Convex Hulls, Oracles, and Homology ⋆

Michael Joswig [1]

Fakultät für Mathematik, Institut für Algebra und Geometrie, Otto-von-Guericke-Universität Magdeburg, D-39106 Magdeburg, Germany, joswig@math.tu-berlin.de

Günter M. Ziegler [1]

Mathematisches Institut, MA 6-2, Technische Universität Berlin, D-10623 Berlin, Germany, ziegler@math.tu-berlin.de

Abstract

This paper presents a new algorithm for the convex hull problem, which is based on a reduction to a combinatorial decision problem CompletenessC, which in turn can be solved by a simplicial homology computation. Like other convex hull algorithms, our algorithm is polynomial (in the size of input plus output) for simplicial or simple input. We show that the "no"-case of CompletenessC has a certificate that can be checked in polynomial time (if integrity of the input is guaranteed).

1 Introduction

Every convex polytope $P \subset \mathbb{R}^d$ can be described as the convex hull of a finite set $\mathcal{P}$ of points or as the (bounded) set of solutions of a finite system $\mathcal{H}$ of linear equations and inequalities [Ziegler(1995), Lect. 1]. In view of the fundamental role that polytopes play in Euclidean geometry and hence for any type of geometric computing, the conversion between the two types of representations, known as the *convex hull problem*, is of key interest. It splits into two separate tasks.

⋆ An extended abstract version of this paper, "Polytope verification by homology computation," has appeared in the Proceedings of EuroCG, Berlin, March 26–28, 2001, pp. 142–145.

[1] Partially supported by Deutsche Forschungs-Gemeinschaft (DFG). In particular, both authors are members of the DFG Research Center FZT86.

The first task is the *facet enumeration problem*: Given a finite set of points $\mathcal{P} \subset \mathbb{R}^d$, determine the combinatorial structure of its boundary. For this one does not want to explicitly enumerate all the faces (the intersections of P with supporting hyperplanes), but one wants sparser date, namely to compute a minimal representation of the convex hull $\operatorname{conv}(\mathcal{P})$ in terms of equations and (facet-defining) inequalities. Here the equations should describe the affine hull $\operatorname{aff}(P)$, while the additional inequalities correspond to the *facets* (faces of codimension 1) of P. If P is full-dimensional in $\mathbb{R}^d$, then the facet-defining inequalities are unique up to scaling.

The second task is the *vertex enumeration problem*: Given a finite system $\mathcal{H}$ of linear (equations and) inequalities, and provided that the set of solutions $P = \bigcap \mathcal{H}$ is bounded, compute the minimal set of points $\mathcal{P}$ whose convex hull is P. This minimal set is unique; it consists of the *vertices* (0-dimensional faces) of P.

The two tasks are dual to each other, via cone polarity. Thus if an LP-type oracle (an algorithm which for a system of inequalities computes a solution, or for a set of points computes a separating hyperplane, cf. [Grötschel et al.(1993)]) is available, every algorithm for the facet enumeration problem can also be used for vertex enumeration, and vice versa.

The use of "oracles" is essential in the following. In our exposition, we attempt to separate geometric from combinatorial data/algorithms, and thus to separate the computational complexity of various sub-problems from the main task. This is achieved by delegating the subproblems in terms of oracles (in the sense of [Lovász(1986)] and [Grötschel et al.(1993)]). The complexity model adopted is that oracle calls cost only one unit, independent of the actual complexity of the subproblem or of the algorithm used to solve it (that is, to implement the subproblem). Thus, for example, the arithmetics of real numbers is delegated to an oracle (as in [Lovász(1986), Part 1]), and calls to such an oracle are counted as "the number of arithmetic operations." Similarly, LP-computations via an oracle cost one unit each, independent of the still-not-resolved complexity status of linear programming. (Here polynomial algorithms such as the ellipsoid method are available in the bit/Turing machine model [Grötschel et al.(1993)], but there no provably strongly-polynomial algorithm yet, whose running time would be bounded by a polynomial of number of arithmetic operations.) Thus, at the *Symbolic Computation* level of "an LP-oracle available" the polynomial algorithms derived below are indeed strongly-polynomial algorithms.

Despite the great interest in the convex hull problem, and despite the fact that a number of different strategies and algorithms have been explored, implemented and analyzed in detail (see the web page [Fukuda(2000)], as well as [Avis(2001)], [Avis(2000)], [Fukuda(2003)], [Gawrilow and Joswig(1997–2003)], and [Gawrilow and Joswig(2001)] for implementations), the problem can be considered "solved" neither in theory, nor in practice. If the dimension d is fixed, Chazelle's celebrated algorithm [Chazelle(1993)] gives an asymptotically worst-case optimal (polynomial time) theoretical solution. Its optimality is based on McMullen's "Upper Bound Theorem" [McMullen(1970)] on the maximal number of facets for a d-polytope with n vertices. However, for any given convex hull problem, the output may be small, but it may also be much larger than the input — indeed, it may be of exponential size, if the dimension is not fixed. This is very relevant, since high-dimensional computations occur in a variety of important applications. Thus one is asking for a convex hull algorithm whose running time is bounded by a polynomial in the size of "input plus output." Such an algorithm would be called *output-sensitive*. The analysis by Avis, Bremner and Seidel [Avis et al.(1997)] shows that, unfortunately, none of the known types of convex hull algorithms is output-sensitive. These can roughly be categorized as follows: Incremental and triangulation producing (e.g., Chazelle's method), incremental without triangulations (e.g., Fourier-Motzkin elimination [Ziegler(1995), Lect. 1]), non-incremental (e.g., reverse search [Avis and Fukuda(1992)]). Note that, by a result of Bremner [Bremner(1999)], only non-incremental methods can possibly be output-sensitive.

The purpose of this paper is to describe a new (non-incremental) convex hull algorithm, based on a completely different principle. To this end, we first present a (folklore) polynomial reduction of FacetEnumeration to the decision problem PolytopeVerification. Then we further reduce to the Completeness problem: Is a given description of a d-polytope by *some* of its vertices and *some* of its facets complete, that is, are we given *all* the vertices and *all* the facets? Looking at the convex hull problem via its reduction to PolytopeVerification or Completeness automatically reveals its inherent self-dual structure. It is an interesting feature that the Completeness problem can be posed both with geometric input data and as an entirely combinatorial problem CompletenessC, where only the incidences between vertices and facets are given.

Let us just mention here one recent occurrence of the combinatorial completeness problem: [McCarthy et al.(2002)] describes a situation where one wants to know whether a given inequality description for a polytope is complete. Moreover, the vertex coordinates in some of their problems are necessarily non-rational, so any coordinate-free/combinatorial approach is welcome. Unfortunately, the most interesting case left "open" by McCarthy et al. (the convex hull of the matrices corresponding to the Coxeter group H_4) is a polytope completeness problem in dimension $d = 16$ with 14,400 vertices: From this data our method generates gigantic boundary matrices that are plainly too large to process.

We have further been informed by Samuel Fiorini (email, January 2002) that he has successfully used a certificate for the "no"-case of CompletenessC that is similar to the one that we describe in Section 6.

Our main contribution is an algorithm to attack the combinatorial CompletenessC problem via deciding whether a certain simplicial homology group of a certain abstract simplicial complex vanishes or not. Moreover, we present a polynomially checkable certificate for non-completeness, provided that the input is valid. For the geometric version the validity of the input can be checked easily. Unfortunately, the complexity status for the homology computation problem is open. The best currently available strategy to decide nontriviality of the homology group in question seems to be to compute boundary matrices and perform Gaussian elimination. Since the boundary matrices in our algorithm can be exponentially large, we do not obtain an output-sensitive method. However, like other methods (e.g., Avis and Fukuda's reverse search [Avis and Fukuda(1992)] or Seidel's gift-wrapping algorithm [Seidel(1991)]) our algorithm is output-sensitive in the case of simplicial polytopes.

2 FacetEnumeration via PolytopeVerification

We start with a more formal description of the facet enumeration problem:

Problem 1 FacetEnumeration$(e, \mathcal{P})$*:*
Input: integer $e \geq 0$*; finite set of points* $\mathcal{P} \subset \mathbb{R}^e$*.*
Output: minimal description of $\operatorname{conv}(\mathcal{P})$ *in terms of equations (for the affine hull of* $\mathcal{P}$*) and inequalities (one for each facet of* $\operatorname{conv}(\mathcal{P})$*)*

It is well-known, see [Avis et al.(1997)], [Fukuda(2000), Node 21], and [Kaibel and Pfetsch(2003), Problems 1–3], that FacetEnumeration has a polynomial reduction to the polytope verification problem:

Problem 2 PolytopeVerification$(e, \mathcal{P}, \mathcal{H})$*:*
Input: integer $e \geq 0$*; finite set of points* $\mathcal{P} \subset \mathbb{R}^e$*; finite set* $\mathcal{H}$ *of closed half-spaces in* $\mathbb{R}^e$
Output: answer ***yes/no*** *to the question whether* $\operatorname{conv}(\mathcal{P}) = \bigcap \mathcal{H}$

Freund and Orlin have shown that a related problem, to decide whether $\bigcap \mathcal{H} \subseteq \operatorname{conv}(\mathcal{P})$, is co-NP-complete [Freund and Orlin(1985)].

3 PolytopeVerification via CompletenessG

Assuming that an LP-type oracle is available, the PolytopeVerification problem is polynomially equivalent to the following *geometric polytope completeness problem*:

Problem 3 CompletenessG$(d, \mathcal{V}, \mathcal{F})$*:*
Input: integer $d \geq 0$; finite set of points $\mathcal{V} \subset \mathbb{R}^d$; finite set $\mathcal{F}$ of closed halfspaces in $\mathbb{R}^d$, such that
- $P := \mathrm{conv}(\mathcal{V})$ *is contained in* $Q := \bigcap \mathcal{F}$
- $\dim P = \dim Q = d$
- *every $v \in \mathcal{V}$ defines a vertex of Q*
- *every $F \in \mathcal{F}$ defines a facet of P*

*Output: answer **yes/no** to the question whether $P = Q$*

As in the case of PolytopeVerification, the roles of vertices and facets are interchangeable for CompletenessG.

We sketch the reduction of PolytopeVerification to CompletenessG. Given any input $(e, \mathcal{P}, \mathcal{H})$ for PolytopeVerification, set $P := \mathrm{conv}(\mathcal{P})$ and $Q := \bigcap \mathcal{H}$. Employ Gaussian elimination to determine $\dim P$. Verify whether all the inequalities in $\mathcal{H}$ are valid for P; if this is not the case, then $P \not\subseteq Q$, so we output **no**; otherwise $P \subseteq Q$ is established. Now extract the set $\mathcal{H}'$ of all halfspaces from $\mathcal{H}$ for which P lies in the bounding hyperplane, that is, all those inequalities which are tight on $\mathrm{aff}\, P$. An LP-type oracle is sufficient to check whether $\bigcap \mathcal{H}' = \mathrm{aff}\, P$; if this is not the case, then we know that $\dim Q > \dim P$, so we can output **no**. Otherwise we proceed by restricting the input to $\mathrm{aff}\, P$, that is, we deal with the situation where P is full-dimensional.

Now remove from $\mathcal{H}$ all the halfspaces which do not determine facets of P; this may be done using Gaussian elimination. (In the case $P = Q$, this removal does not change Q; in the case $P \subset Q$, it may enlarge Q.) Similarly, we now remove from $\mathcal{P}$ all those points which do not arise as intersections of some bounding hyperplanes of halfspaces in $\mathcal{H}$; again this may be done via Gaussian elimination. (In the case of $P = Q$, this removal does not change P; in the case $P \subset Q$, we may lose vertices of P, thus making P smaller.)

Now we have prepared our input for CompletenessG. Indeed, the first two conditions on the input are satisfied, the other two are easily checked: If one of them fails, then output the answer **no**. $\square$

Clearly, an LP-type oracle cannot be avoided in the reduction: The instance CompletenessG$(d, \emptyset, \mathcal{F})$ asks to decide whether $\bigcap \mathcal{F} = \emptyset$. This is known to be strongly polynomially equivalent to finding an optimal solution of a linear program, cf. [Grötschel et al.(1993)].

4 CompletenessG via CompletenessC

The *incidence matrix* of a polytope P with vertex set $\mathcal{V}$ and facet set $\mathcal{F}$ is defined to be the matrix

$$I_P \quad := \quad \left(i_{Fv}\right)_{F\in\mathcal{F},v\in\mathcal{V}} \quad \in \quad \{0,1\}^{\mathcal{F}\times\mathcal{V}},$$

where $i_{Fv} = 1$ if vertex v lies on the facet F (that is, if $v \in F$), and $i_{Fv} = 0$ means that $v \notin F$. This matrix is well-defined up to permutation of rows and of columns, which corresponds to reordering $\mathcal{V}$ and $\mathcal{F}$. A *minor* of a matrix will refer to any submatrix obtained by possibly removing rows and/or columns. A minor J of the incidence matrix I_P is *complete* if $J = I_P$. Thus we arrive at the *combinatorial polytope completeness problem*:

Problem 4 COMPLETENESSC(d, J)*:*
Input: integer $d \geq 0$*; incidence matrix minor* J *of a* d*-polytope*
Output: answer **yes/no** *to the question whether* J *is complete*

It is not obvious that this problem is well defined. However, from Theorem 5 below it follows that there are no two d-polytopes P and P' such that a 0/1-matrix J is both a complete incidence matrix for P and an incomplete minor of an incidence matrix for P'. (See also the related discussion in [Joswig et al.(2001)].) It is clear that COMPLETENESSG has a polynomial reduction to COMPLETENESSC.

It is essential to have the dimension among the input parameters of COMPLETENESSC. This is demonstrated by the following example [Ziegler(1995), p. 71]:

$$J_{\mathrm{KM}} \quad = \quad \begin{pmatrix} 1&1&1&1&0&0&0&0\\ 1&1&0&0&0&0&1&1\\ 1&0&0&1&1&0&0&1\\ 0&1&1&0&0&1&1&0\\ 0&0&1&1&1&1&0&0\\ 0&0&0&0&1&1&1&1 \end{pmatrix}$$

We can identify $\mathcal{V} = \{1, 2, \ldots, 8\}$ and $\mathcal{F} = \{1234, 1278, 1458, 2367, 3456, 5678\}$ with the sets of vertices and facets, respectively, of a 3-dimensional cube (in a suitable "Klee-Minty" vertex numbering; see Figure 1(b) below). Consequently, COMPLETENESSC$(3, J_{\mathrm{KM}})$ = **yes**. But we can also identify $\mathcal{V}$ with the vertices of a cyclic 4-polytope $C_4(8)$. Then each element in $\mathcal{F}$ corresponds to a facet of $C_4(8)$, according to Gale's evenness criterion. Hence COMPLETENESSC$(4, J_{\mathrm{KM}})$ = **no**, since $C_4(8)$ has 20 facets.

A more generic class of examples for which the dimension information is needed arises from the prism construction: Let P be an arbitrary d-polytope and $P' = P \times [0, 1]$ the prism over P. The facets of P' are $P \times \{0\}$, $P \times \{1\}$, and the products of facets of P with the interval $[0,1]$. Call the latter facets of P' *vertical*, and let J_P be an incidence matrix of P. We have $\textsc{Completeness}\mathrm{C}(d, J_P) = \mathbf{yes}$. On the other hand J_P is also a minor of an incidence matrix of P', which corresponds to the vertical facets and, say, the vertices in the bottom facet $P \times \{0\}$. Therefore, $\textsc{Completeness}\mathrm{C}(d+1, J_P) = \mathbf{no}$.

5 CompletenessC via simplicial homology

We will point out that $\textsc{Completeness}\mathrm{C}$ has a topological core. The reader is referred to [Björner(1995)] for a survey of topological combinatorics tools, and to the appendix for a brief introduction to simplicial homology. In the following we will use reduced simplicial homology with coefficients in $\mathbb{Z}_2$. One could use any other commutative coefficient ring with unit, but $\mathbb{Z}_2$ is the natural choice in terms of efficiency and simplicity.

Let $J \in \{0,1\}^{\mathcal{F}\times\mathcal{V}}$ be an incidence matrix minor of some polytope P with vertex set $\mathcal{V}' \supseteq \mathcal{V}$ and facet set $\mathcal{F}' \supseteq \mathcal{F}$. Thus the columns of J are in bijection with a (partial) vertex set $\mathcal{V}$ of P. Each row of J is the characteristic vector of a subset of columns, i.e., of a subset of $\mathcal{V}$. Thus in the following we interpret J as a combinatorial encoding of a system $\mathcal{F}$ of (not necessarily distinct) subsets of $\mathcal{V}$, and with slight abuse of notation we write $\mathcal{F} \subseteq 2^{\mathcal{V}}$. The *crosscut complex* of J is the simplicial complex

$$\Gamma(J) \quad := \quad \Big(\mathcal{V},\ \bigcup\big\{2^F : F \in \mathcal{F}\big\}\Big),$$

the simplicial complex of all sets of vertices that are contained in *some* facet in $\mathcal{F}$. If P is simplicial and J is complete, then the crosscut complex coincides with the boundary complex of P.

Before we state and prove our main result we shall discuss the small cases, where $d \leq 2$, directly. A 1-dimensional polytope is a line segment $[v,w]$ with two vertices v and w, which happen to be also the facets. Its boundary is $\mathbb{S}^0 = \{v,w\}$, and we have $\widetilde{H}_0(\mathbb{S}^0;\mathbb{Z}_2) \cong \mathbb{Z}_2$. A proper partial 1-polytope has one vertex and one facet; its crosscut complex is a single point, and the reduced homology in dimension 0 vanishes. A 2-dimensional polytope is an n-gon; its boundary is the n-cycle, homeomorphic to $\mathbb{S}^1$, and we have $\widetilde{H}_1(\mathbb{S}^1;\mathbb{Z}_2) \cong \mathbb{Z}_2$. A proper partial 2-polytope is the disjoint union of edge paths, each of which is contractible. Hence the first homology of a proper partial 2-polytope vanishes. The same reasoning applies if we replace $\mathbb{Z}_2$ by any other coefficient ring.

For an example of the crosscut complex of a partial 3-polytope see Figure 1(a).

Theorem 5 *The incidence matrix minor $J \in \{0,1\}^{\mathcal{F}\times\mathcal{V}}$ of a d-polytope is complete if and only if $\widetilde{H}_{d-1}(\Gamma(J);\mathbb{Z}_2) \neq 0$.*

PROOF. The set

$$\Pi(P,J) \quad := \quad \bigcup_{F\in\mathcal{F}} \operatorname{conv}\{v\in\mathcal{V} : v\in F\} \quad \subseteq \quad \partial P$$

is a compact subset of the boundary of P: For every "given" facet F of P, it contains the convex hull of all "given" vertices. Thus $\Pi(P,J)$ is a polyhedral complex, called a *partial polytope*, covered by its convex (and hence contractible) cells $\operatorname{conv}\{v \in \mathcal{V} : v \in F\}$. According to the nerve theorem [Björner(1995)], the crosscut complex $\Gamma(J)$ has the same homotopy type as the set $\Pi(P,J)$. In particular, the homology of the set $\Pi(P,J)$ and of the crosscut complex coincide.

In the **yes** case, if the sets of vertices and facets both are complete, $\Pi(P,J)$ is the complete boundary of P, homeomorphic to $\mathbb{S}^{d-1}$, and therefore we have $\widetilde{H}_{d-1}(\Gamma(\mathcal{F});\mathbb{Z}_2) \cong \mathbb{Z}_2$.

In the **no** case, if the vertex or the facet list is incomplete, then $\Pi(P,J)$ is a proper subset of ∂P, which is a subcomplex of a suitable triangulation of ∂P, so it cannot have $(d-1)$-dimensional homology. $\square$

One might be tempted to ask: Why work with the crosscut complex instead of a triangulation of $\Pi(P,J)$? However, in general, such a triangulation cannot derived from the input to CompletenessC nor from the input to CompletenessG; see [Joswig et al.(2000)].

The complexity status of the problem to compute the rank of an arbitrary homology group, or even to decide whether a certain homology group vanishes, seems to be open; see [Kaibel and Pfetsch(2003), Problem 33]. Thus currently our best option is based on explicitly computing simplicial homology via boundary matrices, as in Algorithm 1.

For a brief introduction to simplicial homology including an explicit definition of the k-th boundary matrix ∂_k see the Appendix.

Algorithm 1 CompletenessViaHomology(d,J)
Input: integer $d \geq 0$; an incidence matrix minor J of a d-polytope
Output: answer **yes/no** *to the question whether J is complete*

(1) *generate $\mathbb{Z}_2$-boundary matrices ∂_d and ∂_{d-1} for $\Gamma(J)$*
(2) *if* $\dim_{\mathbb{Z}_2} \ker \partial_{d-1} > \operatorname{rank}_{\mathbb{Z}_2} \partial_d$ *then*
    return ***yes***
  else
    return ***no***

To estimate the costs of this computation, suppose that $n = |\mathcal{V}|$, $m = |\mathcal{F}|$, and that the maximum cardinality of any facet equals s. Thus $J \in \{0,1\}^{m \times n}$, and every row of J contains at most s ones. Then the size of the relevant boundary matrices is bounded from above by $\binom{s}{d+1}m \times \binom{s}{d}m$ and $\binom{s}{d}m \times \binom{s}{d-1}m$, respectively. We use Gaussian elimination over $\mathbb{Z}_2$ to compute the rank and the corank, respectively.

Corollary 6 *The algorithm* CompletenessViaHomology(d, J) *has a polynomial running time if s is bounded by $d + c$, for an absolute constant $c \geq 0$.*

The corollary refers to an interesting case: A d-polytope is *simplicial* if each proper face is a simplex or, equivalently, each facet contains exactly d vertices. We infer that the running time of CompletenessViaHomology for simplicial polytopes is bounded by $O(dm^3)$.

It has been observed [Bremner et al.(1998)] that FacetEnumeration for a polytope P is polynomially equivalent to FacetEnumeration for the dual polytope P^*. Using our techniques, a similar result can be obtained directly. If I is an incidence matrix for P, then the transposed matrix I^{tr} is an incidence matrix for P^*. Any minor J of I is complete if and only if its transpose is a complete minor of I^{tr}. This leads to the following modification of our algorithm. While s was defined above as the maximal row size of the input incidence matrix minor, define

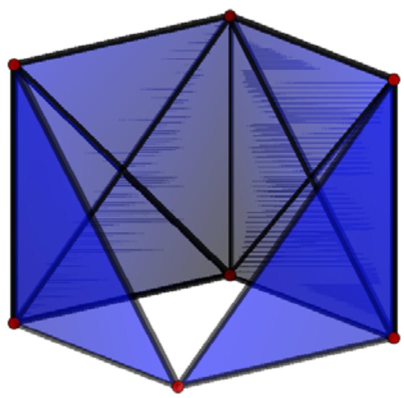
$$s' \ := \ \min\{\text{maximal row size}, \text{maximal column size}\}.$$

Thus we modify our algorithm: It should first compare the sizes of the primal and the dual problem, and then perform the (reduced) homology computation for the smaller problem. The modified algorithm CompletenessViaHomology(d, J) has polynomial running time if s' is bounded by "d plus a constant." In particular, this yields an $O(d(n+m)^3)$-algorithm for the CompletenessC problem specialized to polytopes which are simplicial or *simple*, that is, dual to a simplicial polytope.

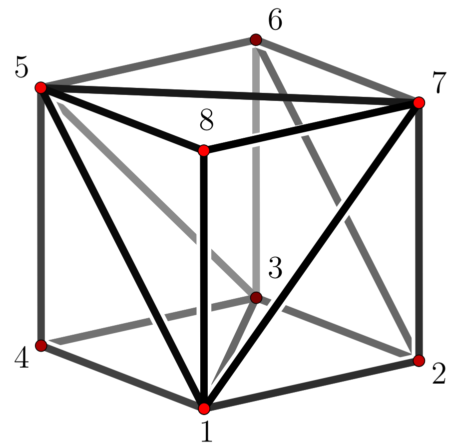
We note, however, that these running times are neither optimal nor the best available: The reverse search algorithm [Avis and Fukuda(1992)] computes the convex hull (and thereby solves CompletenessG) of a simplicial polytope in $O(dnm)$ steps.

6 A Certificate for Incompleteness

Let P be a d-polytope with ordered vertex set $\mathcal{V}' = \{v_1, \dots, v_n\}$ and facet set $\mathcal{F}'$. Inductively, define a sequence $\Delta_0, \dots, \Delta_n$ of polytopal subdivisions of the boundary complex ∂P: Set $\Delta_0 := \partial P$. In order to obtain Δ_k replace each facet F of Δ_{k-1} which contains v_k by the set of cones with apex v_k over those facets of F which do not contain v_k. The final subdivision is a triangulation $\Delta(P) := \Delta_m$ of ∂P, the *pulling triangulation* [Lee(1997)] with respect to the chosen ordering of $\mathcal{V}'$. For an example of a pulling triangulation see Figure 1(b).



(a) The (3-dimensional) crosscut complex of some partial 3-cube C. The two quadrangle faces of C yield tetrahedra in $\Gamma(C)$, which are displayed almost flat. The crosscut complex is homotopy equivalent to $\mathbb{S}^1$ and hence the second homology group vanishes.



(b) The pulling triangulation of the boundary of a 3-cube with respect to a "Klee-Minty" vertex ordering. The facet $\{1,7,8\}$ of the triangulation corresponds to the flag $\{8\} \subset \{7,8\} \subset \{1,2,7,8\}$ of the cube.

Fig. 1. Crosscut complex and pulling triangulation.

The pulling triangulation of ∂P has several nice properties (not shared, for example, by the "placing triangulation") that may be exploited for our purposes. First, its combinatorics is determined by the combinatorics of P; see below. Furthermore, if we use a linear ordering of the vertex set $\mathcal{V}'$ in which the vertices in $\mathcal{V} \subseteq \mathcal{V}'$ come first, then the corresponding pulling triangulation of the boundary of P contains a triangulation of $\Pi(P, J)$ as a subcomplex.

Let us now identify the vertex set $\mathcal{V}'$ with the set $[n] = \{1, \ldots, n\}$ and each facet $F \in \mathcal{F}'$ with the subset of $[n]$ that corresponds to the vertices of F. Thus any triangulation of ∂P is encoded by a collection of d-subsets of $[n]$, that is, to a subset of $\binom{[n]}{d}$. We write $\{v_1, \ldots, v_d\}_<$ for a d-subset of $[n]$ with $v_1 < v_2 < \cdots < v_d$.

Lemma 7 *Let P be a d-polytope whose vertex set is labeled by $[n]$. Then a set $\{v_1, \ldots, v_d\}_< \in \binom{[n]}{d}$ corresponds to a facet of the pulling triangulation of ∂P (with respect to the chosen vertex labeling) if and only if there is a complete flag of faces*

$$\emptyset \subset G_0 \subset G_1 \subset \ \ldots \ \subset G_{d-1} \subset P,$$

such that v_i is the smallest vertex in G_{d-i} for $1 \le i \le d$, that is, if there are facets $F_1, \ldots, F_d$ of P such that

$$v_i \;=\; \min(F_1 \cap \ldots \cap F_i)$$

for $1 \le i \le d$.

PROOF. Every pulling facet $\{v_1, \ldots, v_d\}_<$ lies in a facet $F_1 = G_{d-1}$ of P, with $v_1 = \min G_{d-1}$. It is a cone with apex v_1 and base $G_{d-2} \subset G_{d-1}$. The existence of the rest of the maximal flag $(G_i)_{0 \le i < d}$ follows recursively. Given the flag, the existence of the facets $F_1, \ldots, F_d$ follows [Ziegler(1995), Lect. 2]. (Given a complete flag, the corresponding sequence of facets F_i is uniquely determined if P is simple, but not in general.) $\square$

If we have an arbitrary incidence matrix minor J of a d-polytope P, then we can read the combinatorial characterization of the pulling triangulation from Lemma 7 as the definition of a complex that coincides with the pulling triangulation of ∂P in case J is complete, but that is well-defined in general:

Definition 8 *Given an integer $d > 0$ and a $0/1$-matrix $J \in \{0,1\}^{m \times n}$, which we interpret as the incidence matrix of a set system $\mathcal{F} \subseteq 2^{[n]}$, the* pulling complex *of d and J is*

$$\Delta(d, J) := \Big\{ \{v_1, \ldots, v_d\}_< \in \tbinom{[n]}{d} : \text{there are } \bar{F}_1, \ldots, \bar{F}_d \in \mathcal{F} \text{ such that } v_i = \min(\bar{F}_1 \cap \ldots \cap \bar{F}_i) \text{ for } 1 \le i \le d \Big\}.$$

Lemma 9 *Let P be a d-dimensional polytope with vertex set $\mathcal{V}'$ and facet set $\mathcal{F}'$, and let J be a incidence matrix minor corresponding to subsets $\mathcal{V} \subseteq \mathcal{V}'$ and $\mathcal{F} \subseteq \mathcal{F}'$. Let $\bar{P} \subseteq P$ be the convex hull of the vertices in $\mathcal{V}$. Fix a linear ordering on the vertex set $\mathcal{V}'$ such that the vertices in $\mathcal{V}$ come first.*

Then the simplicial complex $\Delta(d, J)$ is a subcomplex of $\Delta(P)$ as well as of $\Delta(\bar{P})$. In particular, $\Delta(d, J)$ is a proper subcomplex of $\Delta(P)$, unless the minor J is complete, that is, $J = I_P$. In the incomplete case $\Delta(d, J)$ may even be empty.

PROOF. Let $\{v_1, \ldots, v_d\}_< \in \Delta(d, J)$, then there are $\bar{F}_1, \ldots, \bar{F}_d \in \mathcal{F}$ such that $v_i = \min(\bar{F}_1 \cap \ldots \cap \bar{F}_i)$. Since J is an incidence matrix minor of P, there are facets $F_i \supseteq \bar{F}_i$ of P, and by the assumption on the vertex ordering the vertices in $\bar{F}_i$ come first, so $\min(\bar{F}_1 \cap \ldots \cap \bar{F}_i) = \min(F_1 \cap \ldots \cap F_i)$, which yields $\{v_1, \ldots, v_d\}_< \in \Delta(P)$.

Now $\bar{P} = \text{conv}(\mathcal{V})$, and the $\bar{F}_i = F_i \cap \mathcal{V}$ are vertex sets of faces (not necessarily facets) of $\bar{P}$. If the vertices $v_i = \min(\bar{F}_1 \cap \ldots \cap \bar{F}_i)$ are distinct, then the faces $\bar{F}_1 \cap \ldots \cap \bar{F}_i$ form a complete flag in the face lattice of $\bar{P}$, and thus $\{v_1, \ldots, v_d\}_< \in \Delta(\bar{P})$, by Lemma 7. $\square$

In particular, $\Delta(d, J)$ triangulates a subset of the complex $\Pi(P, J)$ that appears in the proof of Theorem 5.

Now we present a polynomially-checkable certificate for the case that J is incomplete. Note, however, that this result does *not* prove that COMPLETENESSC is in co-NP: We are not able to check (in polynomial time) whether the input is valid, that is, whether J is actually an incidence matrix minor of some d-polytope. On the other hand, we could derive that COMPLETENESSG is in co-NP, but that is clear anyway, since any missing facet provides a certificate.

Theorem 10 *Any* **no** *instance of the problem* COMPLETENESSC(d, J) *has a certificate that can be verified in polynomial time.*

PROOF. The minor J is incomplete if and only if the pulling complex $\Delta(d, J)$ is not a complete triangulation of a d-polytope boundary. Two cases arise. The first one is if $\Delta(d, J) = \emptyset$, in which case Algorithm 2 described below will certify in polynomial time that J is not complete.

The second case is if $\Delta(d, J)$ is non-empty but incomplete. In this case (since the dual graph of the pulling triangulation $\Delta(P)$ is connected) there is a facet $\{v_1, \ldots, v_d\} \in \Delta(d, J)$ together with an index i such that there is no second facet of $\Delta(d, J)$ that contains $\{v_1, \ldots, v_d\} \setminus \{v_i\}$. In this situation our certificate is the set $\{v_1, \ldots, v_d\} \setminus \{v_i\}$. Calling ISPULLINGFACET for every d-subset of $[n]$ which contains the certificate, this certificate can be verified in polynomial time, since there are $n - d + 1$ of these subsets. $\square$

Now we proceed by describing the two subroutines needed for Theorem 10. The first one is Algorithm 2: Given an incidence matrix minor J it either finds a facet of $\Delta(d, J)$ in polynomial time or it detects that J is incomplete. The correctness follows from Lemma 7. Our specific formulation of the algorithm produces a pulling triangulation facet which does not contain 1: This restriction does not hurt, since $\Delta(d, J)$ must contain such a facet if J is complete.

Algorithm 2 FINDPULLINGFACET(d, J)
Input: incidence matrix minor $J \in \{0, 1\}^{m \times n}$ *of a d-polytope;*
d-tuple $\{v_1, \ldots, v_d\}_< \in \binom{[n]}{d}$; (d, J) *as above*
Output: a facet $\{v_1, \ldots, v_d\} \in \Delta(d, J)$, *or* ***incomplete***

(1) $S \leftarrow [n]$
(2) for each i *from* 1 *to* d *do*
    $F_i \leftarrow$ *any* $F \in \mathcal{F}$ *such that*
        $\min S \notin F$, $F \cap S \neq \emptyset$, *and* $|F \cap S|$ *is maximal*
    if no such facet exists then
        return ***incomplete***
    $S \leftarrow S \cap F_i$
    $v_i \leftarrow \min S$
(3) return $\{v_1, \ldots, v_d\}_<$

Our second subroutine, Algorithm 3, checks whether a given set of d vertices is a facet of the pulling complex $\Delta(d, J)$ or not. Its correctness again follows from the characterization in Lemma 7. Its running time is bounded by $O(d(n+m))$.

Algorithm 3 IsPullingFacet$(d, J, \{v_1, \ldots, v_d\}_<)$
Input: (d, J) *as above*
Output: answer ***yes/no*** *to the question whether* $\{v_1, \ldots, v_d\} \in \Delta(d, J)$

(1) for each i *from* d *downto* 1 *do*
    compute the set $\mathcal{F}_i$ *of all facets (i.e., rows of* J*) that*
        contain $\{v_i, \ldots, v_d\}$
(2) for each i *from* 1 *to* d *do*
    $F_i \leftarrow$ *any* $F \in \mathcal{F}_i$ *with* $v_i = \min(F_1 \cap \ldots \cap F_{i-1} \cap F)$
  if no such F *exists then*
    return ***no***
(3) return ***yes***

We close our discussion with a pointer to a specific special case: It would be interesting to know whether Completeness(d, J) has a polynomial time solution for the very special case where J has all columns and lacks at most one row.

Acknowledgements

We are grateful to Volker Kaibel, Marc E. Pfetsch and Mark de Longueville for helpful comments. Moreover, the first author is indebted to Günter Rote for an enlightening discussion on the subject.

References

[Avis(2000)] Avis, D., 2000. A revised implementation of the reverse search vertex enumeration algorithm. In: Polytopes—combinatorics and computation

(Oberwolfach, 1997). Vol. 29 of DMV Sem. Birkhäuser, Basel, pp. 177–198.

[Avis(2001)] Avis, D., 2001. lrs — a C implementation of the reverse search vertex enumeration algorithm, version 4.1. http://cgm.cs.mcgill.ca/~avis/C/lrs.html.

[Avis et al.(1997)] Avis, D., Bremner, D., Seidel, R., 1997. How good are convex hull algorithms? Comput. Geom. 7 (5-6), 265–301, 11th ACM Symposium on Computational Geometry (Vancouver, BC, 1995).

[Avis and Fukuda(1992)] Avis, D., Fukuda, K., 1992. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. Discrete Comput. Geom. 8 (3), 295–313, aCM Symposium on Computational Geometry (North Conway, NH, 1991).

[Björner(1995)] Björner, A., 1995. Topological methods. In: Handbook of combinatorics, Vol. 1, 2. Elsevier, Amsterdam, pp. 1819–1872.

[Bremner(1999)] Bremner, D., 1999. Incremental convex hull algorithms are not output sensitive. Discrete Comput. Geom. 21 (1), 57–68.

[Bremner et al.(1998)] Bremner, D., Fukuda, K., Marzetta, A., 1998. Primal-dual methods for vertex and facet enumeration. Discrete Comput. Geom. 20 (3), 333–357, ACM Symposium on Computational Geometry (Nice, 1997).

[Chazelle(1993)] Chazelle, B., 1993. An optimal convex hull algorithm in any fixed dimension. Discrete Comput. Geom. 10 (4), 377–409.

[Freund and Orlin(1985)] Freund, R. M., Orlin, J. B., 1985. On the complexity of four polyhedral set containment problems. Math. Programming 33 (2), 139–145.

[Fukuda(2000)] Fukuda, K., 2000. Frequently asked questions in polyhedral computation. http://www.cs.mcgill.ca/~fukuda/soft/polyfaq/polyfaq.html.

[Fukuda(2003)] Fukuda, K., 2003. CDD — a C-implementation of the double description method; cddlib, version 0.93. http://www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html.

[Gawrilow and Joswig(1997–2003)] Gawrilow, E., Joswig, M., 1997–2003. polymake, version 2.0: a software package for analyzing convex polytopes. http://www.math.tu-berlin.de/polymake.

[Gawrilow and Joswig(2001)] Gawrilow, E., Joswig, M., 2001. polymake: an approach to modular software design in computational geometry. In: Proc. 17th Annual Symp. Computational Geometry, ACM, 2001, June 3-5, Medford, MA. pp. 222–231.

[Grötschel et al.(1993)] Grötschel, M., Lovász, L., Schrijver, A., 1993. Geometric algorithms and combinatorial optimization, 2nd Edition. Vol. 2 of Algorithms and Combinatorics. Springer-Verlag, Berlin.

[Joswig et al.(2000)] Joswig, M., Kaibel, V., Pfetsch, M. E., Ziegler, G. M., 2000. Ambiguous incidences of unbounded polyhedra. Electronic Geometry Models, Model No. 2000.05.001, `http://www.eg-models.de/2000.05.001`.

[Joswig et al.(2001)] Joswig, M., Kaibel, V., Pfetsch, M. E., Ziegler, G. M., 2001. Vertex-facet incidences of unbounded polyhedra. Adv. Geom. 1 (1), 23–36.

[Kaibel and Pfetsch(2003)] Kaibel, V., Pfetsch, M. E., 2003. Some algorithmic problems in polytope theory. In: Joswig, M., Takayama, N. (Eds.), Algebra, Geometry, and Software Systems. Springer-Verlag, Heidelberg, pp. 23–48.

[Lee(1997)] Lee, C. W., 1997. Subdivisions and triangulations of polytopes. In: Goodman, J. E., O'Rourke, J. (Eds.), Handbook of Discrete and Computational Geometry. CRC Press, Boca Raton, pp. 271–290.

[Lovász(1986)] Lovász, L., 1986. An algorithmic theory of numbers, graphs and convexity. Vol. 50 of CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

[McCarthy et al.(2002)] McCarthy, N., Ogilvie, D., Spitkovsky, I., Zobin, N., 2002. Birkhoff's theorem and convex hulls of Coxeter groups. Linear Algebra Appl. 347, 219–231.

[McMullen(1970)] McMullen, P., 1970. The maximum numbers of faces of a convex polytope. Mathematika 17, 179–184.

[Munkres(1984)] Munkres, J. R., 1984. Elements of algebraic topology. Addison-Wesley Publishing Company, Menlo Park, CA.

[Seidel(1991)] Seidel, R., 1991. Small-dimensional linear programming and convex hulls made easy. Discrete Comput. Geom. 6 (5), 423–434.

[Ziegler(1995)] Ziegler, G. M., 1995. Lectures on polytopes. Vol. 152 of Graduate Texts in Mathematics. Springer-Verlag, New York.

Appendix: Simplicial Homology in a Nutshell

Let V be a finite ordered set. An *(abstract) simplicial complex* on V is a non-empty subset $\Delta \subseteq 2^V$ which is closed with respect to forming subsets. A *k-face* σ is an element of Δ of cardinality $k+1$; we then define $\dim \sigma = k$. The dimension of Δ is $\dim \Delta = \max\{\dim \sigma \mid \sigma \in \Delta\}$. The number of k-dimensional faces of Δ is $f_k(\Delta)$, with $f_{-1}(\Delta) = 1$ corresponding to the empty set, and $f_k(\Delta) = 0$ for $k < -1$ or $k > \dim \Delta$.

Let R be an arbitrary commutative ring with unit, and let $C_k(\Delta; R)$ be the free R-module generated by the set of k-faces of Δ. The elements of $C_k(\Delta)$ are called *k-chains*. We define the *boundary* of a k-face $\sigma = \{v_0, \ldots, v_k\}$, where $v_0 < v_1 < \cdots < v_k$, to be the $(k-1)$-chain

$$\partial\sigma = \sum_{i=0}^{k} (-1)^i \{v_0, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k\}.$$

This map has a unique R-linear extension $\partial_k : C_k(\Delta; R) \to C_{k-1}(\Delta; R)$. The empty set generates $C_{-1}(\Delta; R)$, and ∂_0 is surjective. By definition $C_k(\Delta; R) = \{0\}$ for $k < -1$ or $k > \dim \Delta$. The *k-cycles* $Z_k(\Delta; R) = \ker \partial_k$ and the *k-boundaries* $B_k(\Delta; R) = \operatorname{im} \partial_{k+1}$ are free R-modules. For any $(k+1)$-face τ one can verify that $\partial_k(\partial_{k+1}\tau) = 0$, and hence $B_k(\Delta; R) \subseteq Z_k(\Delta; R)$. The quotient

$$\widetilde{H}_k(\Delta; R) = Z_k(\Delta; R)/B_k(\Delta; R)$$

is the k-th *reduced homology module* of Δ with coefficients in R.

We summarize some key properties of (reduced simplicial) homology: If Δ is homotopy equivalent to Δ', then $\widetilde{H}_k(\Delta; R) \cong \widetilde{H}_k(\Delta'; R)$ for all k. If Δ is connected, then $\widetilde{H}_0(\Delta; R) = 0$. If Δ is a triangulation of the d-sphere $\mathbb{S}^d$, then $\widetilde{H}_d(\Delta; R) \cong R$, and all other reduced homology modules vanish.

In our application, we are solely interested in the case where $R = \mathbb{Z}_2$ is a field. Then $C_k(\Delta; \mathbb{Z}_2)$ is a $\mathbb{Z}_2$-vector space of dimension $f_k(\Delta)$, the boundary operator ∂_k is given by a $\mathbb{Z}_2$-matrix of size $f_{k-1}(\Delta) \times f_k(\Delta)$, and $\widetilde{H}_k(\Delta; \mathbb{Z}_2)$ is a vector space of dimension

$$\dim_{\mathbb{Z}_2} Z_k(\Delta; \mathbb{Z}_2) - \dim_{\mathbb{Z}_2} B_k(\Delta; \mathbb{Z}_2) \;=\; f_k(\Delta) - \operatorname{rank}_{\mathbb{Z}_2} \partial_k \;-\; \operatorname{rank}_{\mathbb{Z}_2} \partial_{k+1}.$$

The reader is referred to the monograph [Munkres(1984)] for a detailed presentation.