enumerate the vertices of a Minkowski sum by Weibel [Wei12].

For triangle meshes in $\mathbb{R}^3$ there is the `GNU Triangulated Surface Library` [gts06] written in C. Its functionality comprises dynamic Delaunay and constrained Delaunay triangulations, robust set operations on surfaces, and surface refinement and coarsening for the control of level-of-detail.

Bhaniramka and Wenger have a set of C++ classes for the construction of isosurface patches in convex polytopes of arbitrary dimension [BW15]. These can be used in marching cubes like algorithms for isosurface construction.

GENERAL PURPOSE SYSTEMS AND THEIR MODULES

Plain `Maple` [map15] and `Mathematica` [mat15] only offer 2-dimensional convex hulls and Voronoi diagrams. Higher-dimensional convex hulls can be computed via the `Maple` package `convex` [Fra09]. Additionally, the `Matlab` package `GeoCalcLib` by Schaich provides an interface to `lrs` [Sch16].

Mitchell [Mit] has implemented some of his algorithms related to mesh generation in `Matlab` [mat16]. The finite element meshing program `QMG` by Vavasis can also be used with `Matlab`.

Particularly interesting for real algebraic geometry is `GloptiPoly` [HLL09], a `Matlab` package by Henrion et al., which, e.g., can find global optima of rational functions.

ADDITIONAL WEB SERVICES

Emiris maintains a Web page [Emi01] with several programs that address problems related to convex hull computations and applications in elimination theory. The web site `Polyhedral.info` [pol] was already mentioned.

`polymake` can be tried directly in the web browser from its web page [GJ+16]; it offers a front end to a server.

A Web page [Sch] by Schneiders contains a quite comprehensive survey on software related to meshing.

67.1.5 GEOMETRIC DATA STRUCTURES AND SEARCHING

LIBRARIES

Geometric data structures form the core of the C++ libraries `CGAL` [F+16] and `LEDA` [led]. The algorithms implemented include several different techniques for point location, collision detection, and range searching. See Chapter 68.

As already mentioned above, graph theory plays a role for some of the more advanced geometric algorithms. Several libraries for working with graphs have been developed over the years. It is important to mention in this context the `Boost Graph Library` [SLL15]. This is part of a general effort to provide free peer-reviewed portable C++ source libraries that extend the `STL`.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

67.1.6 COMPUTATIONAL TECHNIQUES

PARALLELIZATION

One important computational technique that is used in various contexts is parallelization. There are different levels at which this can be employed. At the very low level, modern processors can operate on large data vectors, up to 512 bits, in a single instruction, e.g., via the MMX, SSE, or AVX extensions to the x86 architecture.

At the next level the tasks of an algorithm can be split up and run in parallel on current multi-core machines. A widely used API for this approach is `OpenMP` [Ope15b], which provides preprocessor statements for easy (shared-memory) parallelization of C, C++, or Fortran code, e.g., parallel loop execution. The `OpenMP` framework can also be used to some extent on general-purpose graphics processing units (GPGPU).

Parallelization on graphics processing units can provide a large number of processing units, but those are rather limited in the instruction set and memory access. Common frameworks for this are NVIDIA's `CUDA` [Nvi15] or the `Open Computing Language (OpenCL)` [Ope15a].

A different approach to achieve massive parallelization is via the `Message Passing Interface (MPI)`, which is implemented for example in `OpenMPI` [Ope15c]. This allows us to spread the jobs to a large number of nodes connected via low-latency interconnect.

As an example, for the `lrs` code by Avis [Avi15] there is a wrapper `plrs` for shared-memory parallelization that works well up to about 16 threads. Moreover, there is also an `MPI`-based wrapper `mplrs` that can scale up to 1200 threads [AJ15]. Other libraries that also employ `OpenMP` are `Normaliz` [BSSR16] and `PHAT` [BKR].

67.1.7 APPLICATIONS

Applications of computational geometry are abundant and so are the related software systems. Here we list only very few items that may be of interest to a general audience.

STAND-ALONE SOFTWARE

For linear programming problems, essential choices for algorithms include simplex type algorithms or interior point methods. While commercial solvers tend to offer both, the freely available implementations seem to be restricted to either one. Additionally, there are implementations of a few special algorithms for low dimensions that belong to neither category. Altogether there are a large number of implementations, and we can only present a tiny subset here.

Exact rational linear programming can be done with `cdd` [Fuk07]. It uses either a dual simplex algorithm or the criss-cross method. An alternative exact linear programming code is `lrs` [Avi15], which implements a primal simplex algorithm.

`SoPlex` by Wunderling et al. [W$^+$16] implements the revised simplex algorithm both in primal and dual form; its most recent version supports exact rational arithmetic as well as floating-point arithmetic. It is part of the `SCIP Optimization`

Suite [sci16]. For an implementation of interior point methods see PCx by Czyzyk et al. [CMWW06]. Since the interior point method relies on numerical algorithms (e.g., Newton iterations) such implementations are always floating-point.

Glop is Google's linear solver, and it is available as open source [glo], but it can also be used via various Google services.

CPLEX [cpl15], Gurobi [gur], and XPress [xpr15] are widespread commercial solvers for linear, integer, and mixed integer programming. Each program offers a wide range of optimization algorithms. However, none of the commercial products can do exact rational linear optimization.

See also Hohmeyer's code linprog [Hoh96] for an implementation of Seidel's algorithm. This is based on randomization, and it takes expected linear time in fixed dimension; see Section 49.4.

There is an abundance of software for solid modeling. In the open source world Blender [ble16] seems to set the standard. For solid modeling with semi-algebraic sets, use Axel [MW].

Another topic with many applications is graph drawing. GraphViz [gra14] is an extensible package that offers tailor-made solutions for a wide range of applications in this area. An alternative is the Open Graph Drawing Framework [ogd15]. Tulip [AB$^+$16] specializes in the visualization of large graphs.

LIBRARIES

cddlib [Fuk15] and lrs offer C libraries for exact LP solving. CPLEX, Gurobi, OSL, PCx, and XPress can also be used as C libraries, while SoPlex has a C++ library version. Other free C libraries for linear and mixed integer programming include GLPK [Mak16] and lpsolve [Ber13].

GDToolkit [gdt07] is a C++ library for graph drawing, which is free for academic use.

In order to meet certain quality criteria, post-processing of mesh data is important. Varylab is a tool for the optimization of polygonal surfaces according to a variety of criteria [RSW].

GENERAL PURPOSE SYSTEMS AND THEIR MODULES

The linear optimization package PCx comes with an interface to Matlab [mat16].

ADDITIONAL WEB SERVICES

The Computational Infrastructure for Operations Research, or COIN-OR for short, is an open source project on optimization software that was initiated by IBM in 2000. Their web site contains lots of useful information and many links to (open source) software for various kinds of optimization [coi].

The geometrica project [B$^+$15] by Boissonnat et al. studies a variety of applications of computational geometry methods.

67.2 FEATURES OF SELECTED SOFTWARE SYSTEMS

All the software packages listed here have been mentioned previously. In many

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

cases, however, we list features not accounted for so far.

Axel [MW] is an algebraic geometric modeler for manipulation and computation with curves, surfaces or volumes described by semi-algebraic representations. These include parametric and implicit representations of geometric objects.

bliss [JK15] is a tool for computing automorphism groups and canonical forms of graphs. It has both a command line user interface as well as C++ and C programming language APIs.

cdd [Fuk07, Fuk15] is a convex hull code based on the double description method that is dual to Fourier-Motzkin elimination. It also implements a dual simplex algorithm and the criss-cross method for linear optimization. `cdd` comes as a stand-alone program; its C library version is called `cddlib`. The user can choose between exact rational arithmetic (based on the `GMP`) or floating-point arithmetic. Can be used via `SageMath`.

Cinderella [RGK16] is a commercial dynamic geometry software with a free version written in `Java`. It supports standard constructions with points, lines, quadrics and more (e.g., physics simulations). The recent `CindyJS` project [cin] aims at porting `Cinderella`'s functionality to JavaScript.

Cocone [DGG+02] is a set of programs related to the reconstruction of surfaces from point clouds in $\mathbb{R}^3$ via discrete approximation to the medial axis transform: `Tight Cocone` produces "watertight" surfaces from arbitrary input, while `Cocone/SuperCocone` is responsible for detecting the surface's boundary. `Geomview` output. Based on `CGAL` and `LEDA`. Not available for commercial use.

Computational Geometry in C [O'R98, O'R00] is a collection of C and `Java` programs including 2- and 3-dimensional convex hull codes, Delaunay triangulations, and segment intersection.

Geomview [Geo14] is a tool for interactive visualization. It can display objects in hyperbolic and spherical space as well as Euclidean space. `Geomview` comes with several external modules for specific visualization purposes. The user can write additional external modules in C. `Geomview` can be used as a visualization back end, e.g., for `Maple` [map15] and `Mathematica` [mat15]. The extension package `Maniview` can visualize 3-manifolds.

GloptiPoly [HLL09] can solve or approximate the Generalized Problem of Moments (GPM), an infinite-dimensional optimization problem that can be viewed as an extension of the classical problem of moments. This allows one to attack a wide range of optimization problems, including finding optima of multi-variate rational functions.

GraphViz [gra14] is a package with various graph layout tools. This includes hierarchical layouts and spring embedders. The system comes with a customizable graphical interface. Also runs on Windows.

GUDHI [BGJV] is a generic C++ library for computational topology and topological data analysis. Features include simplicial and cubical complexes, alpha complexes, and the computation of persistent (co-)homology.

hull [Cla] computes the convex hull of a point set in general position. The program can also compute Delaunay triangulations, alpha shapes, and volumes of Voronoi regions. The program uses exact machine floating-point arithmetic, and it signals overflow. `Geomview` output.

jReality [WGB+09] is a Java-based, full-featured 3D scene graph package designed for 3D visualization and specialized in mathematical visualization. It provides several backends, including a JOGL one for Java-based OpenGL rendering.

LattE integrale [BBDL+15] is software for counting and enumerating lattice points in polyhedra. Its most recent extension is a hybrid C++ and Maple implementation for computing the top coefficients of weighted Ehrhart quasipolynomials.

lrs [Avi15] is a convex hull code based on the reverse search algorithm due to Avis and Fukuda [AF92]. Exact rational, e.g. via the `GMP`, or floating-point arithmetic. In addition to convex hull computations, `lrs` can do linear optimization (via a primal simplex algorithm), volume computation, Voronoi diagrams, and triangulations. Moreover, `lrs` can, in advance, provide estimates for the running time and output size. It also comes as a C library and with wrappers for parallelization.

nauty [MP16] can compute a permutation group representation of the automorphism group of a given finite graph. As one interesting application this gives rise to an effective method for deciding whether two graphs are isomorphic or not. Such an isomorphism test can be performed directly.

Normaliz [BSSR16] is a tool for computations in affine monoids, vector configurations, lattice polytopes, and rational cones. Some algorithms are parallelized via the `OpenMP` protocol.

Parma Polyhedra Library (PPL) [BHZ16] provides numerical abstractions for applications to the analysis and verification of complex systems. Especially interesting for computational geometry are data types and algorithms for convex polyhedra, which may even be half-open. Fast implementation of the double description method for convex hulls. Can be used via `SageMath`.

PHAT [BKR] is a software library that provides methods for computing the persistence pairs of a filtered cell complex represented by an ordered boundary matrix with mod 2 coefficients. Uses `OpenMP` parallelization.

PolyLib [Loe10] is a library of polyhedral functions. Allows for basic geometric operations on parametrized polyhedra. As a key feature `PolyLib` can compute Ehrhart polynomials, which permits counting the number of integer points in a given polytope.

polymake [GJ+16] is a system for examining the geometrical and combinatorial properties of polytopes. It offers convex hull computation, standard constructions, and visualization. Some of the functionality relies—via interfaces—on external programs including `cdd`, `Geomview`, `jReality`, `lrs`, `nauty`, `PPL`, and `vinci`. `STL`-compatible C++ library; computations in exact rational arithmetic based on `GMP`. Separate modules for simplicial complexes (with homology computation and intersection forms of 4-manifolds), matroids, and other objects.

Regina [BBP+14] deals with manifolds, mostly in dimension 3, and its strength lies in normal surface theory. This way, e.g., it can be checked if a given 3-manifold satisfies the Haken property. Other features include homology and homotopy computations or Turaev–Viro invariants.

qhull [BDH16] computes convex hulls, Delaunay triangulations, Voronoi diagrams, farthest-site Delaunay triangulations, and farthest-site Voronoi diagrams. The algorithm implemented is Quickhull [BDH96]. `qhull` uses floating-point arithmetic only, but the authors incorporated several heuristics to improve the quality of the output. This is discussed in detail on a special Web page [BDH01]

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

in `qhull`'s documentation; it is an important source for everyone interested in using or implementing computational geometry software based on floating-point arithmetic.

SageMath [S+16] is a very comprehensive general mathematics software system, which builds on top of many existing open-source packages. These can be accessed through a common, Python-based language. Their mission statement reads as follows: "Creating a viable free open source alternative to `Magma`, `Maple`, `Mathematica` and `Matlab`."

Snappy [CDW] is a program for studying the topology and geometry of 3-manifolds, with a focus on hyperbolic structures. Can be used via `SageMath`.

SCIP [sci16] can do mixed integer programming (MIP) and mixed integer nonlinear programming (MINLP). Moreover, `SCIP` is also a framework for constraint integer programming and branch-cut-and-price. The larger `SCIP Optimization Suite` contains the `SoPlex` solver [W+16] for (exact and floating-point) linear programs and more.

TOPCOM [Ram14] is a package for examining point configurations via oriented matroids. The main purpose is to investigate the set of all triangulations of a given point configuration. Symmetric point configurations can be treated more efficiently if the user provides information about automorphisms. `TOPCOM` can check whether a given triangulation is regular.

Trelis [tre14] is a commercial meshing tool for surfaces and 3-dimensional objects to be used in finite element analysis. Mesh generation algorithms include quadrilateral and triangular paving, 2- and 3-dimensional mapping, hex sweeping and multi-sweeping, and others. This replaces/contains the former `CUBIT`.

vinci [BEF03] can be seen as an experimental framework for comparing volume computation algorithms. Exact and floating-point arithmetic. Implemented are Cohen & Hickey-triangulations [CH79], Delaunay triangulations (via `cdd` or `qhull`), and others.

REFERENCES

[AB+16] D. Auber, M. Bertrand, et al. `Tulip`, version 4.8.1. `http://tulip.labri.fr/`, 2016.

[ABS97] D. Avis, D. Bremner, and R. Seidel. How good are convex hull algorithms? *Comput. Geom.*, 7:265–301, 1997.

[ACK02] N. Amenta, S. Choi, and R.K. Kolluri. `Power Crust`, unions of balls, and the medial axis transform, version 1.2. `http://web.cs.ucdavis.edu/~amenta/powercrust.html`, 2002.

[AF92] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete Comput. Geom.*, 8:295–313, 1992.

[AGH+15] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. Computing convex hulls and counting integer points with `polymake`. Preprint, arXiv:1408.4653v2, 2015.

[AJ15] D. Avis and C. Jordan. `mplrs`: A scalable parallel vertex/facet enumeration code. Preprint, arXiv:1511.06487, 2015.

[Avi15] D. Avis. `lrs`, `lrslib`, version 6.1. `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`, 2015.

[B+15] J.-D. Boissonnat et al. `Geometrica` — geometric computing. `https://team.inria.fr/geometrica/`, 2015.

[BBDL+15] V. Baldoni, N. Berline, J.A. De Loera, B. Dutra, M. Köppe, S. Moreinis, G. Pinto, M. Vergne, and J. Wu. A user's guide for `LattE integrale` v1.7.3. `http://www.math.ucdavis.edu/~latte/`, 2015.

[BBP+14] B.A. Burton, R. Budney, W. Pettersson, et al. `Regina`: Software for 3-manifold topology and normal surface theory. `http://regina.sourceforge.net/`, 2014.

[BDH96] C.B. Barber, D.P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Software*, 22:469–483, 1996.

[BDH01] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. Imprecision in `qhull`. `http://www.qhull.org/html/qh-impre.htm`, 2001.

[BDH16] C.B. Barber, D.P. Dobkin, and H.T. Huhdanpaa. `qhull`, version 2015.2. `http://www.qhull.org/`, 2016.

[BEF03] B. Büeler, A. Enge, and K. Fukuda. `vinci`, version 1.0.5. `http://www.math.u-bordeaux1.fr/~aenge/index.php?category=software&page=vinci`, 2003.

[Beh07] M. Behle. `azove` — another zero one vertex enumeration tool. `https://people.mpi-inf.mpg.de/~behle/azove.html`, 2007.

[Ber13] M. Berkelaar. `lpsolve`, version 5.5. `http://lpsolve.sourceforge.net/`, 2013.

[BG09] W. Bruns and J. Gubeladze. *Polytopes, Rings, and K-theory.* Springer Monographs in Mathematics, Springer, Dordrecht, 2009.

[BGJV] J.-D. Boissonnat, M. Glisse, C. Jamin, and R. Vincent. `GUDHI` (Geometric Understanding in Higher Dimensions). `http://gudhi.gforge.inria.fr/`.

[BHZ16] R. Bagnara, P.M. Hill, and E. Zaffanella. The Parma Polyhedra Library, version 1.2. `http://bugseng.com/products/ppl/`, 2016.

[BIS16] W. Bruns, B. Ichim, and C. Söger. The power of pyramid decomposition in Normaliz. *J. Symbolic Comput.*, 74:513–536, 2016.

[BKR] U. Bauer, M. Kerber, and J. Reininghaus. `PHAT` (Persistent Homology Algorithm Toolbox), version 1.4.0. `https://bitbucket.org/phat-code/phat`.

[ble16] `Blender` v2.77a. `https://www.blender.org/`, 2016.

[BR07] M. Beck and S. Robins. *Computing the Continuous Discretely. Integer-Point Enumeration in Polyhedra.* Undergraduate Texts in Mathematics, Springer, New York, 2007.

[BSSR16] W. Bruns, R. Sieg, C. Söger, and T. Römer. `Normaliz`. Algorithms for rational cones and affine monoids, version 3.1.1. `http://www.math.uos.de/normaliz`, 2016.

[BW15] P. Bhaniramka and R. Wenger. `IJK`: Isosurface jeneration kode. `http://web.cse.ohio-state.edu/research/graphics/isotable`, 2015.

[C+16] J. Cannon et al. `Magma`, version 2.21-12. `http://magma.maths.usyd.edu.au/magma`, 2016.

[Cab07] Cabrilog. `Cabri 3D`. `http://www.cabri.com/`, 2007.

[CDW] M. Culler, N.M. Dunfield, and J.R. Weeks. `SnapPy`, a computer program for studying the topology of 3-manifolds. `http://snappy.computop.org`.

[CH79] J. Cohen and T. Hickey. Two algorithms for determining volumes of convex polyhedra. *J. Assoc. Comput. Mach.*, 26:401–414, 1979.

[cho] Computational Homology Project. http://chomp.rutgers.edu.

[cin] CindyJS. http://cindyjs.org.

[CL09] T. Christof and A. Löbel. Porta, version 1.4.1. http://porta.zib.de/, 2009.

[Cla] K.L. Clarkson. hull, Version 1.0. http://www.netlib.org/voronoi/hull.html.

[CMS93] K.L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom.*, 3:185–212, 1993.

[CMWW06] J. Czyzyk, S. Mehrotra, M. Wagner, and S. Wright. PCx, version 1.1. http://pages.cs.wisc.edu/~swright/PCx/, 2006.

[coi] COmputational INfrastructure for Operations Research. International Business Machines, http://www.coin-or.org/.

[cpl15] IBM ILOG CPLEX Optimization Studio, Version 12.6.2. IBM Corporation, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, 2015.

[dgd] DGD gallery. http://gallery.discretization.de.

[DGG+02] T. Dey, J. Giesen, S. Goswami, J. Hudson, and W. Zhao. Cocone softwares. http://web.cse.ohio-state.edu/~tamaldey/cocone.html, 2002.

[DHSW03] J.-G. Dumas, F. Heckenbach, D. Saunders, and V. Welker. Computing simplicial homology based on efficient Smith Normal Form algorithms. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 177–206, Springer, Berlin, 2003.

[doc] docker. http://docker.com.

[Dun] N.M. Dunfield. The CompuTop.org Software Archive. http://computop.org/.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Berlin, 1987.

[Emi01] I.Z. Emiris. Computational geometry. http://www-sop.inria.fr/galaad/logiciels/emiris/soft_geo.html, 2001.

[End10] S. Endrass. surf, version 1.0.6. http://surf.sourceforge.net, 2010.

[Epp03] D. Eppstein. Polyominoes and other animals. http://www.ics.uci.edu/~eppstein/junkyard/polyomino.html, 2003.

[Epp12] D. Eppstein. The Geometry Junkyard. http://www.ics.uci.edu/~eppstein/junkyard, 2012.

[ES16] F. Effenberger and J. Spreer. simpcomp — a gap package, version 2.1.6. https://github.com/simpcomp-team/simpcomp, 2016.

[F+16] A. Fabri et al. CGAL, version 4.8. http://www.cgal.org, 2016.

[For01] S.J. Fortune. voronoi. http://cm.bell-labs.com/who/sjf, 2001.

[Fra09] M. Franz. convex — a maple package for convex geometry, version 1.1.3. http://www-home.math.uwo.ca/~mfranz/convex/, 2009.

[Fuk07] K. Fukuda. cdd+, version 0.77a. http://www.inf.ethz.ch/personal/fukudak/cdd_home/, 2007.

[Fuk15] K. Fukuda. cddlib, version 0.94h. http://www.inf.ethz.ch/personal/fukudak/cdd_home/, 2015.

[Gär13] B. Gärtner. Miniball, version 3.0. http://www.inf.ethz.ch/personal/gaertner/miniball.html, 2013.

[gdt07] GDToolkit, version 4.0. Dipartimento di Informatica e Automazione, Università di Roma Tre, Rome, Italy, http://www.dia.uniroma3.it/~gdt, 2007.

[geo] Geomagic. http://www.geomagic.com/.

[Geo13] The Geometer's Sketchpad. McGraw-Hill Education, http://www.keycurriculum.com/, 2013.

[Geo14] Geomview, version 1.9.5. The Geometry Center, http://www.geomview.org, 2014.

[Geo15] GeoGebra. International GeoGebra Institute, http://www.geogebra.org, 2015.

[GJ+16] E. Gawrilow, M. Joswig, et al. polymake, Version 3.0. http://polymake.org, 2016.

[glo] Glop. https://developers.google.com/optimization/lp.

[gmp15] GNU multiprecision arithmetic library, Version 6.1.0. http://www.gmplib.org, 2015.

[gra14] GraphViz, version 2.38. AT&T Lab – Research, http://www.graphviz.org/, 2014.

[gts06] GNU triangulated surface library, version 0.7.6. http://gts.sourceforge.net, 2006.

[gur] Gurobi optimizer v6.5. http://www.gurobi.com.

[HLL09] D. Henrion, J.-B. Lasserre, and J. Löfberg. GloptiPoly 3: moments, optimization and semidefinite programming. *Optim. Methods Softw.*, 24:761–779, 2009.

[Hoh96] M. Hohmeyer. linprog. http://www3.cs.stonybrook.edu/~algorith/implement/linprog/implement.shtml, 1996.

[ima] Imaginary – open mathematics. http://imaginary.org.

[isl16] Integer set library, version 0.17.1. http://isl.gforge.inria.fr/, 2016.

[Jen] A.N. Jensen. Gfan 0.5, a software system for Gröbner fans and tropical varieties. http://home.imf.au.dk/jensen/software/gfan/gfan.html.

[JK15] T. Junttila and P. Kaski. bliss: A tool for computing automorphism groups and canonical labelings of graphs, version 0.73. http://www.tcs.hut.fi/Software/bliss/, 2015.

[JT13] M. Joswig and T. Theobald. *Polyhedral and algebraic methods in computational geometry.* Universitext, Springer, London, 2013.

[jul16] JuliaGeometry — computational geometry with Julia. https://github.com/JuliaGeometry, 2016.

[KHP+13] D. Kirk, P. Heckbert, A. Paeth, et al. Graphics gems. http://www.graphicsgems.org/, 2013.

[KP02] V. Kaibel and M.E. Pfetsch. Computing the face lattice of a polytope from its vertex-facet incidences. *Comput. Geom.*, 23:281–290, 2002.

[led] LEDA, version 6.3. Algorithmic Solution Software GmbH, http://www.algorithmic-solutions.com/enleda.htm.

[Lis98] D. Lischinski. cdt. http://www.cs.huji.ac.il/~danix/code/cdt.tar.gz, 1998.

[Llo04] J.E. Lloyd. Class quickhull3d. https://quickhull3d.github.io/quickhull3d/apidocs/com/github/quickhull3d/QuickHull3D.html, 2004.

[Loe10] V. Loechner. PolyLib — a library of polyhedral functions, version 5.22.5. http://icps.u-strasbg.fr/PolyLib, 2010.

[Mak16] A. Makhorin. GNU linear programming kit, version 4.60. http://www.gnu.org/software/glpk/glpk.html, 2016.

[map15] Maple 2015. Waterloo Maple, Inc., http://www.maplesoft.com/products/maple/, 2015.

[mat15] Mathematica, version 10. Wolfram Research, Inc., http://www.wolfram.com/mathematica/, 2015.

[mat16] Matlab, R2016a. The Mathworks, Inc., http://www.mathworks.com/products/matlab, 2016.

[Mit] S.A. Mitchell. Computational geometry triangulation results. http://www.sandia.gov/~samitch/csstuff/csguide.html.

[Mor] D. Morozov. Dionysus. http://www.mrzv.org/software/dionysus/.

[MP16] B. McKay and A. Piperno. nauty, version 2.6r5. http://pallini.di.uniroma1.it/, 2016.

[mpf16] GNU multiple precision floating-point reliably, version 3.1.4. http://www.mpfr.org, 2016.

[mpi15] Multiple Precision Integers and Rationals, version 2.7.2. http://www.mpir.org, 2015.

[Müc95] E.P. Mücke. Detri, version 2.6a. http://www.geom.uiuc.edu/software/cglist/GeomDir, 1995.

[MW] B. Mourrain and J. Wintz. Axel – algebraic geometric modeling. http://axel.inria.fr.

[Nvi15] Nvidia Corporation. CUDA. http://www.nvidia.com/object/cuda_home.html, 2015.

[ogd15] OGDF: Open graph drawing framework. http://www.ogdf.net, 2015.

[Ope15a] Open compute language (OpenCL). Khronos Group, http://www.khronos.org/opencl/, 2015.

[Ope15b] Open multi-processing (OpenMP). OpenMP Architecture Review Board, http://www.openmp.org/, 2015.

[Ope15c] Open MPI. http://www.open-mpi.org/, 2015.

[O'R98] J. O'Rourke. *Computational Geometry in C*, second edition. Cambridge University Press, Cambridge, 1998.

[O'R00] J. O'Rourke. Computational geometry in C. http://cs.smith.edu/~orourke/books/ftp.html, 2000.

[pol] Polyhedral.info. http://polyhedral.info/software.

[PR03] J. Pfeifle and J. Rambau. Computing triangulations using oriented matroids. In M. Joswig and N. Takayama, editors, *Algebra, Geometry, and Software Systems*, pages 49–75, Springer, Berlin, 2003.

[Ram14] J. Rambau. TOPCOM, version 0.17.5. http://www.rambau.wm.uni-bayreuth.de/TOPCOM, 2014.

[RGK16] J. Richter-Gebert and U.H. Kortenkamp. Cinderella, version 2.9. http://www.cinderella.de, 2016.

[RSW] T. Roerig, S. Sechelmann, and M. Wahren. VaryLab: discrete surface optimization. http://www.varylab.com.

[S$^+$16] W.A. Stein et al. Sage Mathematics Software, version 7.1. The Sage Development Team, http://www.sagemath.org, 2016.

[Sch] R. Schneiders. Software: list of public domain and commercial mesh generators. http://www.robertschneiders.de/meshgeneration/software.html.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.

[Sch16] R. Schaich. `GeoCalcLib`. `http://worc4021.github.io/`, 2016.

[sci16] `SCIP` optimization suite 3.2.1. `http://scip.zib.de/`, 2016.

[She05] J.R. Shewchuk. `Triangle`, version 1.6. `http://www.cs.cmu.edu/~quake/triangle.html`, 2005.

[Ski01] S.S. Skiena. The Stony Brook Algorithm Repository. `http://www3.cs.stonybrook.edu/~algorith/index.html`, 2001.

[SLL15] J. Siek, L.-Q. Lee, and A. Lumsdaine. The `boost graph library (bgl)`, version 1.60.0. `http://www.boost.org/libs/graph/doc/index.html`, 2015.

[Ste09] K. Stephenson. `CirclePack`, version 2.0. `http://www.math.utk.edu/~kens/CirclePack`, 2009.

[tre14] `Trelis` advanced meshing software. `http://www.csimsoft.com/trelis.jsp`, 2014.

[TVJA14] A. Tausz, M. Vejdemo-Johansson, and H. Adams. `JavaPlex`: A research software package for persistent (co)homology. In *Proc. 4th Int. Cong. Math. Software*, vol. 8592 of *LNCS*, pages 129–136, Springer, Berlin, 2014.

[Vav00] S.A. Vavasis. `QMG`, version 2.0, patch 2. `http://www.cs.cornell.edu/Info/People/vavasis/qmg-home.html`, 2000.

[Ver16] S. Verdoolaege. `barvinok`, version 0.39. `http://barvinok.gforge.inria.fr/`, 2016.

[W+02] S.K. Wismath et al. `VisPak`, version 2.0. `http://www.cs.uleth.ca/~wismath/vis.html`, 2002.

[W+16] R. Wunderling et al. The sequential object-oriented simplex class library, version 2.2.1. `http://soplex.zib.de/`, 2016.

[Wee00] J. Weeks. `SnapPea`, version 3.0d3. `http://www.geometrygames.org/SnapPea`, 2000.

[Wei12] C. Weibel. `Minksum`, version 1.7. `https://sites.google.com/site/christopheweibel/research/minksum`, 2012.

[WGB+09] S. Weißmann, C. Gunn, P. Brinkmann, T. Hoffmann, and U. Pinkall. `jReality`: A java library for real-time interactive 3d graphics and audio. In *Proc. 17th ACM Internat. Conf. Multimedia*, pages 927–928, 2009.

[Wol16] `WolframAlpha`. Wolfram Alpha LLC, `http://www.wolframalpha.com/`, 2016.

[xpr15] FICO `xpress` optimization suite. FICO, `http://www.fico.com/en/products/fico-xpress-optimization-suite`, 2015.

[YD10] C. Yap and Z. Du. `Core Library`, version 2.1. `http://cs.nyu.edu/exact/core_pages`, 2010.

Preliminary version (August 10, 2017). To appear in the Handbook of Discrete and Computational Geometry, J.E. Goodman, J. O'Rourke, and C. D. Tóth (editors), 3rd edition, CRC Press, Boca Raton, FL, 2017.