



LOVIS ANDERSON<sup>1</sup> BENJAMIN HILLER<sup>2</sup>

# **A Sweep-Plane Algorithm for the Computation of the Volume of a Union of Polytopes**

---

<sup>1</sup>  0000-0002-4316-1862  
<sup>2</sup>  0000-0003-4335-0775

Zuse Institute Berlin  
Takustr. 7  
14195 Berlin  
Germany

Telephone: +49 30-84185-0  
Telefax: +49 30-84185-125

E-mail: [bibliothek@zib.de](mailto:bibliothek@zib.de)  
URL: <http://www.zib.de>

ZIB-Report (Print) ISSN 1438-0064  
ZIB-Report (Internet) ISSN 2192-7782

# A Sweep-Plane Algorithm for the Computation of the Volume of a Union of Polytopes

Lovis Anderson and Benjamin Hiller

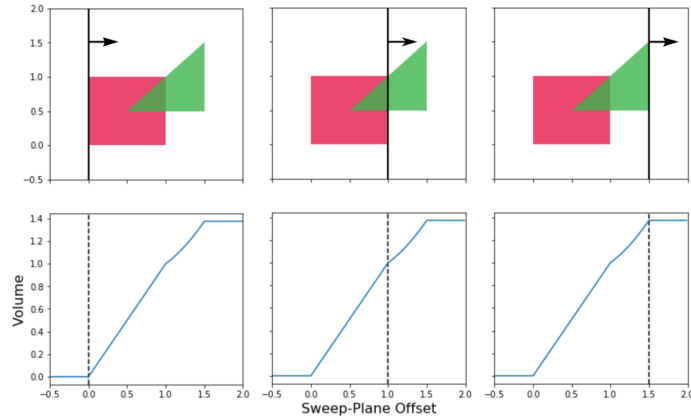
Zuse Institute Berlin  
anderson@zib.de

**Abstract.** Optimization models often feature disjunctions of polytopes as submodels. Such a disjunctive set is initially (at best) relaxed to its convex hull, which is then refined by branching. To measure the error of the convex relaxation, the (relative) difference between the volume of the convex hull and the volume of the disjunctive set may be used. This requires a method to compute the volume of the disjunctive set. We propose a revised variant of an old algorithm by Bieri and Nef [BN83] for this purpose. The algorithm uses a sweep-plane to incrementally calculate the volume of the disjunctive set as a function of the offset parameter of the sweep-plane.

## 1 Introduction

A (*polyhedral*) *disjunctive set* is a set that may be written as a union of polytopes  $\mathcal{P} = \bigcup_{i=1}^r P^i$ . Disjunctive sets are a frequently occurring structure in many optimization models. As they are in general nonconvex sets, global optimization solvers rely on convex relaxations to obtain bounds. Using tight convex relaxations during the solution process is crucial to be able to solve such models. The tightness of the convex relaxation, i.e. the convexification error, may be naturally measured using the volume. The volume may be used to compare two competing convex relaxations [LSS18], or to numerically quantify the convexification error by comparing the volume of  $\mathcal{P}$  and its convex relaxation. This may be used to devise a branching scheme for  $\mathcal{P}$  that quickly improves the tightness of the relaxation [HW17].

Computing the convexification error numerically requires computing the volume of the disjunctive set  $\mathcal{P}$ . Naively, this can be done via inclusion/exclusion and leveraging the well-studied algorithms for computing the volume of a single polytope. However, this is often inefficient. Bieri and Nef [BN83] proposed a method for computing the volume of a disjunctive set which calculates the volume up to a sweep-plane in a *beneath and beyond* approach. The sweep-plane is a hyperplane given by  $\{\mathbf{x} \mid \langle \mathbf{s}, \mathbf{x} \rangle = \lambda\}$  where  $\mathbf{s}$  is the sweep-direction and  $\lambda$  is the time in the sweep. Similar to a tomography, the sweep-plane is moved through space, as visualized by Figure 1. The sweep-plane volume function  $f(\lambda)$  describes the volume  $\text{Vol}(\mathcal{P} \cap \{\mathbf{x} \mid \langle \mathbf{s}, \mathbf{x} \rangle \leq \lambda\})$  that lies behind the sweep-plane. This function allows us to see interesting changes in the volume over the course of the sweep, which may be useful to guide decomposition methods as in [HW17].



**Fig. 1.** Top row: A union of polytopes with sweep-planes at different times. Bottom row: The corresponding sweep-plane volume graphs.

We propose a revised algorithm based on Bieri and Neff's general idea. First we review a formula for the volume of a polytope as introduced in [Law91]. Afterwards we extend the formula to unions of polytopes and discuss the resulting algorithm. A preliminary python implementation of the proposed algorithm can be found at <https://gitlab.com/LovisAnderson/sweepvolume>.

## 2 The Volume of a Polytope

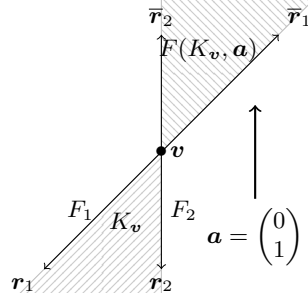
We will first introduce a formula for the volume of a single polytope  $P$ , which is given by  $P := \{\mathbf{x} \in \mathbb{R}^d \mid A\mathbf{x} \leq \mathbf{b}\}$ . We will restrict ourselves to simple polytopes, i.e. polytopes for which at each vertex exactly  $d$  constraints are active. The non-simple case can either be handled by perturbation of the constraints or regularization as described in [BN83].

### 2.1 The Conic Decomposition

The *vertex cone*  $K_{\mathbf{v}}$  at a vertex  $\mathbf{v}$  is given by the constraints that are active at that vertex. We can also describe that cone as a Minkowski sum of the vertex and the positive hull of its rays. In particular the rays on the  $d$  edges suffice to describe the positive hull. We can write  $K_{\mathbf{v}} = \{\mathbf{v}\} + \text{pos}(\mathbf{r}_1, \dots, \mathbf{r}_d)$ . The rays  $\mathbf{r}_1, \dots, \mathbf{r}_d$  are unique up to scaling and can be calculated from the constraint matrix. Let  $A_{\mathbf{v}}$  be the matrix that is given by the constraints that are active at  $\mathbf{v}$ . Then the rays are given by

$$(\mathbf{r}_1, \dots, \mathbf{r}_d) = -A_{\mathbf{v}}^{-1}. \quad (1)$$

A *forward cone*  $F(K_{\mathbf{v}}, \mathbf{s})$  for a vertex cone  $\mathbf{s}$  results from flipping the rays such that they point in the same direction as  $\mathbf{s}$ . Formally,



**Fig. 2.** The vertex cone and the forward cone.

$$F(K_v, \mathbf{s}) := \{v\} + \text{pos}(\bar{\mathbf{r}}_1, \dots, \bar{\mathbf{r}}_d) \quad \text{with} \quad \bar{\mathbf{r}}_i := \begin{cases} \mathbf{r}_i & \text{if } \langle \mathbf{r}_i, \mathbf{s} \rangle > 0, \\ -\mathbf{r}_i & \text{else} \end{cases}$$

Figure 2 visualizes the vertex cone and forward cone in an exemplary way.

Theorem 1 is a slight abbreviation of the main result from [Law91] and shows how a polytope can be decomposed into its forward cones using the indicator function  $\mathbb{1}$ .

**Theorem 1.** *Let  $P$  be a simple polytope and  $\mathbf{s}$  be a direction which induces a total ordering  $\mathbf{v}_1, \dots, \mathbf{v}_n$  on the vertices of  $P$  via the dot product, i.e.  $\langle \mathbf{v}_i, \mathbf{s} \rangle < \langle \mathbf{v}_j, \mathbf{s} \rangle$  for  $i < j$ . Let  $K_i$  be the vertex cone at  $\mathbf{v}_i$  and  $\sigma(K_i, \mathbf{s}) := \#\{i \in \{1, \dots, d\} \mid \langle \mathbf{r}_i, \mathbf{s} \rangle < 0\}$  be the number of rays that have to be flipped in order to obtain the forward cone  $F(K_i, \mathbf{s})$  from  $K_i$ . Then we have*

$$\mathbb{1}_P = \sum_{i \in [n]} (-1)^{\sigma(K_i, \mathbf{s})} \mathbb{1}_{F(K_i, \mathbf{s})}. \quad (2)$$

We are interested in the volume up to a sweep-plane, which depends on the direction  $\mathbf{s}$  and a *time* (or geometrically interpreted *offset*)  $\lambda$ . We denote the *negative halfspace* at time  $\lambda$  by  $H^-(\mathbf{s}, \lambda) := \{\mathbf{x} \in \mathbb{R}^d \mid \langle \mathbf{s}, \mathbf{x} \rangle \leq \lambda\}$ . Through integration we can use Theorem 1 to write the volume up to the sweep-plane at time  $\lambda$  as

$$\text{Vol}(P \cap H^-(\mathbf{s}, \lambda)) = \sum_{i \in [n]} (-1)^{\sigma(K_i, \mathbf{s})} \text{Vol}(F(K_i, \mathbf{s}) \cap H^-(\mathbf{s}, \lambda)). \quad (3)$$

## 2.2 The Sweep-Plane Volume of a Cone

To be able to compute the volume of a polytope by using Equation (3) we need a way to compute the sweep-plane volume of a forward cone. The volume of a simplex might be calculated with this neat determinant formula, see [Ste66].

**Fact 2** *The volume of a simplex  $S \subset \mathbb{R}^d$  with vertices  $\mathbf{s}_1, \dots, \mathbf{s}_{d+1}$  is:*

$$\text{Vol}(S) = \frac{1}{d!} |\det((\mathbf{s}_2 - \mathbf{s}_1 \dots \mathbf{s}_{d+1} - \mathbf{s}_1))| \quad (4)$$

We can adopt Equation (4) such that we can use it to compute the sweep-plane volume of a forward cone. The forward cone up to a sweep-plane is a simplex which can be described by its vertex and rays. In this simplex, the offset  $\lambda$  of the sweep is only reflected in the length of the rays and not in their direction.

**Proposition 1.** *Let  $K_{\mathbf{v}} = \{\mathbf{v}\} + \text{pos}(\mathbf{r}_1, \dots, \mathbf{r}_d)$  be a vertex cone of a polytope  $P$ . Let  $\mathbf{s}$  be such that it induces a total ordering on the vertices of  $P$ . Let  $\lambda_{\mathbf{v}} := \langle \mathbf{s}, \mathbf{v} \rangle$ . Then for the volume of the forward cone  $F(K_{\mathbf{v}}, \mathbf{s})$  and  $\lambda > \lambda_{\mathbf{v}}$  it holds that*

$$\text{Vol}(F(K_{\mathbf{v}}, \mathbf{s}) \cap H^-(\mathbf{s}, \lambda)) = (-1)^{\sigma(K_{\mathbf{v}}, \mathbf{s})} (\lambda - \lambda_{\mathbf{v}})^d \frac{|\det(\mathbf{r}_1, \dots, \mathbf{r}_d)|}{d! \prod_{i \in [d]} \langle \mathbf{s}, \mathbf{r}_i \rangle}. \quad (5)$$

For  $\lambda \leq \lambda_{\mathbf{v}}$  clearly holds  $\text{Vol}(F(K_{\mathbf{v}}, \mathbf{s}) \cap H^-(\mathbf{s}, \lambda)) = 0$ .

We need the vertices of  $P$  to be strictly ordered with respect to  $\mathbf{s}$  to assure that the dot product in Proposition 1 is not 0. However, this can always be achieved by perturbing  $\mathbf{s}$ .

### 3 Extension to Unions of Polytopes

We want to extend our decomposition to a union of polytopes  $\mathcal{P} = \bigcup_{i=1}^r P^i$ . We now look at all constraints which describe polytopes in  $\mathcal{P}$  and assume we have them ordered as  $\{\mathbf{a}_i^T \mathbf{x} \leq b_i, i \in \{1, \dots, n\}\}$ . We call  $\mathcal{A} := \{(\mathbf{a}_i, b_i) \mid i \in \{1, \dots, n\}\}$  a *hyperplane arrangement*. The corresponding equality constraints  $\{\mathbf{a}_i^T \mathbf{x} = b_i \mid i \in \{1, \dots, n\}\}$  decompose the space into *cells*, which are defined as the closures of the connected components of

$$\mathbb{R}^d \setminus \{\mathbf{x} \in \mathbb{R}^d \mid \exists i \text{ s.t. } \mathbf{a}_i^T \mathbf{x} = b_i\}.$$

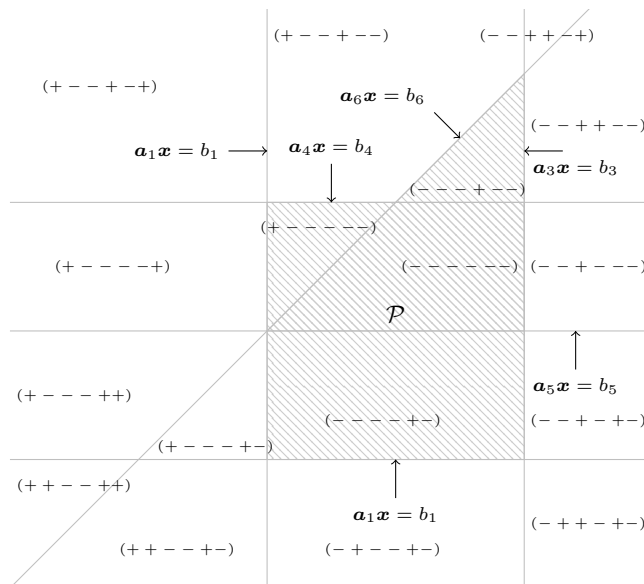
That means that the cells are the (closed) sets which are not intersected by a hyperplane which is given by an equality constraint, see Figure 3. The union of polytopes can be written as the union of a subset of these cells. All cells are polytopes and are mutually internally disjoint. Therefore the sum of the volumes of the cells which are subsets of the union of polytopes equals the total volume of the union of polytopes.

**Proposition 2.** *Let  $\mathcal{P} = \bigcup_{i=1}^r P_i$  be a union of polytopes and  $\mathcal{A}_{\mathcal{P}}$  be the corresponding hyperplane arrangement. Let  $C_1, \dots, C_l$  be the cells of  $\mathcal{A}_{\mathcal{P}}$  which are subsets of  $\mathcal{P}$ . We can then write  $\text{Vol}(\mathcal{P}) = \sum_{i=1}^l \text{Vol}(C_i)$ .*

The following definition generalizes the well-known concept of a position vector [Ede87] and helps us to describe cells and other faces of a hyperplane arrangement.

**Definition 1.** *Let  $\mathcal{A} := \{(\mathbf{a}_i, b_i) \mid i \in \{1, \dots, m\}\}$  be a hyperplane arrangement. We call a vector  $(p_1, \dots, p_m) =: \mathbf{p} \in \{+, -, 0, \bullet\}^m$  a position vector for  $\mathcal{A}$ . We associate with it the set*

$$\mathcal{A}(\mathbf{p}) := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a}_i^T \mathbf{x} \begin{cases} \leq b_i & \text{if } p_i = -, \\ \geq b_i & \text{if } p_i = +, \\ = b_i & \text{if } p_i = 0, \end{cases} \quad i \in \{1, \dots, m\}\}.$$



**Fig. 3.** The hyperplane arrangement  $\mathcal{A}(\mathcal{P})$ , with  $\mathcal{P}$  being the union of a square and a triangle and the region describing position vectors.

To ensure that no two position vectors correspond to the same set we also demand that for all  $i \in [m]$  with  $p_i = \bullet$  there is a  $\mathbf{x}_1 \in \mathcal{A}(\mathbf{p})$  with  $\mathbf{a}_i^T \mathbf{x}_1 < b_i$  and a  $\mathbf{x}_2 \in \mathcal{A}(\mathbf{p})$  with  $\mathbf{a}_i^T \mathbf{x}_2 > b_i$ . If there is no such pair  $\mathbf{x}_1, \mathbf{x}_2$  we can replace  $\bullet$  with  $+$  or  $-$ .

We can describe any face of the hyperplane arrangement using position vectors. Including  $\bullet$  in the definition also allows us to describe polytopes and vertex cones. This allows us to compare those sets in a combinatorial way. For position vectors  $\mathbf{p}^1, \mathbf{p}^2$  we say that  $\mathbf{p}^1$  is *included* in  $\mathbf{p}^2$  if

$$\begin{aligned} p_i^1 &= p_i^2 \vee p_i^2 = \bullet && \text{if } p_i^1 \in \{+, -\} \\ p_i^2 &= \bullet && \text{if } p_i^1 = \bullet. \end{aligned}$$

If  $\mathbf{p}^1$  is included in  $(\mathbf{p}^2)$  then  $\mathcal{A}(\mathbf{p}^1) \subseteq \mathcal{A}(\mathbf{p}^2)$  and we write  $\mathbf{p}^1 \subseteq \mathbf{p}^2$ .

**Corollary 1.** Let  $\mathcal{P} = \bigcup_{i=1}^r$  be a union of polytopes and  $\mathcal{A}_{\mathcal{P}}$  be the corresponding hyperplane arrangement. Let  $C_1, \dots, C_l$  be the cells of  $\mathcal{A}_{\mathcal{P}}$  which are subsets of  $\mathcal{P}$ . Let  $\mathbf{v}$  be a vertex whose active constraints in the arrangement are given by the indices  $J := \{j_1, \dots, j_d\}$ . We define the restricted position vector of  $K_{\mathbf{v}}$  only featuring the active constraints at  $\mathbf{v}$  as  $\mathbf{p}^{\mathbf{v}}(K_{\mathbf{v}}) := (p_1^{\mathbf{v}}, \dots, p_m^{\mathbf{v}})$  with  $p_i^{\mathbf{v}} := \begin{cases} p_i(K_{\mathbf{v}}) & \text{if } i \in J \\ \bullet & \text{else} \end{cases}$ . A cone  $K_{\mathbf{v}}$  at a vertex  $\mathbf{v}$  of the hyperplane arrangement is the vertex cone of some cell  $C_j$  if and only if there is a  $P_i$  such that

$$\mathbf{p}(\mathbf{v}) \subseteq \mathbf{p}(P_i) \quad \wedge \quad \mathbf{p}^{\mathbf{v}}(K_{\mathbf{v}}) \subseteq \mathbf{p}(P_i). \quad (6)$$

This allows us to algorithmically find all vertex cones of cells belonging to  $\mathcal{P}$ .

## 4 The Algorithm

The steps in the algorithm for computing the sweep-plane volume function for a union of polytopes  $\mathcal{P} := \bigcup_{i=1}^r P_i$  can be summarized as follows:

1. Calculate the vertices of the hyperplane arrangement
2. For each vertex calculate the vertex cones belonging to  $\mathcal{P}$  using Corollary 1
3. Calculate the sweep-plane volume function using Proposition 1

In our implementation we do step 1 and 2 independently of step 3. We build a data structure which we can then use in step 3 to calculate the sweep-plane volume function. Almost all of the computational complexity lies in building the data structure. Once the data structure is built, we are able to calculate sweep-plane volume functions for many different directions. Through comparing these functions, insights into the structure of the disjunctive set might be obtained. Detecting structural changes in volume might be useful to guide decomposition methods as in [HW17].

*Acknowledgements* The authors thank the DFG for support within project A04 in CRC TRR154 and the BMBF Research Campus Modal (fund number 05M14ZAM) and ICT COST Action TD1207 for additional support.

## References

- BN83. Hanspeter Bieri and Walter Nef, *A sweep-plane algorithm for computing the volume of polyhedra represented in boolean form*, *Linear Algebra and its Applications* **52** (1983), 69–97.
- Ede87. Herbert Edelsbrunner, *Algorithms in combinatorial geometry*, Springer, 1987.
- HW17. B. Hiller and T. Walther, *Improving branching for disjunctive polyhedral models using approximate convex decompositions*, ZIB-Report 17-68, Zuse Institute Berlin, 2017.
- Law91. Jim Lawrence, *Polytope volume computation*, *Mathematics of computation* **57** (1991), no. 195, 259–271.
- LSS18. Jon Lee, Daphne Skipper, and Emily Speakman, *Algorithmic and modeling insights via volumetric comparison of polyhedral relaxations*, *Mathematical Programming Series B* **170** (2018), no. 1, 121–140.
- Ste66. P. Stein, *A note on the volume of a simplex*, *The American Mathematical Monthly* **73** (1966), no. 3, 299–301.