Technische Universität Berlin

Master's Thesis

The Computation of the Volume of the Union of Polytopes via a Sweep-Plane Algorithm

---

Author:
Lovis Anderson

Supervisor:
Prof. Dr. Thorsten Koch

Assistant Supervisor:
Dr. Benjamin Hiller

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

at the

Technische Universität Berlin
Institut für Mathematik

15. Januar 2018

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den

.....

Lovis Anderson

Zusammenfassung

Das Thema dieser Arbeit ist ein Volumen-Algorithmus für die Vereinigung von Polytopen. Der Algorithmus basiert auf der Arbeit von Bieri und Nef [BN83]. Er berechnet das Volumen der Vereinigung von Polytopen mit einem Sweep-Verfahren. Dabei wird eine Hyperebene im Raum verschoben und das Volumen auf der einen Seite der Hyperebene berechnet. Umso weiter die Hyperebene verschobe wird, desto größer ist auch der Halbraum. Unser Algorithmus berechnet das Volumen einer Vereinigung von Polytopen geschnitten mit dem Halbraum der Sweep-Ebene als eine Funktion abhängig von der Veschiebung. Ab einem gewissen Punkt liegt der Körper dabei komplett im Halbraum der Sweep-Ebene und das Volumen bleibt konstant.
Unser Algorithmus unterscheidet sich in zwei Punkten von dem Algorithmus von Bieri und Nef. Erstens funktioniert er nur auf der Vereinigung von Polytopen, wohingegen der Algorithmus von Bieri und Nef für Nef-Polyeder funktioniert. Diese sind eine Verallgemeinerung von Polyedern, die auch die Klasse der Vereinigung von Polytopen umfasst. Für uns ist das allerdings kein Nachteil, da unsere Datensätze zu Vereinigungen von Polytopen führen. Zweitens ist unser Algorithmus in zwei Teile aufgeteilt. Im ersten Teil wird eine Datenstruktur entwickelt, aus der im zweiten Teil zusammen mit einer Richtung die Sweep-Ebenen-Volumenfunktion berechnet wird. Der Großteil der Komplexität liegt im ersten Teil des Algorithmus. Das hat den Vorteil, dass wir die Volumenfunktionen für viele verschiedene Richtungen berechnen können. So können Einblicke in die Struktur des Körpers gewonnen werden.
Der Algorithmus beruht auf zwei verschiedenen Zerlegungsansätzen. Zuerst können wir mit Hilfe von *Anordnungen von Hyperebenen* eine Vereinigung von Polytopen $\mathcal{P}$ in ihre Zellen zerlegen. Dabei berufen wir uns auf die Arbeit[GH17] von Gerstner und Holtz, in der das Konzept der Positionsvektoren eingeführt wird. Diese nutzen wir um die Ecken und ihre benachbarten Zellen zu bestimmen. So erhalten wir eine Zerlegung unserer Vereinigung in Zellen, deren paarweise Schnitte kein Volumen haben. Das zweite Zerlegungskonzept ist die *konische Zerlegung*, wie sie von Lawrence in [Law] eingeführt wurde. Mit Hilfe dieser können wir die Indikatorfunktion eines Polytops als die Summe der Indikatorfunktionen seiner Vorwärtskegel schreiben. Die Sweep-Ebenen Volumenfunktionen können dann leicht mit Hilfe einer altbekannten Formel [Ste66] für das Volumen von Simplices berechnet werden.

Contents

1 Introduction

In this thesis we discuss an algorithm for the computation of the volume of the union of polytopes. The algorithm is based on the work of Bieri and Nef [BN83], who in 1983 published an algorithm for the computation of the volume of Nef polyhedra which are a generalization of polyhedra.
While there are many algorithms for the volume of polytopes and also for the volume of the union of polytopes our approach has a big upside. It calculates the volume up to a sweep-plane in a *beneath and beyond* approach. The sweep-plane is a hyperplane $H(\boldsymbol{a}, \lambda)$ given by a direction $\boldsymbol{a}$ and a time λ. While the parameter λ is increasing, the part beneath the sweep-plane, namely $H^-(\boldsymbol{a}, \lambda)$, is growing. For polytopes $P_i, i \in [k]$ and their union $\mathcal{P} := \bigcup_{i \in [k]} P_i$ the volume $\mathrm{Vol}(\mathcal{P} \cap H^-(\boldsymbol{a}, \lambda))$ beneath the sweep-plane is calculated as a function of λ. This function allows us to see interesting changes in the volume.
The main advantage of our algorithm over the algorithm of Bieri and Nef is that we separated the algorithm into two parts. In the first part a data structure is developed. Most of the computations are done while building the data structure. Together with the direction of a sweep the data structure corresponds to a decomposition of the union of polytopes, which is named the *conic decomposition*. This decomposition is used in the second part, where the sweep-plane volume function is calculated for a specific direction. Since the complexity of the calculation of the sweep-plane volume function is low, we are able to compute it for many different directions. The algorithm of Bieri and Nef does not have this separation and therefore it would be computational expensive to calculate the sweep-plane volume function for different sweep directions.

1.1 Motivation

This thesis evolved from the project "Combinatorial switching for routing gas flows" at Konrad-Zuse-Zentrum Berlin. The project is a subproject of the DFG-funded Collaborative Research Center TRR154 "Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks". The subproject is about modelling and analysing configurations for compressor stations in a gas network. To control the gas flow in a gas network one needs to select operation modes for the compressor stations. For different configurations we get different bounds on physical variables as the pressure, the mass flow and the adiabatic head. Including different configurations is leading to non-convex MINLPs. For approximations of those models, a decomposition into disjoint polytopes is desirable. The idea was to develop methods for the approximative convex decomposition (ACD) in arbitrary dimension.

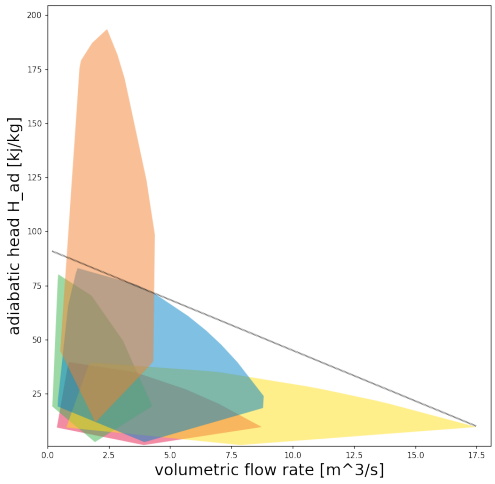


Figure 1.1: A union of polytopes and a cut.

Through the algorithm of Bieri and Nef [BN83] the idea came up to find an ACD in a *branch-and-bound* manner. The goal is to recursively decompose the body as long as the convexification error is too high. With the algorithm we are able to scan the volume through the body similar to a tomography. By scanning the volume from different directions we hope to find hyperplanes which separate the union of polytopes into two bodies which are closer to their respective convex hull. Since we could not find an implementation of the algorithm in [BN83] we decided to implement it on our own.
In order to find these cutting hyperplanes we looked at the difference of the sweep-plane volumes of the convex hull and the non-convex body. The difference is a continuously differentiable function of the sweep time λ. We looked at the maximum of its derivative. The idea behind this heuristic is that at these times the convex body diverges the most from its non-convex counterpart. If that heuristic is useful, which it seems like, it finds a good cut for a single direction. What we are doing is to compare the best cuts for many directions and choosing the one with the highest value in the derivative. The analysis is not concluded but the heuristic looks promising. Figure 1.1 shows a cut generated by this heuristic.

1.2 Literature

The central source for our work is the algorithm developed by Bieri and Nef[BN83]. In their paper a sweep-plane volume algorithm for Nef polyhedra in arbitrary dimension is presented. Nef polyhedra are a generalization of polyhedra. They are generalized in the sense that you are not only allowed to take intersections of halfspaces but also

their union and their complements. We are only interested in the class of unions of polytopes which contains the class of polytopes but is contained in the class of Nef polyhedra. Bieri and Nef's algorithm implicitly uses a decomposition concept. They give an example but a proof is missing.
However, we found the theoretical background in the paper [Law] by Lawrence. There the conic decomposition, which is the decomposition of a simple polytope into forward cones, is presented. This decomposition is based on the Brianchon-Gram formula, which has been proven independently by Brianchon and Gram for dimension 3. The conic decomposition is also described in [Haa05] and [AG16], where the decomposition is generalized to the non-simple case. Since we deal with the union of polytopes, we needed to extend the conic decomposition. This could not be done by simply summing up the decompositions of the polytopes as we would count the intersecting parts multiple times.
Bieri and Nef used a cell decomposition algorithm described in [BN82] to extend the conic decomposition to the union of polytopes. This algorithm computes all adjacent faces of a vertex recursively by the dimension. Since we are only interested in the $d-1$ dimensional faces this seemed inefficient and we decided for a different approach. We decompose the union of polytopes into non-intersecting parts and then use the conic decomposition on these parts. We are using the fact that a union of polytopes can be written as a union of cells of the underlying hyperplane arrangement. This is achieved by using the theory of hyperplane arrangements and position vectors as described by Gerstner and Holtz [GH17]. The concept of matching position vectors is used to enumerate over adjacent cells of a vertex. We extended this concept to be able to test if those cells are part of the union of polytopes.
Interestingly there is a decomposition described in [GH17], the so called *orthant decomposition*, which is a decomposition of the space into cones and is very similar to the conic decomposition. But for us the part about the enumeration of the cells and the theory of position vectors in [GH17] is more important.
To calculate the volume of the cones we used the determinant as it is described in [Ste66].
A good overview of the exact computation of the volume of polyhedra is given in [BEF00]. There the conic decomposition is discussed along several triangulation methods.
For the efficient but not exact computation of the volume of polytopes Cousins and Vempala recently published an algorithm [CV16] with a target accuracy parameter that perfoms impressively. They calculated the volume of 100 dimensional objects such as the isotropic simplex and the unit cube in minutes.
Bringmann and Friedrich developed a fully polynomial-time randomized approximation scheme [BF10] that is working in arbitrary dimension on the union and intersection of different classes of bodies, including polytopes.

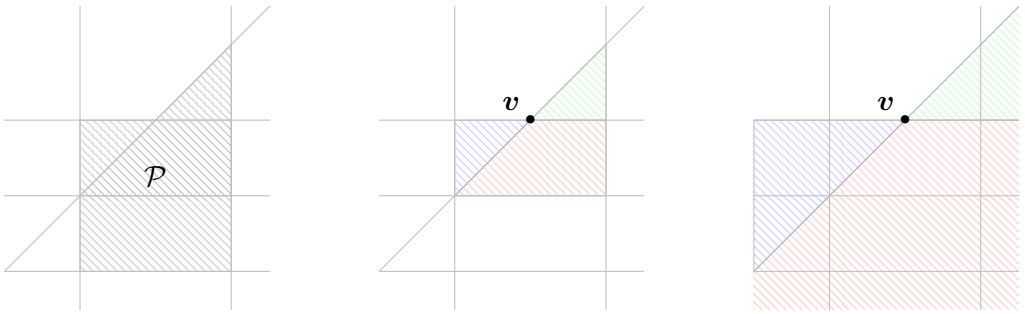
1.3 The Algorithm

In this section we are presenting a sketch of our algorithm. The algorithm is split into two parts. In the first part, the data structure is developed that will later be used to calculate the sweep-plane volume function for a specific direction.

1.3.1 Data Structure Algorithm

The data structure consists of events. An event is a tuple of a vertex and the set of the corresponding vertex cones in regularized form. Algorithm 1 gets a union of polytopes $\mathcal{P}$ as input and returns a set of events. It first computes all vertices of the underlying hyperplane arrangement. Afterwards it finds adjacent cells for each vertex and tests whether these cells belong to $\mathcal{P}$ and if so computes the respective vertex cones. A vertex cone is the intersection of halfspaces belonging to the active hyperplanes at a vertex. For a graphical example of a union of polytopes, active cells at a vertex and the respective vertex cones see Figure 1.2.

The calculation of the volume of a vertex cone is easy if the cone is simplicial. Simplicial vertex cones are the intersection of exactly dimension many halfspaces. At the vertex of a non-simplicial vertex cone more than dimensional many hyperplanes meet. Such a vertex is called degenerate. For a degenerate vertex we will introduce the regularization method, which is the topic of Section 2.3. Most of the total computational complexity lies in the data structure algorithm, Algorithm 1.



(a) The hyperplane arrangement and active cells.

(b) The active cells adjacent to $\boldsymbol{v}$.

(c) The vertex cones at $\boldsymbol{v}$ for the active cells.

Figure 1.2: The union of a triangle and a cube.

Algorithm 1 Data Structure

Input A union of polytopes $\mathcal{P}$.
Output A set of events E, whereby each event is a vertex together with its cones.

$\mathcal{H}_\mathcal{P} \leftarrow$ hyperplane arrangement belonging to $\mathcal{P}$
$E \leftarrow \emptyset$, the events
for $\boldsymbol{v}$ vertex of $\mathcal{H}$ **do**
    $\mathcal{K}(\boldsymbol{v}) \leftarrow \emptyset$, the vertex cones belonging to $\mathcal{P}$ of $\boldsymbol{v}$
    for C adjacent cell of $\boldsymbol{v}$ **do**
        if C belongs to $\mathcal{P}$ **then**
            $K \leftarrow$ the vertex cone at $\boldsymbol{v}$ that belongs to C
            if $\boldsymbol{v}$ is degenerate **then**
                $K \leftarrow$ Regularization(K)
            end if
            $\mathcal{K}(\boldsymbol{v}) \leftarrow K(\boldsymbol{v}) \cup K$
        end if
    end for
    $E \cup (v, \mathcal{K}(\boldsymbol{v}))$
end for
return E

1.3.2 Sweep-Plane Volume Algorithm

With the data structure as input we are able to calculate the sweep-plane volume function for a specific direction in Algorithm 2 by using the conic decomposition. This is a weighted decomposition of a polyhedron into its forward cones, which are created by orientating the vertex cones in the direction of the sweep. The conic decomposition will be the topic of Section 2.2. The geometry behind the conic decomposition is visualized for a simple example in Figure 1.3. Algorithm 2 first sorts the events by occurrence in the sweep and iteratively calculates the sweep-plane volume of the corresponding cones. The time λ_v at which an event with vertex $\boldsymbol{v}$ occurs on the sweep given by the direction $\boldsymbol{a}$, is given by the scalar product $\langle \boldsymbol{a}, \boldsymbol{v} \rangle$.
The sweep-plane volume functions for cones are polynomials and therefore their sum is also polynomial. The sweep-plane volume function is consequently a piecewise polynomial function. It is piecewise polynomial since events contribute to the volume only if the sweep-plane has already met them. In particular the volume can be expressed as

$$\mathrm{Vol}(P \cap H^-(\boldsymbol{a}, \lambda)) = \sum_{\substack{i \in [n] \\ \lambda_i \leq \lambda}} (\lambda - \lambda_i)^d \gamma_i. \tag{1.1}$$

Here λ_i is the time the sweep meets event e_i and $\gamma_i \in \mathbb{R}$ is the (possibly negative) growth factor of volumes of the forward cones at e_i. In Chapter 4 we will learn how to determine γ. We will also find out that we do not need to calculate the forward cones explicitly.

Algorithm 2 Sweep-Plane Volume

Input Set of events E as returned by Algorithm 1 and a vector $\boldsymbol{a}$.
Output The sweep-plane volume as a piecewise polynomial function p.

sort E by $\lambda_{\boldsymbol{v}} = \langle \boldsymbol{a}, \boldsymbol{v} \rangle$, the time at which the sweep-plane meets the vertex
$p \leftarrow []$
for $e = (\boldsymbol{v}, \mathcal{K})$ event in E **do**
    $\gamma = 0$, the growth factor of the volume of the forward cones of e
    for cone $K \in \mathcal{K}$ **do**
        calculate forward cone F of K with respect to $\boldsymbol{a}$
        calculate γ_F, the growth factor of F
        $\gamma = \gamma + \gamma_F$
    end for
    p.append($[\lambda_{\boldsymbol{v}}, \gamma]$)
end for
return p

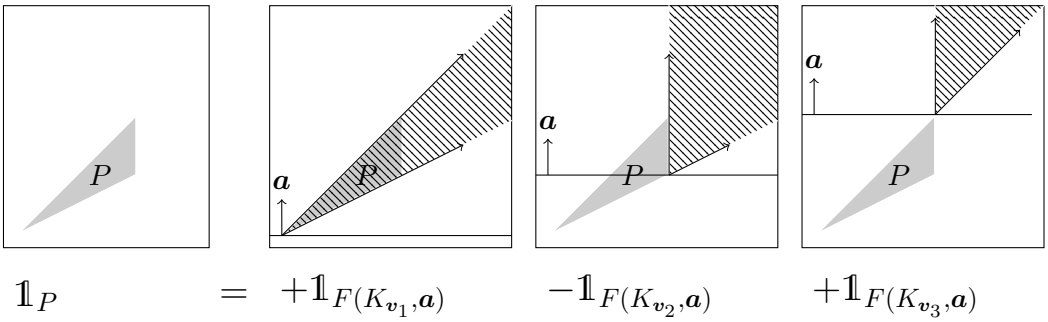


Figure 1.3: The indicator function of a polytope as weighted sum of the indicator functions of the forward cones.

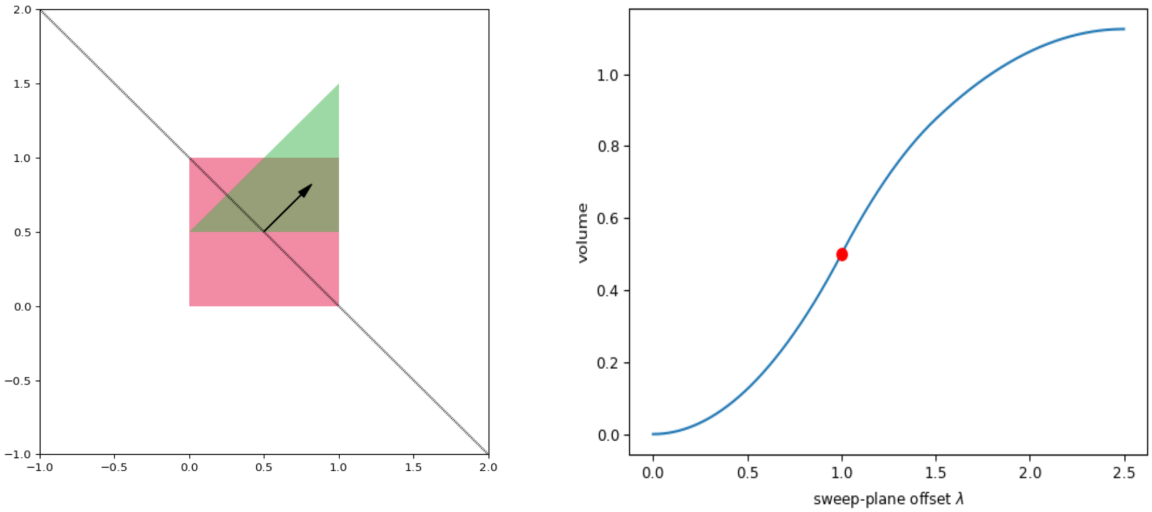


Figure 1.4: A union of polytopes and the sweep-plane with direction $\binom{1}{1}$ at $\lambda = 1$. and the corresponding volume graph.

2 Polyhedra

Our algorithm computes the volume of a union of polytopes. To understand how it works we will learn in Section 2.1 what polytopes, polyhedra and cones are and how we can describe them. We will then introduce the conic decomposition for simple polytopes in Section 2.2. In Section 2.3 we will then learn how we handle the non-simple case.

2.1 Preliminaries

We describe a hyperplane by its normal vector $\boldsymbol{a}$ and its offset λ.

Definition 2.1 For $\boldsymbol{a} \in \mathbb{R}^d \backslash \{\mathbf{0}\}$ and $\lambda \in \mathbb{R}$ the set $H(\boldsymbol{a}, \lambda) := \{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{x}, \boldsymbol{a} \rangle = \lambda\}$ is called a *hyperplane*. We use $H^-(\boldsymbol{a}, \lambda) := \{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{x}, \boldsymbol{a} \rangle \leq \lambda\}$ as the notation for the closed *negative halfspace* and $H^+(\boldsymbol{a}, \lambda) := \{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{x}, \boldsymbol{a} \rangle \geq \lambda\}$ for the closed *positive halfspace* defined by $H(\boldsymbol{a}, \lambda)$.
For the hyperplane itself we will later also use the notation $H^0(\boldsymbol{a}, \lambda) := H(\boldsymbol{a}, \lambda)$. If a point $\boldsymbol{x}$ lays on the hyperplane $H(\boldsymbol{a}, \lambda)$, that means if $\boldsymbol{x} \in H(\boldsymbol{a}, \lambda)$, we call $H(\boldsymbol{a}, \lambda)$ *active* at $\boldsymbol{x}$.

A hyperplane has different representations.

Corollary 2.2 *The positive halfspace $H^+(\boldsymbol{a}, \lambda)$ is equal to the negative halfspace $H^-(-\boldsymbol{a}, -\lambda)$.*

The notion of the positive (negative) halfspace is only useful in combination with the parameters of the hyperplane.

Definition 2.3 The finite intersection of closed halfspaces $P = \bigcap_{i \in [m]} H^{+/-}(\boldsymbol{a}_i, \lambda_i)$ is called a *polyhedron*. If P is bounded, we call it a *polytope*. The 0-dimensional faces of a polytope are called *vertices*.

We can write any polyhedra as the intersection of only negative (or only positive) halfspaces. It is just a matter of the orientation of the normal vector $\boldsymbol{a}$ as we have seen in Corollary 2.2. For the sake of simplicity we will restrict ourself to the case $P = \bigcap_{i \in [m]} H^-(\boldsymbol{a}, \lambda)$ if convenient.

Remark We can also describe a polytope P as the solutions of the corresponding linear inequality system $Ax \leq \lambda$.

Our object of interest is more general than polytopes, as it is the finite union of polytopes.

Definition 2.4 Let P_i be polytopes for $i \in [m]$. We denote $\mathcal{P} := \bigcup_{i\in[m]} P_i$ for the *union of polytopes*.

We want to describe the union of polytopes by using local properties. Later we will decompose the union of polytopes into cells which we then decompose into forward cones. To be able to do so we need the concept of a vertex cone. A vertex cone is the cone given by the halfspaces active at the vertex as in Example 2.6 and Figure 2.1.

Definition 2.5 Let $P = \bigcap_{i\in[m]} H^-(\boldsymbol{a}_i, \lambda_i)$ be a polyhedron and $\boldsymbol{v}$ be a vertex of P. Let $I_{\boldsymbol{v}} \subset [m]$ be the indices of the halfspaces active at $\boldsymbol{v}$. Then

$$K_{\boldsymbol{v}} := \bigcap_{j\in I_{\boldsymbol{v}}} H^-(\boldsymbol{a}_j, \lambda_j)$$

is *called the vertex cone of* $\boldsymbol{v}$ *at* P.

Example 2.6 We define the hyperplanes

$$H_1 := H(\begin{pmatrix}-1\\0\end{pmatrix}, 0), \qquad H_2 := H(\begin{pmatrix}0\\-1\end{pmatrix}, 0), \qquad H_3 := H(\begin{pmatrix}1\\0\end{pmatrix}, 1),$$
$$H_4 := H(\begin{pmatrix}0\\1\end{pmatrix}, 1), \qquad H_5 := H(\begin{pmatrix}0\\-1\end{pmatrix}, -0.5), \qquad H_6 := H(\begin{pmatrix}-1\\1\end{pmatrix}, 0.5).$$

Let $P_1 := H_1^- \cap H_2^- \cap H_3^- \cap H_4^-$ and $P_2 := H_3^- \cap H_5^- \cap H_6^-$. Then $\mathcal{P} = P_1 \cup P_2$ is a union of polytopes. Furthermore $\boldsymbol{v} := \begin{pmatrix}1\\\frac{1}{2}\end{pmatrix}$ is a vertex of P_2. Its vertex cone is given by $H_3^- \cap H_5^-$. See Figure 2.1.

It should be noted that in a union of polytopes a vertex can have multiple vertex cones that correspond to different polytopes. We will first decompose the union into cells, which are polyhedra and then decompose those cells into cones. We will find out that we can easily calculate the volume of a vertex cone intersected with a halfspace if this intersection is a simplex.

Definition 2.7 A vertex cone $K_{\boldsymbol{v}} \subset \mathbb{R}^d$ that can be written as $\bigcap_{i\in[d]} H^-(\boldsymbol{a}_i, \lambda_i)$ with $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d$ linear independent is called a *simplicial cone* at $\boldsymbol{v}$.
A polyhedron P is *simple* if at every vertex exactly d of the defining hyperplanes are active. We call such a vertex *non-degenerate* and a vertex at which more than d hyperplanes are active *degenerate*.

A vertex can be degenerate even if its vertex cone at a polytope is simplicial if there is a redundant halfspace or if it lies in multiple polytopes. To describe simplicial vertex cones by means of its vertex and rays the following two definitions are needed.

Definition 2.8 For two sets $A, B \subseteq \mathbb{R}^d$ the *Minkowski sum* $A + B$ is defined as $A + B := \{\boldsymbol{x} + \boldsymbol{y} | \boldsymbol{x} \in A, \boldsymbol{y} \in B\}$.

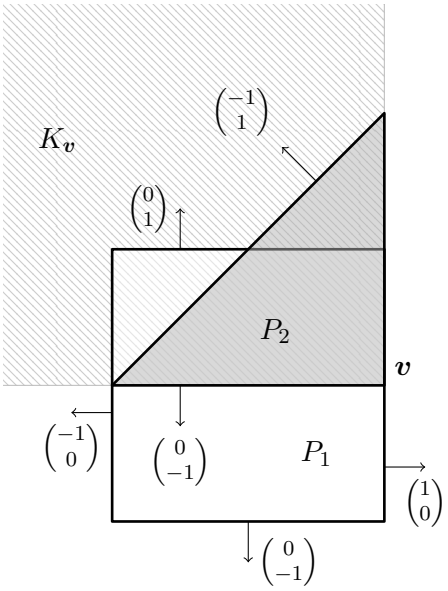


Figure 2.1: A union of the square P_1 and the triangle P_2 from Example 2.6.

Definition 2.9 Let $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m \in \mathbb{R}^d$ be vectors. The *positive hull* of those vectors is defined by $\text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m) := \{ \sum_{i \in [m]} \mu_i \boldsymbol{r}_i | \mu_i \in \mathbb{R}_{\geq 0} \}$.

The following proposition is crucial for computing the volume of the building blocks of our decomposition, the vertex cones.

Proposition 2.10 *Let $K \subset \mathbb{R}^d$ be a simplicial cone with one vertex $\boldsymbol{v}$. Let $H^-(\boldsymbol{a}_i, \lambda_i)$, $i \in [d]$ be the halfspaces that generate K.*
Let $A := \begin{pmatrix} \boldsymbol{a}_1^T \\ \vdots \\ \boldsymbol{a}_d^T \end{pmatrix}$ *and* $\boldsymbol{\lambda} := \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_d \end{pmatrix}$. *Let* $\begin{pmatrix} \boldsymbol{r}_1 & \ldots & \boldsymbol{r}_d \end{pmatrix} = R := -A^{-1}$. *Then*

$$K = \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d). \tag{2.1}$$

Proof: Clearly $K = \{\boldsymbol{x} \in \mathbb{R}^n | A\boldsymbol{x} \leq \boldsymbol{\lambda}\}$. We will prove eq. (2.1) by set inclusion.

"$\{\boldsymbol{v}\} + pos(-\boldsymbol{r}_1, \ldots, -\boldsymbol{r}_d) \subseteq \{\boldsymbol{x} \in \mathbb{R}^n | A\boldsymbol{x} \leq \boldsymbol{\lambda}\}$":
Let $\boldsymbol{x} = \boldsymbol{v} + \sum_{i \in [d]} \mu_i \boldsymbol{r}_i$. We have to show that $\boldsymbol{x} \in H^-(\boldsymbol{a}_j, \lambda_j)$ for $j \in [d]$. Because of

$AR = -I_d$ we know $\langle \boldsymbol{a}_j, \boldsymbol{r}_i \rangle = \begin{cases} 0 \text{ if } i \neq j \\ -1 \text{ if } i = j \end{cases}$. We have

$$\begin{aligned} \langle \boldsymbol{a}_j, \boldsymbol{x} \rangle &= \langle \boldsymbol{v} + \sum_{i \in [d]} \mu_i \boldsymbol{r}_i, \boldsymbol{a}_j \rangle \\ &= \langle \boldsymbol{v}, \boldsymbol{a}_j \rangle + \sum_{i \in [d]} \langle \mu_i \boldsymbol{r}_i, \boldsymbol{a}_j \rangle \\ &= \lambda_j - \mu_j. \end{aligned}$$

With $\mu \geq 0$ it follows $\boldsymbol{x} \in H^-(\boldsymbol{a}_j, \lambda_j)$ for all $j \in [d]$ and therefore $\boldsymbol{x} \in K$.

"$\{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d) \supseteq \{\boldsymbol{x} \in \mathbb{R}^n | A\boldsymbol{x} \leq \boldsymbol{\lambda}\}$":
Let $\boldsymbol{x} \in K$. Since R has full rank its columns form a basis of $\mathbb{R}^d$. Thus we can write $\boldsymbol{x} = \boldsymbol{v} + \sum_{i \in [d]} \langle \mu_i \boldsymbol{r}_i \rangle$ for some $\mu_i \in \mathbb{R}$, for $i \in [d]$. It suffices to show that $\mu_i \geq 0$ for $i \in [i]$. Then:

$$\begin{aligned} \langle \boldsymbol{x}, \boldsymbol{a}_k \rangle &= \langle \boldsymbol{v} + \sum_{i \in [d]} \mu_i \boldsymbol{r}_i, \boldsymbol{a}_k \rangle \\ &= \langle \boldsymbol{v}, \boldsymbol{a}_k \rangle + \sum_{i \in [d]} \mu_i \langle \boldsymbol{r}_i, \boldsymbol{a}_k \rangle \\ &= \lambda_k - \mu_k. \end{aligned}$$

Assuming $\mu_k < 0$ would therefore imply $\boldsymbol{x} \notin H^-(\boldsymbol{a}_k, \lambda_k)$, in contradiction to $\boldsymbol{x} \in K$. □

If we have a cone given by $K = \{v\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_s)$ we call $\boldsymbol{r}_i$ *a ray of* K for $i \in [s]$. We now have two ways to write a simplicial cone with one vertex. This also holds true for general polyhedra and both descriptions together are known as the *double description*. This term was introduced by Motzkin et al. in [MRTT53]. For our purpose the proposition above is sufficient, since we only need both descriptions for simplicial cones with a single vertex.
We define the volume as the Lebesgue measure of a set, where we denote the Lebesgue measure with μ instead of the more common λ, since we use λ as parameter for the offset of a hyperplane.

Definition 2.11 The volume of a measurable set $M \subset \mathbb{R}^d$ is defined as the Lebesgue measure μ of M, that means

$$\text{Vol}(M) := \mu(M) \tag{2.2}$$

Later it will be helpful to describe sets by their indicator function.

Definition 2.12 Let $M \subset \mathbb{R}^d$ be a measurable set. Then the indicator function

$\mathbb{1}_M : \mathbb{R}^d \to \{0, 1\}$ is defined via

$$\mathbb{1}_M(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \notin M \\ 1 & \text{if } \boldsymbol{x} \in M \end{cases}. \tag{2.3}$$

The indicator function and the volume are directly connected by the Lebesgue integral.

Corollary 2.13 *Let $M \subset \mathbb{R}^d$ be a measurable set. Then the following holds true for the volume:*

$$\mathrm{Vol}(M) = \int_{\mathbb{R}^d} \mathbb{1}_M d\mu \tag{2.4}$$

2.2 Conic Decomposition

In this section we are introducing a decomposition concept for polytopes. The characteristic of this decomposition is that the polytope can be written as a weighted sum of cones. That means that we can write the indicator function of a simple polytope P as

$$\mathbb{1}_P = \sum_{v \text{ is vertex}} (-1)^{\sigma(K_{\boldsymbol{v}}, \boldsymbol{a})} \mathbb{1}_{F(K_{\boldsymbol{v}}, \boldsymbol{a})}. \tag{2.5}$$

Here $F(K_{\boldsymbol{v}}, \boldsymbol{a})$ are cones characterized by the rays at $\boldsymbol{v}$ and the sweep-plane direction $\boldsymbol{a}$. Geometrically speaking, the rays of the vertex cones are flipped such that they are oriented along the sweep direction. The resulting cone is called the *forward cone* $F(K_{\boldsymbol{v}}, \boldsymbol{a})$. Its indicator function is part of the sum, whereby its sign depends on the number of rays that have to be flipped to convert the vertex cone into the forward cone. In this section we will proof eq. (2.5) for simple polytopes only and follow the proof that is given in [Law]. In Section 2.3 we will give a variation for the non-simple case which was proven by Haase [Haa05].
We want to define the *forward cone*. To be able to do so we first need to describe *forward* in relation to the sweep direction. The decomposition only works with sweep directions for which no ray lies in the sweep-plane.

Definition 2.14 A direction $\boldsymbol{a} \in \mathbb{R}^d$ is a *valid sweep direction* if no vertex cone defining ray $\boldsymbol{r}$ is found on the sweep-plane at the time $\lambda = 0$, which is equivalent to $\langle \boldsymbol{r}, \boldsymbol{a} \rangle \neq 0$ for all rays $\boldsymbol{r}$.

If we choose our sweep direction randomly it should always be valid.

Remark The probability that a sweep direction is not valid is 0 if it is chosen uniformly at random from $\mathbb{Q}^d$.

Definition 2.15 Let $\boldsymbol{r}$ be a ray at the vertex $\boldsymbol{v}$ and $\boldsymbol{a}$ be a valid sweep direction. Then $\boldsymbol{r}$ is *directed beyond (beneath) with respect to* $\boldsymbol{a}$ if $\langle \boldsymbol{a}, \boldsymbol{r} \rangle > 0$ ($\langle \boldsymbol{a}, \boldsymbol{r} \rangle < 0$).
For $\alpha \in \mathbb{R} \backslash \{0\}$ the ray $\alpha \boldsymbol{r}$ is called a *forward ray to* $\boldsymbol{r}$ iff it is directed beyond.

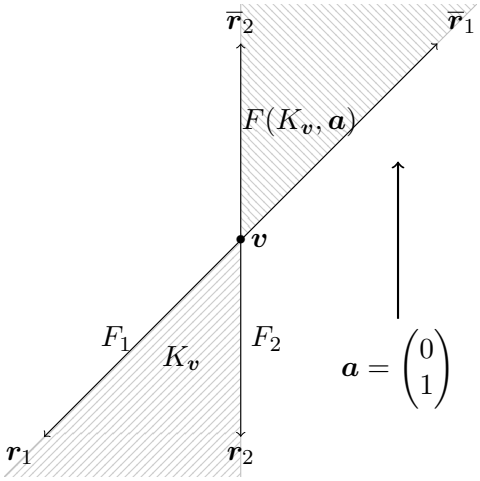


Figure 2.2: The local structure of the vertex $\boldsymbol{v}$ as in Example 2.17 and Example 2.25.

For a ray $\boldsymbol{r}$ and a sweep direction $\boldsymbol{a}$ with $\langle \boldsymbol{r}, \boldsymbol{a} \rangle \neq 0$ we can find a forward ray through $\frac{\boldsymbol{r}}{\langle \boldsymbol{a}, \boldsymbol{r} \rangle}$. Every vertex cone corresponds with a valid sweep direction to a forward cone. This cone is build from the vertex and the forward rays.

Definition 2.16 Let $K_{\boldsymbol{v}} := \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)$ be a simplicial vertex cone. Let $\boldsymbol{a}$ be a sweep direction with $\langle \boldsymbol{a}, \boldsymbol{r}_i \rangle \neq 0$ for all $i \in [d]$. Let $\overline{\boldsymbol{r}}_i := \frac{\boldsymbol{r}_i}{\langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}$ for $i \in [d]$. Then $F(K_{\boldsymbol{v}}, \boldsymbol{a}) := \{\boldsymbol{v}\} + pos(\overline{\boldsymbol{r}}_1, \ldots, \overline{\boldsymbol{r}}_d)$ is called the *forward cone of* $K_{\boldsymbol{v}}$ *with respect to* $\boldsymbol{a}$.

It does not matter which forward ray is used since through taking the positive hull the scaling is irrelevant. The following example illustrates Definition 2.16.

Example 2.17 We define the vertex $\boldsymbol{v} := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and the rays $\boldsymbol{r}_1 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$, $\boldsymbol{r}_2 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$ as well as the vertex cone $K_{\boldsymbol{v}} := \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \boldsymbol{r}_2)$. For the sweep direction $\boldsymbol{a} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ we have

$$\overline{\boldsymbol{r}_1} := \frac{\boldsymbol{r}_1}{\langle \boldsymbol{a}, \boldsymbol{r}_1 \rangle} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$\overline{\boldsymbol{r}_2} := \frac{\boldsymbol{r}_2}{\langle \boldsymbol{a}, \boldsymbol{r}_2 \rangle} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The forward cone is given by $F(K_{\boldsymbol{v}}) = \{v\} + \text{pos}(\overline{\boldsymbol{r}}_1, \overline{\boldsymbol{r}}_2)$.

As we have learned in Proposition 2.10, we can compute the rays $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d$ that span the local cone of $\boldsymbol{v}$ by taking the inverse of the matrix containing the active hyperplanes as rows. For a directions $\boldsymbol{r}_i$, $i \in [d]$ we can check whether it is a forward ray by evaluating the scalar product $\langle \boldsymbol{r}_i, \boldsymbol{a} \rangle$.

The sign of a forward cone in eq. (2.5) is defined by the number of flips that have to be performed to transfer the vertex cone into the forward cone.

Definition 2.18 Let $K_{\boldsymbol{v}} := \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)$ be a simplicial vertex cone. Let $\boldsymbol{a}$ be a valid sweep. We define the number of rays of the cone that are directed beneath as

$$\sigma(K_{\boldsymbol{v}}, \boldsymbol{a}) := |\{\boldsymbol{r}_i | \langle \boldsymbol{r}_i, \boldsymbol{a} \rangle < 0, i \in [d]\}|. \tag{2.6}$$

We are now able to state the theorem which lets us write the indicator function of our polytope as the signed sum of its forward cones as we did in eq. (2.5). This theorem and its proof are borrowed from [Law].

Theorem 2.19 *Let $\boldsymbol{a}$ be a valid sweep direction. Then for a simple polytope $P \subset \mathbb{R}^d$ with vertices $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ and vertex cones $K_1, \ldots, K_d$ it holds true that*

$$\mathbb{1}_P = \sum_{i \in [n]} (-1)^{\sigma(K_i, \boldsymbol{a})} \mathbb{1}_{F(K_i, \boldsymbol{a})}$$

To be able to prove Theorem 2.19 we need some tools. First, we define the cone generated by a polytope P at one of its faces. This is a polyhedron generated by the face and all rays that are directed into the polytope. It is a generalization of the vertex cone.

Definition 2.20 Let P be polytope and F be a non-empty face of P. Then the polyhedron generated by P at F is given by

$$\gamma(F, P) := \{\boldsymbol{z} + \alpha(\boldsymbol{y} - \boldsymbol{x}) | \boldsymbol{x}, \boldsymbol{z} \in F, \boldsymbol{y} \in P, \alpha \geq 0\}.$$

We also need the following theorem which is, according to [Law], a strengthened version of the *Brianchon-Gram Theorem* which is also known as *Gram's relation*. The proof is similar to the proof of Gram's relation given in [She67].

Theorem 2.21 *Let P be a polytope. Then*

$$\mathbb{1}_P = \sum_{\emptyset \neq G \text{ face of } P} (-1)^{dim(G)} \mathbb{1}_{\gamma(G,P)}. \tag{2.7}$$

The following lemma is a combinatorial fact that is needed for the proof of the subsequent Proposition 2.24.

Lemma 2.22 *For a set T with $|T| = n > 0$, a set $W \subseteq T$ and a natural number $d > 0$ it holds true that*

$$\sum_{W \subseteq S \subseteq T} (-1)^{d-|S|} = \begin{cases} 0 & \text{if } W \neq T \\ (-1)^{d-|W|} & \text{if } W = T \end{cases}. \tag{2.8}$$

Proof: Clearly eq. (2.8) holds true for $W = T$. For $W \neq T$ the statement is equivalent to the statement that the number of uneven sets between W and T is equal to the number of even sets between W and T. We fix an element $a \in T$ and look at the function $f : \{S \subseteq T | W \subseteq S, |S| \text{ is even } \} \to \{S \subseteq T | W \subseteq S, |S| \text{ is odd } \}$ defined by $f(S) = (S \backslash \{a\}) \cup (\{a\} \backslash S)$. Clearly f is a bijection and therefore the numbers of even and uneven sets are equal. □

In the subsequent Proposition 2.24 we prove another identity which is then used to proof Theorem 2.19. This identity allows us to write the indicator function on the forward cone at a vertex as the weighted sum of indicator functions. This sum contains the indicator functions of polyhedra generated by P at faces containing $\boldsymbol{v}$, for which the sweep meets $\boldsymbol{v}$ as the last point. These faces are characterized in the following definition.

Definition 2.23 Let $\boldsymbol{a}$ be a sweep direction, P be a polytope and $\boldsymbol{v}$ be a vertex of P. Then we define the set $\mathcal{G}_{\boldsymbol{v}} := \{G | G \text{ is a face of } P \text{ and } \arg\max_{\boldsymbol{x} \in G}\{\langle \boldsymbol{x}, \boldsymbol{a}\rangle\} = \boldsymbol{v}\}$ of faces for which $\langle \cdot, \boldsymbol{a}\rangle$ attains its maximum at $\boldsymbol{v}$.

Proposition 2.24 *Let P be a full-dimensional simple polytope, $\boldsymbol{v}$ be a vertex of P with local vertex cone $K_{\boldsymbol{v}}$ and let $\boldsymbol{a}$ be a valid sweep direction. Then it holds true that*

$$(-1)^{\sigma(K_{\boldsymbol{v}}, \boldsymbol{a})} \mathbb{1}_{F(K_{\boldsymbol{v}}, \boldsymbol{a})} = \sum_{G \in \mathcal{G}_v} (-1)^{dim(G)} \mathbb{1}_{\gamma(G,P)}. \tag{2.9}$$

Proof: Let $K_{\boldsymbol{v}} = \bigcap_{i \in [d]} H^-(\boldsymbol{a}_i, \lambda_i)$. The 2^d subsets of $[d]$ and the set of faces containing $\boldsymbol{v}$ are in bijective correspondence via

$$f(S) := \{\boldsymbol{x} \in \mathbb{R}^d | \langle \boldsymbol{a}_i, \boldsymbol{x}\rangle = \lambda_i, i \in S\} \cap P. \tag{2.10}$$

This bijection is order-reversing, that means that from $S \subseteq T$ it follows $f(T) \subseteq f(S)$. Also $\dim(f(S)) = d - |S|$. It is also true that $\gamma(f(S), P) = \{\boldsymbol{x} | \boldsymbol{x} \in H^-(\boldsymbol{a}_i, \lambda_i), i \in S\}$. It follows that $\boldsymbol{x} \in f(S)$ if and only if $S \subseteq T_{\boldsymbol{x}}$ with

$$T_{\boldsymbol{x}} := \{i \in [d] | \boldsymbol{x} \in H^-(\boldsymbol{a}_i, \lambda_i)\}. \tag{2.11}$$

Since P is simple and full-dimensional we can write

$$\boldsymbol{a} = \sum_{i \in [d]} \gamma_i \boldsymbol{a}_{\boldsymbol{v}_i}. \tag{2.12}$$

Let $W \subseteq [d]$ be the set of indices of the hyperplanes active at $\boldsymbol{v}$ for which $\gamma_i < 0$ for $i \in W$. With eq. (2.12) W is the unique smallest set for which the sweep attains its maximum in $\boldsymbol{v}$ at the corresponding face $f(W)$. So $\arg\max_{\boldsymbol{x} \in f(W)}\{\langle \boldsymbol{a}, \boldsymbol{x}\rangle\} = \boldsymbol{v}$. For every face $G \in \mathcal{G}_{\boldsymbol{v}}$ the hyperplanes $H(\boldsymbol{a}_i, \lambda_i)$ must be active for $i \in W$. That means that for any S with $f(S) \in \mathcal{G}_{\boldsymbol{v}}$ it holds true that $W \subseteq S$. With eq. (2.11) it follows that for any $\boldsymbol{x} \in \mathbb{R}^d$ the value of the right hand side of eq. (2.9) is

$$\sum_{G \in \mathcal{G}_{\boldsymbol{v}}} (-1)^{dim(G)} \mathbb{1}_{\gamma(G,P)}(\boldsymbol{x}) = \sum_{W \subseteq S \subseteq T_{\boldsymbol{x}}} (-1)^{dim(G)} \mathbb{1}_{\gamma(f(S),P)}(\boldsymbol{x}). \tag{2.13}$$

With $x \in \gamma(f(S), P)$ if and only if $S \subseteq T_{\boldsymbol{x}}$ it follows:

$$\sum_{W \subseteq S \subseteq T_{\boldsymbol{x}}} (-1)^{dim(G)} \mathbb{1}_{\gamma(f(S),P)}(\boldsymbol{x}) = \sum_{W \subseteq S \subseteq T_{\boldsymbol{x}}} (-1)^{d-|S|}. \tag{2.14}$$

With Lemma 2.22 it follows

$$\sum_{W \subseteq S \subseteq T_{\boldsymbol{x}}} (-1)^{d-|S|} = \begin{cases} 0 & \text{if } W \neq T_{\boldsymbol{x}} \\ (-1)^{d-|W|} & \text{if } W = T_{\boldsymbol{x}} \end{cases}. \tag{2.15}$$

The forward cone at F can be written as the interesection

$$F(K_{\boldsymbol{v}}, \boldsymbol{a}) = \bigcap_{i \in [d] \setminus W} H^+(\boldsymbol{a}_i, \lambda_i) \cap \bigcap_{j \in W} H^-(\boldsymbol{a}_i, \lambda_i). \tag{2.16}$$

Therefore

$$\begin{aligned} \sum_{G \in \mathcal{G}_v} (-1)^{dim(G)} \mathbb{1}_{\gamma(G,P)} &= \sum_{W \subseteq S \subseteq T_{\boldsymbol{x}}} (-1)^{d-|S|} \\ &= (-1)^{\sigma(K_{\boldsymbol{v}}, \boldsymbol{a})} \sigma(F(K_{\boldsymbol{v}}, \boldsymbol{a}))(\boldsymbol{x}). \end{aligned}$$

□

The following example helps to comprehend the geometry that is inherent to the identity from eq. (2.9).

Example 2.25 Let P be a polytope with vertices

$$\boldsymbol{v}_1 := \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \boldsymbol{v}_2 := \begin{pmatrix} 1 \\ 0.5 \end{pmatrix}, \qquad \boldsymbol{v}_3 := \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

We look at $\boldsymbol{v}_3$ and its vertex cone $K_{\boldsymbol{v}_3} := \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \boldsymbol{r}_2)$, whereby $\boldsymbol{r}_1 := \begin{pmatrix} -1 \\ -1 \end{pmatrix}$ and $\boldsymbol{r}_2 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}$. Let $\boldsymbol{a} := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ be the sweep direction. Then the forward cone $F(K_{\boldsymbol{v}_3}, \boldsymbol{a})$ is given by

$$F(K_{\boldsymbol{v}_3}, \boldsymbol{a}) = \{\boldsymbol{v}_3\} + \text{pos}(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}).$$

The local structure of P at the vertex $\boldsymbol{v}_3$ is visualized in Figure 2.2. Let F_1 (F_2) be the face at the edge corresponding to $\boldsymbol{r}_1$ ($\boldsymbol{r}_2$). The polyhedron $\gamma(F_1, P)$ ($\gamma(F_2, P)$) is given by the halfspace right (left) from the respective edge. Then, by Equation (2.9), it holds true that

$$\begin{aligned} \mathbb{1}_{F(K_v, \boldsymbol{a})} &= \mathbb{1}_{\gamma(\boldsymbol{v}, P)} && - \mathbb{1}_{\gamma(F_1, P)} - \mathbb{1}_{\gamma(F_2, P)} && + \mathbb{1}_{\gamma(P, P)} \\ &= \mathbb{1}_{K_v} && - \mathbb{1}_{\gamma(F_1, P)} - \mathbb{1}_{\gamma(F_2, P)} && + \mathbb{1}_{\mathbb{R}^2}. \end{aligned}$$

For the conic decomposition of P look at Figure 2.3.

We are now able to prove Theorem 2.19.

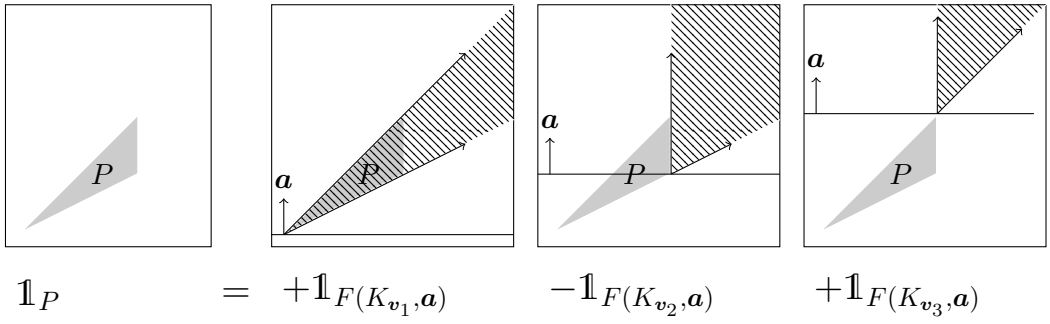


Figure 2.3: The conic decomposition of a polytope. See example 2.25.

Proof: With eq. (2.9) we have

$$\sum_{\substack{\boldsymbol{v}\\ \text{vertex of P}}} (-1)^{\sigma(K_{\boldsymbol{v}},\boldsymbol{a})}\mathbb{1}_{F(K_{\boldsymbol{v}},\boldsymbol{a})} = \sum_{\substack{\boldsymbol{v}\\ \text{vertex of P}}} \sum_{G\in\mathcal{G}_{\boldsymbol{v}}} (-1)^{\sigma(K_{\boldsymbol{v}},\boldsymbol{a})}\mathbb{1}_{F(K_{\boldsymbol{v}},\boldsymbol{a})}. \tag{2.17}$$

Since every bounded face is contained in exactly one $\mathcal{G}_{\boldsymbol{v}}$ it holds true that

$$\sum_{\substack{\boldsymbol{v}\\ \text{vertex of P}}} \sum_{G\in\mathcal{G}_{\boldsymbol{v}}} (-1)^{\sigma(K_{\boldsymbol{v}},\boldsymbol{a})}\mathbb{1}_{F(K_{\boldsymbol{v}},\boldsymbol{a})} = \sum_{\substack{G \text{ bounded}\\ \text{face of } P}} (-1)^{dim(G)}\mathbb{1}_P. \tag{2.18}$$

With Theorem 2.21 it then follows

$$\sum_{\substack{G \text{ bounded}\\ \text{face of } P}} (-1)^{dim(G)}\mathbb{1}_{\gamma(G,P)} = \mathbb{1}_{\gamma(G,P)}. \tag{2.19}$$

□

Since the volume of a measurable set can be described as the integral over the indicator function as in Corollary 2.13, the following corollary is a direct consequence of Theorem 2.19.

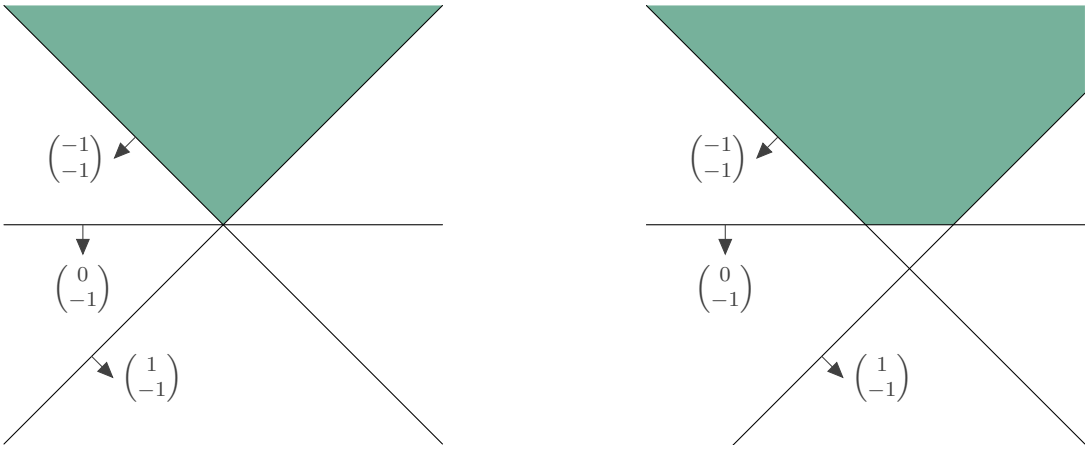
Corollary 2.26 *Let P be a simple polytope with vertices $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n$ and $\boldsymbol{a}$ be a valid sweep direction. Then*

$$\operatorname{Vol}(P\cap H^-(\boldsymbol{a},\lambda)) = \sum_{i=1}^{n}(-1)^{\sigma(K_{\boldsymbol{v}_i},\boldsymbol{a})}\operatorname{Vol}(F(K_{\boldsymbol{v}_i},\boldsymbol{a})\cap H^-(\boldsymbol{a},\lambda)).$$

2.3 Regularization of Degenerate Vertices

In this section we will develop a method which reduces a degenerate vertex $\boldsymbol{v}$ to the non-degenerate case. We will find simple polyhedra $P(\varepsilon)$ which converge to the vertex cone $K_{\boldsymbol{v}}$ as $\varepsilon \to 0$ with respect to the Hausdorff metric. That means that we can handle the non-simple case locally and do the regularization for each vertex cone independently.
The idea and technique of the regularization process is borrowed from [BN83], where it is described, but not formalized. In this chapter we will develop a formalization of the regularization. We will build a new polyhedron by starting with a cone defined by d of the active hyperplanes at $\boldsymbol{v}$. We will then consecutively shift the other hyperplanes, such that the previous polyhedron is completely contained in the shifted open halfspace.
There are other ways to handle the degenerate case, such as decomposing a non-simplicial cone into multiple simplicial cones. Our approach is similar, as we have only simplicial cones in our generated polyhedron.

Example 2.27 Let $H_1 := H(\begin{pmatrix} 0 \\ -1 \end{pmatrix}, 0)$, $H_2 := H(\begin{pmatrix} -1 \\ -1 \end{pmatrix}, 0)$, $H_3 := H(\begin{pmatrix} 1 \\ -1 \end{pmatrix}, 0)$ and $K := \bigcap_{i=1}^{3} H_i^-$. Let $H_3(\varepsilon) := H(\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \varepsilon)$ with $\varepsilon > 0$. We look at $K(\varepsilon) := H_1^- \cap H_2^- \cap H_3^-(\varepsilon)$ and observe that it is the result of shifting the hyperplane in direction of its normal vector by ε. It is easy to see that $\lim_{\varepsilon \to 0} K(\varepsilon) = K$ holds true. In Figure 2.4 the shift for $\varepsilon = 1$ is depicted.



(a) The polyhedron K defined by H_1^-, H_2^-, H_3^-.

(b) The polyhedron $K(1)$ defined by $H_1^-, H_2^-, H_3^-(1)$.

Figure 2.4: A polytope and the polytope that results from shifting a hyperplane outwards. See Example 2.27.

Definition 2.28 Let $\boldsymbol{v}$ be a vertex with the vertex cone $K_{\boldsymbol{v}}$ generated by the halfspaces $H^-(\boldsymbol{a}_1, \lambda_1), \ldots, H^-(\boldsymbol{a}_m, \lambda_m)$. Let $\Omega := \{\omega_1, \ldots, \omega_m\} \subset \mathbb{R}_{\geq 0}$ be a set of positive real numbers. Let $\varepsilon > 0$. Then $H_i(\omega_i, \varepsilon) := H(\boldsymbol{a}_i, \lambda_i + \varepsilon\omega_i)$ is the by $\omega_i\varepsilon$ *shifted hyperplane*. For better readibility we will omit ω_i and write $H_i(\varepsilon)$ if the set Ω is already defined. $P(\Omega, \varepsilon) := \bigcap_{i\in[m]} H_i^-(\omega_i, \varepsilon)$ defines a polyhedron for each $\varepsilon > 0$.

All hyperplanes are shifted by the same factor ε, which serves as a scaling parameter for the shift. Later we see that the structure of the polyhedron is independent of the scaling.

Corollary 2.29 *Let $\boldsymbol{v}$ be a vertex with vertex cone $K_{\boldsymbol{v}} := \bigcap_{i\in[m]} H^-(\boldsymbol{a}_i, \lambda_i)$. Let $\Omega := \{\omega_1, \ldots, \omega_m\} \subset \mathbb{R}_{\geq 0}$. It holds true that $\lim_{\substack{\varepsilon\to 0\\ \varepsilon>0}} P(\Omega, \varepsilon) = K_{\boldsymbol{v}}$.*
Furthermore let $H^-(\boldsymbol{a}, \lambda)$ be a sweep-plane with $\boldsymbol{v} \in H^-(\boldsymbol{a}, \lambda)$. Then

$$\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = \lim_{\substack{\varepsilon\to 0\\ \varepsilon>0}} \mathrm{Vol}(P(\Omega, \varepsilon) \cap H^-(\boldsymbol{a}, \lambda)).$$

Proof: The shifted polyhedrons clearly converge to the original cone for $\varepsilon \to 0$ with respect to the Haussdorf metric. With $P(\Omega, \varepsilon_1) \subset P(\Omega, \varepsilon_2)$ for $\varepsilon_1 \leq \varepsilon_2$ and the Lebesgue measure being continuous from above the volume converges as well. □

For all $\varepsilon > 0$ the incidence structure of the polyhedron $P(\Omega, \varepsilon)$ generated by a shift Ω is the same. For our purpose the slightly weaker following proposition is sufficient.

Proposition 2.30 *Let $\boldsymbol{v}$ be a vertex with vertex cone $K_{\boldsymbol{v}} := \bigcap_{i\in[m]} H^-(\boldsymbol{a}_i, \lambda_i)$. Let $\Omega := \{\omega_1, \ldots, \omega_m\} \subset \mathbb{R}_{\geq 0}$ be a set of positive shifting parameters. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l$ be the vertices of $P(\Omega, 1)$. Then the vertices of $P(\Omega, \varepsilon)$ are given by $\boldsymbol{v}_i(\varepsilon) := \boldsymbol{v} + \varepsilon(\boldsymbol{v}_i - \boldsymbol{v})$ for $i \in [l]$ and $\varepsilon > 0$.*
Let $H_{\boldsymbol{v}_{i_1}}(1), \ldots, H_{\boldsymbol{v}_{i_m}}(1)$ be the hyperplanes active at $\boldsymbol{v}_i$. Then precisely the hyperplanes $H_{\boldsymbol{v}_{i_1}}(\varepsilon), \ldots, H_{\boldsymbol{v}_{i_m}}(\varepsilon)$ are active at $\boldsymbol{v}_i(\varepsilon)$. The cone $K_{\boldsymbol{v}_i}(\varepsilon) := \bigcap_{j\in[m]} H_{i_j}(\varepsilon)$ can be described as the Minkowski sum $\boldsymbol{v}_i(\varepsilon) + K_{\boldsymbol{0}}^{v_i}$ for $i \in [l]$ whereby $K_{\boldsymbol{0}}^{v_i}$ is a cone with $\boldsymbol{0}$ as its only vertex which is independent of ε.

Proof: Let $\boldsymbol{v}_j$ be a vertex of $P(\Omega, 1)$ and $I_{\boldsymbol{v}_j} \subset [m]$ be the indices of the active hyperplanes at $\boldsymbol{v}_j$. That means $\boldsymbol{v}_j \in H_i(1)$ if and only if $i \in I_{\boldsymbol{v}_j}$.
We want to show that $\boldsymbol{v}_j(\varepsilon) = \boldsymbol{v} + \varepsilon(\boldsymbol{v}_j - \boldsymbol{v})$ is a vertex of $P(\Omega, \varepsilon)$. First we show $\boldsymbol{v}_j(\varepsilon) \in H_i(\varepsilon)$ for $i \in I_{\boldsymbol{v}_j}$. This follows from $\langle \boldsymbol{a}_i, \boldsymbol{v}_j\rangle = \lambda_i + \omega_i$ and $\langle \boldsymbol{a}_i, \boldsymbol{v}\rangle = \lambda_i$:

$$\langle \boldsymbol{a}_i, (\boldsymbol{v} + \varepsilon(\boldsymbol{v}_j - \boldsymbol{v}))\rangle = \lambda_i + \varepsilon(\lambda_i + \omega_i - \lambda_i). \tag{2.20}$$

The same argument works if we have a strict inequality in eq. (2.20). That means for $i \notin I_{\boldsymbol{v}_j}$ it holds

$$\langle \boldsymbol{a}_i, (\boldsymbol{v} + \varepsilon(\boldsymbol{v}_j - \boldsymbol{v}))\rangle < \lambda_i + \varepsilon(\lambda_i + \omega_i - \lambda_i). \tag{2.21}$$

Therefore $\boldsymbol{v}_j(\varepsilon)$ is a vertex of $P(\Omega, \varepsilon)$ and the same hyperplanes are active at $\boldsymbol{v}_j(\varepsilon)$ as at $\boldsymbol{v}_j$.
We need to prove that $\boldsymbol{v}_1(\varepsilon), \cdots, \boldsymbol{v}_l(\varepsilon)$ are all vertices of $P(\Omega, \varepsilon)$. Let $\boldsymbol{w}$ be a vertex of $P(\Omega, \varepsilon)$. We define $\overline{\boldsymbol{w}} := \frac{\boldsymbol{w}-v}{\varepsilon} + v$ and with the same argument as in 2.20 we see that $\overline{\boldsymbol{w}}$ is a vertex of $P(\Omega, 1)$. Since $\overline{\boldsymbol{w}}(\varepsilon) = \boldsymbol{w}$, all vertices of $P(\Omega, \varepsilon)$ correspond to vertices in $P(\Omega, 1)$ and vice versa.
Let $K_{\boldsymbol{v}_i}(\varepsilon) := \bigcap_{j \in [m]} H_{i_j}(\varepsilon)$ be the cone at the vertex $\boldsymbol{v}_i(\varepsilon)$. We look at the linear inequality system $A\boldsymbol{x} \leq \boldsymbol{b}$ which is defined by the hyperplanes active at $\boldsymbol{v}_i(\varepsilon)$. Then for $\boldsymbol{x} \in K_{\boldsymbol{v}_i}(\varepsilon)$ it holds true that

$$\begin{aligned} A\boldsymbol{x} &\leq \boldsymbol{b} \\ A\boldsymbol{x} &\leq A\boldsymbol{v}_i(\varepsilon) \\ A(\boldsymbol{x} - \boldsymbol{v}_i(\varepsilon)) &\leq \boldsymbol{0}. \end{aligned}$$

Let $K_{\boldsymbol{0}}^{\boldsymbol{v}_i} := \{\boldsymbol{x} | A\boldsymbol{x} \leq \boldsymbol{0}\}$ which is clearly a cone with $\boldsymbol{0}$ as its only vertex. We can then write $K_{\boldsymbol{v}_i}(\varepsilon) = \boldsymbol{v}_i(\varepsilon) + K_{\boldsymbol{0}}^{\boldsymbol{v}_i}$. □

We found out that our local vertex cones are, up to translation, independent of the factor ε by which the shifting is scaled. We will now find an Ω for which the corresponding shifted polytope $P(\Omega, 1)$ is simple. With Proposition 2.30 it follows that $P(\Omega, \varepsilon)$ is simple for $\varepsilon > 0$. We start our construction by choosing d many hyperplanes active at $\boldsymbol{v}$ with linear independent normal vectors. From there we iteratively shift the hyperplanes in such a way that all old vertices are in the open shifted halfspace and therefore are left untouched. Furthermore we will show that new vertices that occur by shifting a hyperplane are non-degenerate.

Proposition 2.31 *Let $\boldsymbol{v}$ be a vertex with vertex cone $K_{\boldsymbol{v}} := \bigcap_{i \in [m]} H^-(\boldsymbol{a}_i, \lambda_i)$. Without loss of generality let $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d$ be linear independent and let $P_d := \bigcap_{i \in [d]} H^-(\boldsymbol{a}_1, \lambda_1)$. Then P_d is a simplicial cone.*
For $j \in \{d, m-1\}$ let $P_{j+1} := \bigcap_{i \in [j]} H^-(\boldsymbol{a}_1, \lambda_1) \cap H_{j+1}^-$ with

$$H_{j+1} := H(\boldsymbol{a}_{j+1}, \lambda_{j+1} + \omega_{j+1})$$

and ω_{j+1} chosen in such a way that all vertices of P_j are contained in the open halfspace $\mathring{H}_{j+1}^-$. Then all vertices of P_m are non-degenerate.

Proof: Since P_d is the intersection of exactly d many hyperplanes, $\boldsymbol{v}$ is the only degenerate vertex of P_d.
We will show by induction that all vertices of P_m are non-degenerate. P_d only has one vertex which is non-degenerate. We assume that P_j has only non-degenerate vertices and look at P_{j+1}.
For any vertex $\boldsymbol{v}$ of P_j it holds true that it is a vertex of P_{j+1} since the halfspace H_{j+1}^- is constructed in such a way that $\boldsymbol{v}$ is in its interior. For any new vertex $\tilde{\boldsymbol{v}}$ of P_{j+1} it

holds true that at maximum $d-1$ hyperplanes of P_j can be active at $\tilde{\boldsymbol{v}}$. Otherwise $\tilde{\boldsymbol{v}}$ would already be a vertex of P_j or P_j would be not simple. Only one new hyperplane is added and at least d hyperplanes have to be active at $\tilde{\boldsymbol{v}}$ for it to be a vertex. That means exactly d hyperplanes are active at $\tilde{\boldsymbol{v}}$ and therefore $\tilde{\boldsymbol{v}}$ is non-degenerate. □

We are now able to construct a simple polytope that converges to our vertex cone iteratively.

Definition 2.32 Let $K_{\boldsymbol{v}} = \bigcap_{i\in[m]} H^-(\boldsymbol{a}_i, \lambda_i)$ be a vertex cone. Let $\Omega = \{\omega_1, \dots, \omega_m\}$ be a set of shifting parameters according to Proposition 2.31. Then we call $P(\Omega, 1)$ a *regularization* of $K_{\boldsymbol{v}}$. Let $K_1(1), \dots, K_l(1)$ be the vertex cones of $P((\Omega, 1)$ with $K_i(1) = \{\boldsymbol{v}_i\} + \text{pos}(\boldsymbol{r}_{i_1}, \dots, \boldsymbol{r}_{i_d})$. The *regularized vertex cones* for the vertex are then given by $K_i := \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_{i_1}, \dots, \boldsymbol{r}_{i_d})$.

The following theorem generalizes the conic decomposition from Theorem 2.19 to non-simple polytopes. The generalization is described in [Haa05] and [AG16]. The theorem states that we can regularize a polytope by regularizing degenerate vertices locally.

Theorem 2.33 *Let $P \subset \mathbb{R}^d$ be a polytope with vertices $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$. Let $P_{\boldsymbol{v}_i}(\Omega, 1)$ be a regularization of the vertex cone $K_{\boldsymbol{v}_i}$ for $i \in [n]$. Let $K_{\boldsymbol{v}_i}^{(1)}, \dots, K_{\boldsymbol{v}_i}^{(m_i)}$ be the regularized vertex cones at $\boldsymbol{v}_i$. Then*

$$\mathbb{1}_P = \sum_{i\in[n]} \sum_{j\in[m_i]} (-1)^{\sigma(K_{\boldsymbol{v}_i}^{(j)}, \boldsymbol{a})} \mathbb{1}_{F(K_{\boldsymbol{v}_i}^{(j)}, \boldsymbol{a})}. \tag{2.22}$$

2.3.1 The Algorithm

Algorithm 3 first calculates a simple polyhedron $P(\Omega, 1)$ as in Proposition 2.31. Let $\boldsymbol{v}_1, \dots, \boldsymbol{v}_l$ be the vertices of $P(\Omega, 1)$ and $K_{\boldsymbol{v}_i}(\varepsilon) = \{\boldsymbol{v}_i(\varepsilon)\} + \text{pos}(\boldsymbol{r}_{i_1}, \dots, \boldsymbol{r}_{i_d})$ be the vertex cone at $\boldsymbol{v}_i$ for $i \in [l]$. With Proposition 2.30 it holds that

$$\lim_{\varepsilon\to 0} K_{\boldsymbol{v}_i}(\varepsilon) = \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_{i_1}, \dots, \boldsymbol{r}_{i_d}). \tag{2.23}$$

So we can calculate the vertex cones for $P(\Omega, 1)$ and then replace their vertices by $\boldsymbol{v}$.

Remark The vertices of P can be calculated by using the double description method as first introduced in [MRTT53]. A fast implementation is described by Fukuda and Prodon in [FP96]. We used the corresponding python package `pycddlib 2.0.0`.

In Example 2.27 we have looked at a degenerate vertex in dimension 2. Even though the vertex is degenerate the vertex cone can be described as the intersection of 2 halfspaces and is therefore simplicial. This is always the case in dimension 2. In Example 2.34 we look at the non-simplicial case, where the degeneracy is not given by a redundant halfspace.

Algorithm 3 Regularization

Input A cone $K = \bigcap_{i\in[m]} H^-(\boldsymbol{a}_i, \lambda_i)$ with vertex $\boldsymbol{v}$.

Output A set $\mathcal{K}$ of vertex cones in regularized form.

$\mathcal{H} \leftarrow \{H^-(\boldsymbol{a}_i, \lambda_i) | i \in [n]\}$
sort $\{H^-(\boldsymbol{a}_i, \lambda_i), i| \in [s]\}$ such that $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_d$ are linear independent
$P \leftarrow \bigcap_{j\in[d+1]} H_j$
$i \leftarrow d+1$
while $i \leq m$ **do**
    $\mathcal{V} \leftarrow$ vertices of P
    $\omega_i = \max\{\langle \boldsymbol{a}_i, \boldsymbol{v}\rangle | \boldsymbol{v} \in \mathcal{V}\} + 1$
    replace $H^-(\boldsymbol{a}_i, \lambda_i)$ with $H^-(\boldsymbol{a}_i, \lambda_i + \omega_i)$ in $\mathcal{H}$
    $P \leftarrow \bigcap_{j\in[i]} H_j$
    $i = i + 1$
end while
$\mathcal{K} \leftarrow \emptyset$
for $K_i = \{\boldsymbol{v}_i(1)\} + \text{pos}(\boldsymbol{r}_{i_1}, \ldots, \boldsymbol{r}_{i_d})$ vertex cone of P **do**
    $\mathcal{K}$.add$((\boldsymbol{v}, (\boldsymbol{r}_{i_1}, \ldots, \boldsymbol{r}_{i_d})))$
end for
return $\mathcal{K}$

Example 2.34 Let

$$H_1 := H(\begin{pmatrix}-1\\0\\-1\end{pmatrix}, 0), \qquad H_2 := H(\begin{pmatrix}0\\-1\\-1\end{pmatrix}, 0),$$
$$H_3 := H(\begin{pmatrix}0\\1\\-1\end{pmatrix}, 0), \qquad H_4 := H(\begin{pmatrix}1\\0\\-1\end{pmatrix}, 0).$$

We look at $K = \bigcap_{i\in[4]} H_i^-$. $\boldsymbol{v} = \begin{pmatrix}0\\0\\0\end{pmatrix}$ is the only (degenerate) vertex of K. The normal vectors of H_1, H_2 and H_3 are linear independent. Therefore we start with $P_3 = H_1^- \cap H_2^- \cap H_3^-$. P_3 is a simplicial cone with $\boldsymbol{v}$ as vertex. We set

$$\omega_i := 0, i \in [3] \quad \text{and} \quad \omega_4 := \langle \begin{pmatrix}1\\0\\-1\end{pmatrix}, \begin{pmatrix}0\\0\\0\end{pmatrix} \rangle + 1 = 1.$$

Let $\Omega := \{\omega_1, \omega_2, \omega_3, \omega_4\}$. We look at $P(\Omega, \varepsilon) = \bigcap_{i\in[4]} H_i^-(\omega_i, \varepsilon)$, for which the vertices

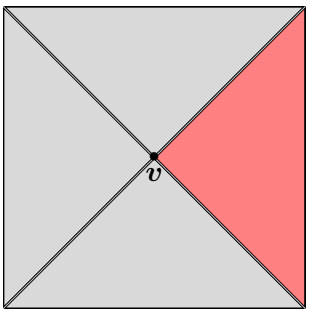
are given by $\boldsymbol{v}(\varepsilon) = H_1 \cap H_2 \cap H_3 = \boldsymbol{v}$ and $\boldsymbol{w}(\varepsilon) = H_2 \cap H_3 \cap H_4(\omega_4, \varepsilon) = \begin{pmatrix} \varepsilon \\ 0 \\ 0 \end{pmatrix}$. By Proposition 2.10 the vertex cones of $P(\Omega, \varepsilon)$ are given by

$$K_{\boldsymbol{v}}(\varepsilon) = \boldsymbol{v} + \text{pos}(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.5 \\ 0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} -0.5 \\ -0.5 \\ 0.5 \end{pmatrix})$$
$$K_{\boldsymbol{w}}(\varepsilon) = \boldsymbol{w}(\varepsilon) + \text{pos}(\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5 \\ -0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix}).$$

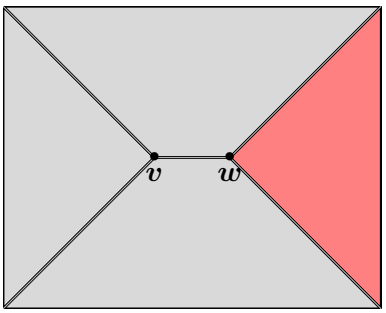
Algorithm 3 returns

$$K_1 := (\boldsymbol{v}, (\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -0.5 \\ 0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} -0.5 \\ -0.5 \\ 0.5 \end{pmatrix})) \quad \text{and} \quad K_2 := (\boldsymbol{v}, (\begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.5 \\ -0.5 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \end{pmatrix})).$$

K and $P(\Omega, 1)$ are depicted in Figure 2.5.



(a) The "top view" of cone K at the vertex $\boldsymbol{v}$.



(b) The "top view" of the polyhedron $P(\Omega, 1)$.

Figure 2.5: A non-simplicial vertex cone and the regularized polytope $P(\Omega, 1)$ that is the result of shifting the red hyperplane by 1. See Example 2.34.

3 The Data Structure

In this chapter we will develop methods to calculate the relevant vertices of a union of polytopes and their vertex cones. If we have Polytopes $P_1, \dots, P_k$ and their union $\mathcal{P} := \bigcup_{i\in[k]} P_i$. The hyperplanes that define those polytopes separate the space into different cells. $\mathcal{P}$ is a subset of those cells. The indicator function of a cell can be written as the weighted sum of the indicator functions of its forward cones by Theorem 2.19. We can extend this to the sum of the indicator functions of the cells. First we determine all vertices of the hyperplane arrangement, then we test which cells are part of $\mathcal{P}$ and compute the corresponding vertex cones.

3.1 Hyperplane Arrangements

Definition 3.1 [S$^+$04] A finite set of hyperplanes $\mathcal{H} := \{H(\boldsymbol{a}_1, \lambda_1), \dots, H(\boldsymbol{a}_m, \lambda_m)\}$ is called a *hyperplane arrangement*. We call $\mathcal{H}$ *non-degenerate* if every vertex of $\mathcal{H}$ is non-degenerate.

The defining hyperplanes of a union of polytopes are a hyperplane arrangement.

Definition 3.2 Let $P_i = \bigcap_{j\in I_i} H^{+/-}(\boldsymbol{a}_j, \lambda_j)$ be polytopes with $I_i \subset \mathbb{N}$ and $|I_i| < \infty$, for $i \in [r]$. For their union $\mathcal{P} := \bigcup_{i\in[r]} P_i$ we define

$$\mathcal{H}_\mathcal{P} := \{H(\boldsymbol{a}_j, \lambda_j) | j \in \bigcup_{i\in[r]} I_i\}$$

as the $\mathcal{P}$ *underlying hyperplane arrangement*.

For the union of polytopes $\mathcal{P}$ and the corresponding hyperplane arrangement $\mathcal{H}_\mathcal{P}$ it holds true that $\mathcal{P}$ is a subset of the cells of $\mathcal{H}$. Also every vertex of $\mathcal{P}$ has to be a vertex of $\mathcal{H}$. A special role is taken by the full dimensional faces of a hyperplane arrangement.

Definition 3.3 For a hyperplane arrangement $\mathcal{H}$ the closures of the connected components of the set $R = \mathbb{R}^d \setminus \bigcup_{H\in\mathcal{H}} H^0$ are called *cells*.

A hyperplane arrangement splits the space into different cells. We want to find characteristics that are decisive when a cell of $\mathcal{H}_\mathcal{P}$ is part of $\mathcal{P}$. The following combinatorial description lets us write any polyhedron that can be found in the hyperplane arrangement as a vector.

Definition 3.4 Let $\mathcal{H} ::= \{H(\boldsymbol{a}_1, \lambda_1), \ldots, H(\boldsymbol{a}_m, \lambda_m)\}$ be a hyperplane arrangement. We call a vector $\begin{pmatrix} p_1 \\ \vdots \\ p_m \end{pmatrix} =: \boldsymbol{p} \in \{+, -, 0, \bullet\}^m$ which corresponds to the set $\mathcal{H}(\boldsymbol{p}) := \{\boldsymbol{x} | \boldsymbol{x} \in H_i^{p_i}$ for i $\in [m]$ and $p_i \neq \bullet\}$ a *position vector*.

Remark Definition 3.4 is a slight modification of the common definition of a position vector, e.g. given in [FP08]. There the position vectors are from a smaller set, namely $\boldsymbol{p} \in \{+, -, 0\}^k$, which corresponds to the set $\mathcal{H}(\boldsymbol{p}) := \{\boldsymbol{x} | \boldsymbol{x} \in H_i^{p_i}$ for i $\in [k]\}$. With our definition, we can describe any polyhedron that can be described with a subset of the given hyperplanes. That allows us to describe vertex cones and polytopes that span multiple cells of the hyperplane arrangement.
By allowing a set given by a position vector to be independent from a hyperplane, the position vector is not necessarily unique for a set.

Definition 3.5 Let $\mathcal{H} := \{H(\boldsymbol{a}_1, \lambda_1), \ldots, H(\boldsymbol{a}_m, \lambda_m)\}$ be a hyperplane arrangement and $\boldsymbol{p}$ be a position vector.
If for every $i \in [m]$ with $\boldsymbol{p}_i = \bullet$ it holds true that $\mathcal{H}(\boldsymbol{p}) \nsubseteq H^+(\boldsymbol{a}_i, \lambda_i)$ and $\mathcal{H}(\boldsymbol{p}) \nsubseteq H^-(\boldsymbol{a}_i, \lambda_i)$, we call $\boldsymbol{p}$ the *exact position vector* of the set $\mathcal{H}(\boldsymbol{p})$.

Clearly for every set that corresponds to a position vector there is an exact position vector. Cells are the intersection of halfspaces and vertices are the intersection of at least d many hyperplanes. As a consequence their exact position vectors have certain characteristics.

Corollary 3.6 *Let $\mathcal{H} = \{H(\boldsymbol{a}_1, \lambda_1), \ldots, H(\boldsymbol{a}_m, \lambda_m)\}$ be a hyperplane arrangement. Let C be a cell of $\mathcal{H}$. Then the exact position vector $\boldsymbol{p}(C)$ has only $+$ and $-$ entries, i.e. $\boldsymbol{p}(C) \in \{+, -\}^m$.*
The exact position vector $\boldsymbol{p}(\boldsymbol{v})$ of a vertex $\boldsymbol{v}$ of $\mathcal{H}$ has at least d many 0 entries. Furthermore $\boldsymbol{p}(\boldsymbol{v}) \in \{+, 0, -\}^m$.

Since we want to find a decomposition of a union of polytopes into cells of the corresponding hyperplane arrangement, we are looking for characteristics of cells that are part of that union. Position vectors help us to analyze the incidence structure of vertices, vertex cones, cells and polytopes.

Definition 3.7 We say that two position vectors $\boldsymbol{p}^1, \boldsymbol{p}^2 \in \{+, -, 0, \bullet\}^m$ *match*, if for all $i \in [m]$ one of the following holds true:

(i) $p_i^1 = p_i^2$

(ii) $p_i^1 \in \{\bullet, 0\}$

(iii) $p_i^2 \in \{\bullet, 0\}$

We need to differ between the cells that are part of the union of polytopes and those which are not.

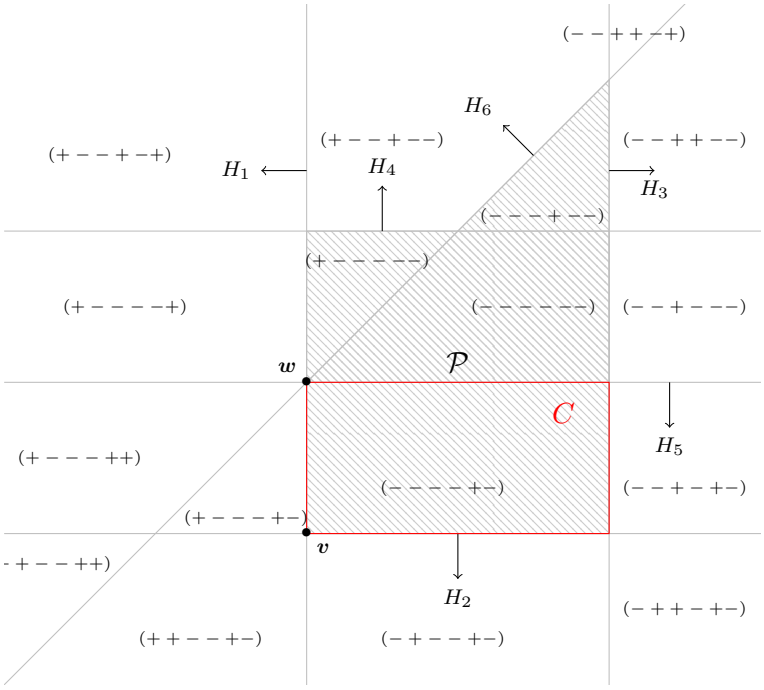


Figure 3.1: The hyperplane arrangement of the union of a square and a triangle with the cell describing position vectors. See Example 3.9 and Example 3.15.

Definition 3.8 For the union of polytopes $\mathcal{P}$ and the corresponding hyperplane arrangement $\mathcal{H}_\mathcal{P} = \{H(\boldsymbol{a}_1, \lambda_1), \ldots, H(\boldsymbol{a}_m, \lambda_m)\}$ and a position vector $\boldsymbol{p}$ with $\mathcal{H}(\boldsymbol{p}) \neq \emptyset$ we call $\boldsymbol{p}$ *active at* $\mathcal{P}$ or short *active* if $\mathcal{H}(\boldsymbol{p}) \subseteq \mathcal{P}$.

Example 3.9 We return to Example 2.6. The $\mathcal{P}$ underlying hyperplane arrangement $\mathcal{H}_\mathcal{P}$ and the position vectors of the cells are shown in Figure 3.1.

We look at the vertex $\boldsymbol{v_2} := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. The exact position vector of $\boldsymbol{v}$ is given by $\boldsymbol{p}(\boldsymbol{v}) = (00 - - + -)$. The (exact) position vectors of P_1 and P_2 are given by

$$\boldsymbol{p}(P_1) = (- - - - \bullet\bullet)\,, \qquad \boldsymbol{p}(P_2) = (\bullet\ \bullet - \bullet - -)\,.$$

The position vectors of the four adjacent cells of $\boldsymbol{v}$ are given by

$$\begin{aligned} \boldsymbol{p}_1 &:= (+ + - - + -)\,, & \boldsymbol{p}_2 &:= (- + - - + -)\,, \\ \boldsymbol{p}_3 &:= (+ - - - + -)\,, & \boldsymbol{p}_4 &:= (- - - - + -)\,. \end{aligned}$$

$\boldsymbol{p}(P_2)$ and $\boldsymbol{p}_i$ do not match for $i \in [4]$. $\boldsymbol{p}(P_1)$ matches $\boldsymbol{p}_4$ but not $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3$. Therefore $C := \mathcal{H}(\boldsymbol{p}_4)$ is the only active cell adjacent to $\boldsymbol{v}$. The corresponding vertex cone of $\boldsymbol{v}$ at the cell $\mathcal{H}(\boldsymbol{p}_4)$ is given by the position vector $\boldsymbol{p}(K_{\boldsymbol{v}}(C)) = (- -\bullet\ \bullet\ \bullet)$.

We can decompose a union of polytopes $\mathcal{P}$ into the active cells of the corresponding

hyperplane arrangement $\mathcal{H}_\mathcal{P}$.

Proposition 3.10 *Let $\mathcal{P} := \bigcup_{i\in[k]} P_i$ with $P_i \subset \mathbb{R}^d$ being full-dimensional polytopes. Then there exist cells $C_1, \dots, C_l$ of $\mathcal{H}_\mathcal{P}$ such that we can write $\mathcal{P} = \bigcup_{j\in[l]} C_j$.*

Proof: The union of all cells of $\mathcal{H}_\mathcal{P}$ is the entire space $\mathbb{R}^d$. Therefore there are cells $C_i, i \in [r]$ of $\mathcal{H}$ that cover $\mathcal{P}$, that means $\mathcal{P} \subset \bigcup_{i\in[r]} C_i$. W.l.o.g. let $\mathcal{C} := \{C_1, \dots, C_l\}$ be the smallest cover.
We assume that there is a point $\boldsymbol{v} \in \bigcup_{i\in[l]} C_i$ with $\boldsymbol{v} \notin \mathcal{P}$. Let C be a cell of the cover with $\boldsymbol{v} \in C$. We know that there is a point $\overline{\boldsymbol{v}} \in C \cap \mathcal{P}$ since otherwise $C \notin \mathcal{C}$ because $\mathcal{C}$ is a minimal cover. We assume that $\overline{\boldsymbol{v}}$ is in the interior of the cell C, denoted by $\mathring{C}$. If there is is no point $\overline{\boldsymbol{v}} \in \mathring{C} \cap \mathcal{P}$ then we do not need C for the cover since $\mathcal{P}$ is the union of full dimensional polytopes.
We can conclude that there has to be a hyperplane in $\mathcal{H}_\mathcal{P}$ that separates $\overline{\boldsymbol{v}}$ and $\boldsymbol{v}$. But that stands in contradiction to C being a cell of $\mathcal{H}_\mathcal{P}$. □

We have learned that $\mathcal{P}$ is the union of a subset of cells of the hyperplane arrangement $\mathcal{H}_\mathcal{P}$.

Corollary 3.11 *Let $\mathcal{P}$ be a union of polytopes which can be described as $\mathcal{P} = \bigcup_{i\in[l]} C_i$ for some cells $C_1, \dots, C_l$ of the hyperplane arrangement $\mathcal{H}_\mathcal{P}$. Then*

$$\mathrm{Vol}(\mathcal{P}) = \sum_{i\in[l]} \mathrm{Vol}(C_i).$$

Proof: The boundary of all cells has Lebesgue measure 0. Therefore it holds true that $\mathrm{Vol}(C_i) = \mathrm{Vol}(\mathring{C}_i)$ for $i \in [l]$. Therefore

$$\mathrm{Vol}(\mathcal{P}) = \mathrm{Vol}(\bigcup_{i\in[l]} C_i) = \mathrm{Vol}(\bigcup_{i\in[l]} \mathring{C}_i). \tag{3.1}$$

Since the sets $\mathring{C}_i, \mathring{C}_j$ are disjoined for $i \neq j \in [l]$ it holds

$$\mathrm{Vol}(\bigcup_{i\in[l]} \mathring{C}_i) = \sum_{i\in[l]} \mathrm{Vol}(\mathring{C}_i) = \sum_{i\in[l]} \mathrm{Vol}(C_i).$$

□

As we have learned in Section 2.2 we can describe a polytope as the weighted sum of its forward cones. For a vertex $\boldsymbol{v}$ and a polytope P the forward cone depends on the vertex cone at $\boldsymbol{v}$ and the sweep-plane. Since we want to be able to compute the volume for multiple sweep-planes efficiently, it makes sense to save the vertex cones of the cells in the data structure. That means we need a way to find the vertex cones at a vertex. To determine those cones we first look for the adjacent cells of a vertex.

Proposition 3.12 *Let $\boldsymbol{v}$ be a vertex of a hyperplane arrangement $\mathcal{H} = \{H_1, \dots, H_m\}$ with exact position vector $\boldsymbol{p}(\boldsymbol{v})$. Let C be a cell of $\mathcal{H}$ with exact position vector $\boldsymbol{p}(C)$. Then the following are equivalent*

(i) $\boldsymbol{v} \in C$

(ii) $\boldsymbol{p}(\boldsymbol{v})$ *matches* $\boldsymbol{p}(C)$.

Proof: Let $\boldsymbol{v} \in C$. Then $p_i(\boldsymbol{v}) = p_i(C)$ if H_i is not active at $\boldsymbol{v}$ since $\boldsymbol{v} \in C$. If H_i is active at $\boldsymbol{v}$, then $p_i(\boldsymbol{v}) = 0$. Therefore the position vectors match.
The other direction directly follows from the definition of matching position vectors. □

This way we can find the adjacent cells to a vertex by iterating over all possible matching position vectors. In a non-degenerate hyperplane arrangement all matching position vectors correspond to cells. In a degenerate hyperplane arrangement some of the position vectors correspond to cells which are not full-dimensional.

Proposition 3.13 *Let $\mathcal{H}$ be the hyperplane arrangement corresponding to the union of polytopes $\mathcal{P} = \bigcup_{i \in [k]} P_i$. Let $\boldsymbol{p}(P_i)$ be the position vector of P_i for $i \in [k]$. Let C be a cell of $\mathcal{H}$ with position vector $\boldsymbol{p}(C)$.*
Then C is active if and only if there is a $j \in [l]$ such that $\boldsymbol{p}(C)$ and $\boldsymbol{p}(P_j)$ match.

Proof: Let C be active. We want to show that there is an P_i with matching position vector. First we observe that every active cell is contained in at least one polytope P_i. From $C \subseteq P_i$ then follows $p_j(C) = p_j(P_i)$ if $p_j(P_i) \in \{+, -\}$. Therefore the position vectors match.
Let C be a cell and P_i a polytope such that $\boldsymbol{p}(C)$ and $\boldsymbol{p}(P_i)$ match. Let $\boldsymbol{x} \in C$. Since the position vectors match, $\boldsymbol{x}$ is in all P_i describing halfspaces. That means $\boldsymbol{x} \in P_i$ and therefore C is active. □

This allows us to test whether a cell is active by comparing the position vector to the position vectors of the polytopes. Combined with Proposition 3.12 we are able to determine the active cells of a hyperplane arrangement that are adjacent to a given vertex. For the conic decomposition we need the forward cones. A forward cone only depends on the vertex cone and the sweep direction. The vertex cone $K_{\boldsymbol{v}}(C)$ of $\boldsymbol{v}$ at C is defined as in Definition 2.5 by the halfspaces of C that correspond to the active hyperplanes at $\boldsymbol{v}$.

Corollary 3.14 *Let $\mathcal{H} = \{H_1, \dots, H_m\}$ be a hyperplane arrangement. Let C be a cell of $\mathcal{H}$ and $\boldsymbol{v} \in C$ be a vertex. Then the vertex cone of $\boldsymbol{v}$ at C is given by the position vector*

$$p_i(K_{\boldsymbol{v}}(C)) := \begin{cases} p_i(C) & \text{if } p_i(\boldsymbol{v}) = 0 \\ \bullet & \text{else} \end{cases}. \tag{3.2}$$

3.2 The Algorithm

We now are able to give a more detailed version of Algorithm 1. In Section 3.2.1 we will discuss the computational complexity of Algorithm 4 for the non-degenerate case. The algorithm and the runtime estimation are based on [GH17] where all vertices and cells of a hyperplane arrangement are calculated. We are not explicitly stating how to compute the vertices of the hyperplane arrangement. The key there is to iterate over all subsets of d hyperplanes. For the non-degenerate case their intersections are the vertices of the hyperplane arrangement. For degenerate hyperplane arrangements some of those intersections might be empty. Also it can happen that more than d many hyperplanes intersect in a vertex. All vertices and incidences can be found by iterating over the subsets of the hyperplane arrangement that have d elements. For clever ways to iterate over those subsets the interested reader is referred to [Knu05]. We also need to iterate over the adjacent cells of a vertex, which can be done by using Proposition 3.12.

Remark The decomposition of $\mathcal{P}$ into cells as described in Proposition 3.10 only works for the union of full-dimensional polytopes. Anyhow, with our procedure we find all cells $C_1, \dots, C_l$ that are active at $\mathcal{P}$. The remaining part of $\mathcal{P}$, namely $\mathcal{P} \setminus \bigcup_{i \in [l]} C_i$, has volume 0. Therefore our algorithm works even if some polytopes are not full-dimensional.

Example 3.15 We return to Example 3.9, for which the hyperplane arrangement is displayed in Figure 3.1. We now look at the degenerate vertex $\boldsymbol{w} = \begin{pmatrix} 0 \\ \frac{0}{5} \end{pmatrix}$ for which the exact position vector is $\boldsymbol{p}(\boldsymbol{w}) = (0---00)$. The 8 position vectors that match $\boldsymbol{p}(\boldsymbol{w})$ and result from replacing the 0's with $+/-$ are

$$\begin{aligned}
&\boldsymbol{p}_1 = (+---++)\,, \quad \boldsymbol{p}_2 = (+----+-)\,, && \boldsymbol{p}_3 = (+-----)\,,\\
&\boldsymbol{p}_4 = (+----+)\,, \quad \boldsymbol{p}_5 = (-----++)\,, && \boldsymbol{p}_6 = (-----+-)\,,\\
&\boldsymbol{p}_7 = (------)\,, \quad \boldsymbol{p}_8 = (-----+)\,.
\end{aligned}$$

The sets $\mathcal{H}(\boldsymbol{p}_3)$ and $\mathcal{H}(\boldsymbol{p}_5)$ are not full-dimensional but equal $\{\boldsymbol{w}\}$. The remaining position vectors describe the 6 adjacent cells of $\boldsymbol{v}_4$. Furthermore $\boldsymbol{p}_6$, $\boldsymbol{p}_7$ and $\boldsymbol{p}_8$ match $\boldsymbol{p}(P_1)$ and are therefore active. Since $\boldsymbol{v}$ is degenerate the vertex cones have to get regularized by Algorithm 3. The vertex and its regularized cones are then added to the datastructure.

Algorithm 4 Data Structure Algorithm

Input A hyperplane arrangement $\mathcal{H}$ and polytopes $\boldsymbol{p}(P_1), \ldots, \boldsymbol{p}(P_k)$ as position vectors of $\mathcal{H}$.
Output A list of events $E = [(\boldsymbol{v}_1, \mathcal{K}_{\boldsymbol{v}_1}), \ldots, (\boldsymbol{v}_n, \mathcal{K}_{\boldsymbol{v}_n})]$.

$E \leftarrow []$
compute vertices $\mathcal{V}$ of $\mathcal{H}$
for $\boldsymbol{v} \in \mathcal{V}$ **do**
    $\mathcal{K}_{\boldsymbol{v}} \leftarrow []$, a list of vertex cones at $\boldsymbol{v}$
    compute exact position vector $\boldsymbol{p}(\boldsymbol{v})$
    for position vector $\boldsymbol{p}(C) \in \{+, -\}^{|\mathcal{H}|}$ that matches $\boldsymbol{p}(\boldsymbol{v})$ **do**
        if $\boldsymbol{p}(C)$ matches $\boldsymbol{p}(P_i)$ for any $i \in [k]$ **then**
            if $\boldsymbol{v}$ is non-degenerate **then**
                $\mathcal{K}_{\boldsymbol{v}}$.add($K_{\boldsymbol{v}}(C)$)
            else if $K_{\boldsymbol{v}}$ is full-dimensional **then**
                $K \leftarrow$ Regularization($K_{\boldsymbol{v}}$)
                $\mathcal{K}_{\boldsymbol{v}}$.union($K$)
            end if
        end if
    end for
    if $\mathcal{K}_{\boldsymbol{v}}$ is not empty **then**
        E.append($\boldsymbol{v}, \mathcal{K}_{\boldsymbol{v}}$)
    end if
end for
return E

3.2.1 Computational Complexity

Let P_i be full-dimensional polytopes for $i \in [k]$ and $\mathcal{P} := \bigcup_{i \in [k]} P_i$. We assume that we have a non-degenerate underlying hyperplane arrangement $\mathcal{H}_{\mathcal{P}}$ with $|\mathcal{H}_{\mathcal{P}}| = m$. $\mathcal{H}_{\mathcal{P}}$ has exactly $\binom{m}{d}$ many vertices. To calculate the coordinates of $\boldsymbol{v}$, we have to solve the corresponding linear equation system which costs $\mathcal{O}(d^3)$ operations. The position vector of a vertex is non-zero at $m - d$ many hyperplanes. There the position code can be calculated by calculating the scalar product which can be done in $\mathcal{O}(d)$.
A vertex has exactly 2^d many adjacent cells. To test whether a cell is active it is needed to test whether its position vector matches the position of any polytope. This can be done in $\mathcal{O}(mk)$.
That means the computational complexity of the algorithm is given by

$$\mathcal{O}(\binom{m}{d}(d^3 + 2^d mk + (m-d)d)). \tag{3.3}$$

4 Sweep-Plane Volume

In this chapter we will learn how to calculate the sweep-plane volume. We will first introduce a volume formula for the simplex, which we then use to calculate the volume of simplicial cones. We will give an explicit sweep-plane volume formula for the general, potentially degenerate case. In the second section we are giving a more detailed version of Algorithm 2.

4.1 A Formula for the Volume

First we develop a formula for the volume of a simplex. We follow the proof given in [Ste66].

Theorem 4.1 *The volume of a simplex $S \subset \mathbb{R}^d$ with vertices $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{d+1}$ is given by the equation:*

$$\mathrm{Vol}(S) = \frac{1}{d!}|\det(\begin{pmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \ldots & \boldsymbol{s}_{d+1} \\ 1 & 1 & \ldots & 1 \end{pmatrix})| \tag{4.1}$$

Proof: It is easy to see that there is an affine transformation $f(x) = Ax + b$ that maps the vertices $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n+1}$ onto vertices $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n+1}$ with

$$\boldsymbol{x}_1^T = \begin{pmatrix} x_0 & 0 & \ldots 0 \end{pmatrix}, \quad \boldsymbol{x}_i^T = \begin{pmatrix} 0 & x_{i_2} & \ldots & x_{i_d} \end{pmatrix} \qquad i \in \{2, \ldots, d+1\}.$$

We can choose an affine transformation that fulfills the above properties with orthogonal A, i.e. $|\det(A)| = 1$. This transformation maps S onto $X = \mathrm{conv}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1})$, the convex hull of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{d+1}$. With the multiplicativity of the determinant it follows that

$$\begin{aligned} |\det(\begin{pmatrix} \boldsymbol{s}_1 & \boldsymbol{s}_2 & \ldots & \boldsymbol{s}_{d+1} \\ 1 & 1 & \ldots & 1 \end{pmatrix})| &= |\det(A)\det(\begin{pmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix})| \\ &= |\det(\begin{pmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix})|. \end{aligned}$$

Every point $\boldsymbol{y} \in X$ can be written as a convex combination $\boldsymbol{y} = \sum_{i \in [d+1]} \alpha_i \boldsymbol{x}_i$ with $\sum_{i=2}^{d+1} \alpha_i = 1 - \alpha_1$.

The idea now is to find a parametrization of X into the first coordinate and the corresponding $n-1$ dimensional simplex. Let $\hat{\boldsymbol{x}}_i$ be the point that results from

deleting the first row of $\boldsymbol{x}_i$ for $i \in \{2, \ldots, d+1\}$. Furthermore let

$$S_{\alpha_1} := \text{conv}((1-\alpha_1)\hat{\boldsymbol{x}}_2, \ldots, (1-\alpha_1)\hat{\boldsymbol{x}}_{d+1}).$$

We can then write every point $\boldsymbol{y} = \sum_{i\in[d+1]} \alpha_i \boldsymbol{x}_i \in X$ as $y = \alpha_1 x_0 \times y'$ with $y' = \sum_{i=2}^{d+1} \alpha_i \hat{\boldsymbol{x}}_i \in S_{\alpha_1}$. It follows $\text{Vol}(X) = \int_X \mathbf{1} d\mu = \int_{[0,1]} x_0 [\int_{S_{\alpha_1}} \mathbf{1} d\mu \,] d\alpha_1$. See Figure 4.1.
We will prove the assumption by induction. Equation (4.1) holds for the two dimensional case. We assume that eq. (4.1) is also true for the $n-1$ dimensional case. With the above observations we have

$$\begin{aligned}
\text{Vol}(S) &= \int_{[0,1]} x_0 [\int_{S_{\alpha_1}} \mathbf{1} d\mu] d\alpha_1 \\
&= \int_{[0,1]} x_0 [\frac{1}{(d-1)!} |\det(\begin{pmatrix} (1-\alpha_1)\hat{\boldsymbol{x}}_2 & \ldots & (1-\alpha_1)\hat{\boldsymbol{x}}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix})|] d\alpha_1 \\
&= \int_{[0,1]} x_0 [\frac{(1-\alpha_1)^{d-1}}{(d-1)!} |\det(\begin{pmatrix} \hat{\boldsymbol{x}}_2 & \ldots & \hat{\boldsymbol{x}}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix})|] d\alpha_1 \\
&= \frac{1}{(d-1)!} \det(\begin{pmatrix} \hat{\boldsymbol{x}}_2 & \ldots & \hat{\boldsymbol{x}}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix}) \int_{[0,1]} (1-\alpha_1)^{(d-1)} x_0 d\alpha_1.
\end{aligned}$$

For the integral it holds true that $\int_{[0,1]} (1-\alpha_1)^{(d-1)} x_0 d\alpha_1 = \frac{x_0}{d}$. With Laplace's formula it is easy to see that

$$\begin{aligned}
\text{Vol}(S) &= \frac{1}{(d-1)!} \det(\begin{pmatrix} \hat{\boldsymbol{x}}_2 & \ldots & \hat{\boldsymbol{x}}_{d+1} \\ 1 & \ldots & 1 \end{pmatrix}) \frac{x_o}{d} \\
&= \frac{x_0}{(d)!} \det(\begin{pmatrix} x_{2_2} & \ldots & x_{d+1_2} \\ \vdots & \ddots & \vdots \\ x_{2_d} & \ldots & x_{d+1_d} \\ 1 & \ldots & 1 \end{pmatrix} \\
&= \frac{1}{(d)!} |\det(\begin{pmatrix} x_0 & 0 & \ldots & 0 \\ 0 & x_{2_2} & \ldots & x_{d+1_2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & x_{2_d} & \ldots & x_{d+1_d} \\ 1 & 1 & \ldots & 1 \end{pmatrix})|.
\end{aligned}$$

□

At the point at which we have to calculate the sweep-plane volume, we will have a cone represented as the Minkowki sum of its vertex and its rays. The following corollary

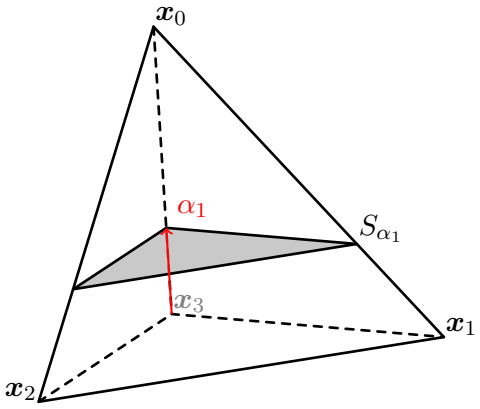


Figure 4.1: The parametrization of the simplex.

makes it easier to calculate the volume of a cone intersected with a sweep-plane.

Corollary 4.2 *The volume of a simplex $S \subset \mathbb{R}^d$ with vertices $\boldsymbol{s}_1, \dots, \boldsymbol{s}_{d+1}$ is given by the equation:*

$$\mathrm{Vol}(S) = \frac{1}{d!}|\det(\begin{pmatrix}\boldsymbol{s}_2 - \boldsymbol{s}_1 & \dots & \boldsymbol{s}_{d+1} - \boldsymbol{s}_1\end{pmatrix})| \tag{4.2}$$

Proof: We know that eq. (4.1) holds true. If we subtract the first column from the other columns we get:

$$\mathrm{Vol}(S) = \frac{1}{d!}|\det(\begin{pmatrix}\boldsymbol{s}_1 & \boldsymbol{s}_2 - \boldsymbol{s}_1 & \dots & \boldsymbol{s}_{d+1} - \boldsymbol{s}_1 \\ 1 & 0 & \dots & 0\end{pmatrix})|$$

Laplace's formula along the last row then gives eq. (4.2) □

This formula can be used to calculate the sweep-plane volume of forward cones.

Proposition 4.3 *Let $K_{\boldsymbol{v}} = \{\boldsymbol{v}\} + pos(\boldsymbol{r}_1, \dots, \boldsymbol{r}_d)$ be a simplicial cone. Let $\boldsymbol{a}$ be a sweep direction with $\langle \boldsymbol{a}, \boldsymbol{r}_i \rangle > 0$ for $i \in [d]$. With $\lambda_{\boldsymbol{v}} := \langle \boldsymbol{a}, \boldsymbol{v} \rangle$ it holds true that*

$$\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}})^d \frac{|\det(\boldsymbol{r}_1, \dots, \boldsymbol{r}_d)|}{d! \prod\limits_{i \in [d]} \langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}. \tag{4.3}$$

Furthermore we denote $\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}})^d \gamma$ *with*

$$\gamma := \frac{|\det(\boldsymbol{r}_1, \dots, \boldsymbol{r}_d)|}{d! \prod\limits_{i \in [d]} \langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}.$$

Proof: Let $\lambda < \lambda_{\boldsymbol{v}}$. Then the set $K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)$ is empty and therefore $\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = 0$. With $\mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda) = 0$ the right hand side of eq. (4.3) is equal 0. For

$\lambda = \lambda_{\boldsymbol{v}}$ we have $K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda) = \{\boldsymbol{v}\}$ and therefore both sides of eq. (4.3) are equal 0. Let $\lambda > \lambda_{\boldsymbol{v}}$. Since $\boldsymbol{v}$ is a non-degenerate vertex, $K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)$ is a simplex.
We show that the vertices of $K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)$ are given by $\boldsymbol{v}$ and $\frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \boldsymbol{r}_j + \boldsymbol{v}$ for $j \in [d]$. That is the case if and only if $\frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \boldsymbol{r}_j + \boldsymbol{v} \in H(\boldsymbol{a}, \lambda)$. Therefore it remains to show $\langle \frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \boldsymbol{r}_j + \boldsymbol{v}, \boldsymbol{a} \rangle = \lambda$:

$$\begin{aligned}
\langle \frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \boldsymbol{r}_j + \boldsymbol{v}, \boldsymbol{a} \rangle &= \langle \frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \boldsymbol{r}_j, \boldsymbol{a} \rangle + \langle \boldsymbol{v}, \boldsymbol{a} \rangle \\
&= \frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_j \rangle} \langle \boldsymbol{r}_j, \boldsymbol{a} \rangle + \lambda_{\boldsymbol{v}} \\
&= (\lambda - \lambda_{\boldsymbol{v}}) + \lambda_{\boldsymbol{v}} \\
&= \lambda
\end{aligned}$$

With eq. (4.2) it follows that

$$\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = \frac{1}{d!} |\det(\frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_1 \rangle} \boldsymbol{r}_1, \ldots, \frac{(\lambda - \lambda_{\boldsymbol{v}})}{\langle \boldsymbol{a}, \boldsymbol{r}_d \rangle} \boldsymbol{r}_d)|. \tag{4.4}$$

From the multilinearity of the determinant it follows that

$$\mathrm{Vol}(K_{\boldsymbol{v}} \cap H^-(\boldsymbol{a}, \lambda)) = (\lambda - \lambda_{\boldsymbol{v}})^d \frac{|\det(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)|}{d! \prod\limits_{i \in [d]} \langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}. \tag{4.5}$$

□

We are now able to calculate the volume for forward cones. In the decomposition of our union of polytopes the sign of a forward cone is determined by the number of rays that have to be flipped to construct the forward cone from the vertex cone. See Definition 2.18.

Proposition 4.4 *Let $K_{\boldsymbol{v}} = \{\boldsymbol{v}\} + \mathrm{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)$ be a simplical cone. Let $\boldsymbol{a}$ be a valid sweep direction for $K_{\boldsymbol{v}}$ and $F(K_{\boldsymbol{v}}, \boldsymbol{a}) = \{\boldsymbol{v}\} + \mathrm{pos}(\overline{\boldsymbol{r}}_1, \ldots, \overline{\boldsymbol{r}}_d)$ be the corresponding forward cone. Let $\lambda_{\boldsymbol{v}} := \langle \boldsymbol{a}, \boldsymbol{v} \rangle$. Then*

$$\mathrm{Vol}(F(K_{\boldsymbol{v}}, \boldsymbol{a}) \cap H^-(\boldsymbol{a}, \lambda)) = \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda) (-1)^{\sigma(K_{\boldsymbol{v}}, \boldsymbol{a})} (\lambda - \lambda_{\boldsymbol{v}})^d \frac{|\det(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)|}{d! \prod\limits_{i \in [d]} \langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}. \tag{4.6}$$

Proof: With Proposition 4.3 we have

$$\mathrm{Vol}(F(K_{\boldsymbol{v}}, \boldsymbol{a}) \cap H^-(\boldsymbol{a}, \lambda)) = \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda) (\lambda - \lambda_{\boldsymbol{v}})^d \frac{|\det(\overline{\boldsymbol{r}}_1, \ldots, \overline{\boldsymbol{r}}_d)|}{d! \prod\limits_{i \in [d]} \langle \boldsymbol{a}, \overline{\boldsymbol{r}}_i \rangle}.$$

We know $\overline{\boldsymbol{r}} = \mu_i \boldsymbol{r}_i$ for some $\mu_i \in \mathbb{R}\backslash\{0\}$ and $i \in [d]$. We can then write eq. (4.6) as

$$\begin{aligned}\operatorname{Vol}(F(K_{\boldsymbol{v}}, \boldsymbol{a}) \cap H^-(\boldsymbol{a}, \lambda)) &= \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda) \frac{(\lambda - \lambda_{\boldsymbol{v}})^d |\det(\mu_1 \boldsymbol{r}_1, \ldots, \mu_d \boldsymbol{r}_d)|}{d!} \prod_{i \in [d]} \frac{1}{\langle \boldsymbol{a}, \mu_i \boldsymbol{r}_i \rangle} \\ &= \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}})^d \prod_{i \in [d]} \frac{|\mu_i|}{\mu_i} \frac{|\det(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)|}{d! \prod_{i \in [d]} \langle \boldsymbol{a}, \boldsymbol{r}_i \rangle}.\end{aligned}$$

With $(-1)^{\sigma(F(K_{\boldsymbol{v}}, \boldsymbol{a}))} = \prod_{i \in [d]} \frac{|\mu_i|}{\mu_i}$ the proof is complete. □

This allows us to write the volume by only using the rays of the vertex cones. Combining the results of this chapter, we have this rather long but complete formula for the sweep-plane volume.

Corollary 4.5 *Let $\mathcal{P}$ be a union of polytopes, $\mathcal{H}(\mathcal{P})$ the underlying hyperplane arrangement, $C_1, \ldots, C_l$ the active cells of $\mathcal{H}(\mathcal{P})$ and $\mathcal{V} := \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ the vertices that are incident to an active cell. Let $K^1_{\boldsymbol{v}_i}, \ldots, K^{m_i}_{\boldsymbol{v}_i}$ be the regularized vertex cones that correspond to active cells at $\boldsymbol{v}_i$ for $i \in [n]$. Let $K^j_{\boldsymbol{v}_i} = \{\boldsymbol{v}_i\} + \operatorname{pos}(\boldsymbol{r}_1^{i_j}, \ldots, \boldsymbol{r}_d^{i_j})$ for $i \in [n]$ and $j \in [m_i]$. We write $\lambda_{\boldsymbol{v}} := \langle \boldsymbol{a}, \boldsymbol{v} \rangle$ for any $\boldsymbol{v} \in \mathbb{R}^d$. Then*

$$\operatorname{Vol}(P \cap H^-(\boldsymbol{a}, \lambda)) = \sum_{i \in [n]} \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}_i})^d \sum_{j \in [m_i]} \gamma_{i_j} \tag{4.7}$$

with $\gamma_{i_j} := \frac{|\det(\boldsymbol{r}_1^{i_j}, \ldots, \boldsymbol{r}_d^{i_j}))|}{d!} \prod_{l \in [d]} \frac{1}{\langle \boldsymbol{a}, \boldsymbol{r}_l^{i_j} \rangle}$ for $i \in [n]$ and $j \in [m_i]$.

Proof: We know $\operatorname{Vol}(P \cap H^-(\boldsymbol{a}, \lambda)) = \sum_{i \in [l]} \operatorname{Vol}(C_i \cap H^-(\boldsymbol{a}, \lambda))$ by Corollary 3.11. With Theorem 2.33 we can write every cell C_i as the weighted sum of its regularized forward cones. We change the order by, instead of summing over the cells and their forward cones, summing over the vertices and their forward cones. We get

$$\operatorname{Vol}(P \cap H^-(\boldsymbol{a}, \lambda)) = \sum_{i \in [n]} \sum_{j \in [m_i]} (-1)^{\sigma(K^j_{\boldsymbol{v}_i}, \boldsymbol{a})} \operatorname{Vol}(F(K^j_{\boldsymbol{v}_i}, \boldsymbol{a}) \cap H^-(\boldsymbol{a}, \lambda)). \tag{4.8}$$

By combining Proposition 4.3 and Equation (4.6) we get

$$\begin{aligned}\operatorname{Vol}(P \cap H^-(\boldsymbol{a}, \lambda)) &= \sum_{i \in [n]} \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}_i})^d \sum_{j \in [m_i]} (-1)^{2\sigma(K^j_{\boldsymbol{v}_i}, \boldsymbol{a})} \gamma_{i_j} \\ &= \sum_{i \in [n]} \mathbb{1}_{[\lambda_{\boldsymbol{v}}, \infty)}(\lambda)(\lambda - \lambda_{\boldsymbol{v}_i})^d \sum_{j \in [m_i]} \gamma_{i_j}.\end{aligned}$$

□

4.2 The Algorithm

We are now able to give a more detailed version of Algorithm 2. The computational complexity of Algorithm 5 depends on the number of forward cones in the decomposition. If the underlying hyperplane arrangement is non-degenerate, the number of forward cones is bounded from above by $2^d\binom{n}{d}$, whereby n is the number of hyperplanes. That is due to the fact that every vertex is incident to exactly 2^d cells in a non-degenerate hyperplane arrangement. In reality the number of forward cones for a union of polytopes is much lower. We refer to Section 5.2 for a better runtime estimation.

Algorithm 5 Sweep-Plane Volume

Input A set of events E as returned by Algorithm 1 and a vector $\boldsymbol{a}$.
Output The sweep-plane volume as a piecewise polynomial function p.

sort E by $\lambda_{\boldsymbol{v}} = \langle \boldsymbol{a}, \boldsymbol{v} \rangle$
$p \leftarrow []$
for $e = (\boldsymbol{v}, \mathcal{K}_{\boldsymbol{v}})$ event in E **do**
    $\gamma = 0$
    for cone $K_{\boldsymbol{v}} = \{\boldsymbol{v}\} + \text{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d) \in \mathcal{K}_{\boldsymbol{v}}$ **do**
        $\gamma_F = \frac{|\det(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)|}{d!} \prod_{l \in [d]} \frac{1}{\langle \boldsymbol{a}, \boldsymbol{r}_l \rangle}$
        $\gamma = \gamma + \gamma_F$
    end for
    p.append($[\lambda_{\boldsymbol{v}}, \gamma]$)
end for
return p

Example 4.6 We return to Example 2.25. The polytope and the graph of the sweep-plane volume function are shown in Figure 4.2.
The vertex cones are given by

$$K_{\boldsymbol{v}_1} = \{\boldsymbol{v}_1\} + \text{pos}\left(\begin{pmatrix}1\\ \frac{1}{2}\end{pmatrix}, \begin{pmatrix}1\\ 1\end{pmatrix}\right), \qquad K_{\boldsymbol{v}_2} = \{\boldsymbol{v}_2\} + \text{pos}\left(\begin{pmatrix}-1\\ -\frac{1}{2}\end{pmatrix}, \begin{pmatrix}0\\ 1\end{pmatrix}\right),$$

$$K_{\boldsymbol{v}_2} = \{\boldsymbol{v}_3\} + \text{pos}\left(\begin{pmatrix}0\\ -\frac{1}{2}\end{pmatrix}, \begin{pmatrix}-1\\ -1\end{pmatrix}\right).$$

The vertices are already sorted with respect to the sweep direction $\boldsymbol{a} := \begin{pmatrix}0\\ 1\end{pmatrix}$. We

have the events $e_i = (\boldsymbol{v}_i, (K_{\boldsymbol{v}_i}))$ for $i \in [3]$. We have

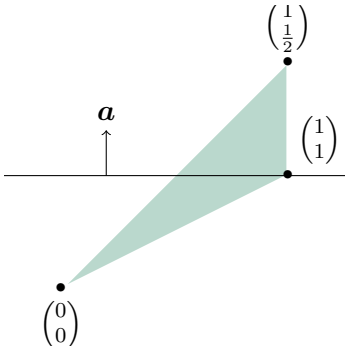
$$\gamma_1 = \frac{|\det \begin{pmatrix} 1, 1 \\ \frac{1}{2}, 1 \end{pmatrix}|}{2!} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} \rangle} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rangle} = \frac{1}{2}$$

$$\gamma_2 = \frac{|\det \begin{pmatrix} -1, 0 \\ -\frac{1}{2}, 1 \end{pmatrix}|}{2!} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -\frac{1}{2} \end{pmatrix} \rangle} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rangle} = -1$$

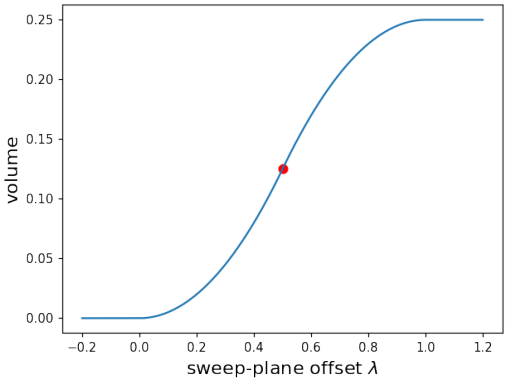
$$\gamma_3 = \frac{|\det \begin{pmatrix} 0, -1 \\ -\frac{1}{2}, -1 \end{pmatrix}|}{2!} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{2} \end{pmatrix} \rangle} \cdot \frac{1}{\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \rangle} = \frac{1}{2}.$$

Furthermore $\langle \boldsymbol{a}, \boldsymbol{v}_1 \rangle = 0$, $\langle \boldsymbol{a}, \boldsymbol{v}_2 \rangle = 1$ and $\langle \boldsymbol{a}, \boldsymbol{v}_3 \rangle = \frac{1}{2}$. Algorithm 5 returns $p = [[0, \frac{1}{2}], [\frac{1}{2}, -1], [1, \frac{1}{2}]]$ which corresponds to the piecewise polynomial function

$$\begin{aligned} p(\lambda) := &+ \mathbb{1}_{[0,\infty)}(\lambda)\lambda^2 \frac{1}{2} \\ &- \mathbb{1}_{[\frac{1}{2},\infty)}(\lambda)(\lambda - \frac{1}{2})^2 \\ &+ \mathbb{1}_{[1,\infty)}(\lambda)(\lambda - 1)^2 \frac{1}{2}. \end{aligned}$$



(a) The polytope and the sweep-plane at the time $\lambda = \frac{1}{2}$.



(b) The graph of the sweep-plane volume function.

Figure 4.2: A triangle and the corresponding volume graph. See Example 4.6.

5 Implementation and Computational Results

In this chapter we will first discuss details regarding the implementation. Afterwards we will look at runtimes on different data sets and compare the algorithm with an algorithm based on the *inclusion-exclusion principle*. After that we take a look at our work in progress, which is finding ACD cuts with the help of our volume algorithm.

5.1 Implementation

We tested the algorithm in different dimensions against a monte-carlo implentation and in dimension two and three against an algorithm based on the inclusion-exclusion principle. We implemented the algorithm in python, since it allows fast and easy prototyping.

5.1.1 Hyperplane Comparison

Preferably every hyperplane is contained only once in a hyperplane arrangement. For that reason we need a way to compare hyperplanes. Different vectors $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^d$ and numbers $\lambda_1, \lambda_2 \in \mathbb{R}$ can lead to the same set $H(\boldsymbol{a}_1, \lambda_1) = H(\boldsymbol{a}_2, \lambda_2)$. Therefore it makes sense to find an unique representation for a hyperplane.

Definition 5.1 We call a hyperplane $H(\boldsymbol{a}, \lambda)$ *normed* if and only if the first non-zero entry of $\boldsymbol{a}$ is equal to 1.

Corollary 5.2 *Every hyperplane $H(\boldsymbol{a}, \lambda)$ has a normed representation. That means there is a $\gamma \in \mathbb{R}$ such that $H(\boldsymbol{a}, \lambda) = H(\frac{\boldsymbol{a}}{\gamma}, \frac{\lambda}{\gamma})$ and $H(\frac{\boldsymbol{a}}{\gamma}, \frac{\lambda}{\gamma})$ is normed.*

Proof: Clearly we can set γ equal to the first non-zero entry of $\boldsymbol{a}$ and obtain the desired result. □

5.1.2 Neutralizing Vertex Cones

Often the same forward cone is included twice, but with different sign, in the conic decomposition of the union of polytopes. If this is the case, those volumes neutralize themselves in the conic decomposition.

Proposition 5.3 *Let $K_1, K_2 \subset \mathbb{R}^d$ be two vertex cones at the non-degenerate vertex $\boldsymbol{v}$ and $\boldsymbol{a}$ be a valid sweep direction. Further let $\boldsymbol{p}(K_1)$, $\boldsymbol{p}(K_2)$ be the position vectors of the cones with respect to $\mathcal{H} := \{H | H \text{ is active hyperplane at } \boldsymbol{v}\}$. Since $\boldsymbol{v}$ is non-degenerate there are exactly d many hyperplanes in $\mathcal{H}$.*
If there is a $j \in [d]$ with $p_i(K_1) = p_i(K_2)$ for $i \neq j$ and one of the following:

(i) $p_j(K_1) = +$ *and* $p_j(K_2) = -$

(ii) $p_j(K_1) = -$ *and* $p_j(K_2) = +$

Then it holds true that

(i) $F(K_1, \boldsymbol{a}) = F(K_2, \boldsymbol{a})$

(ii) $\sigma(K_1, \boldsymbol{a}) = -\sigma(K_2, \boldsymbol{a})$

Proof: We look at the to the position vectors corresponding $d \times d$ linear inequality systems. We can write $A_{K_1}\boldsymbol{x} \leq \boldsymbol{\lambda}_1$ and $A_{K_2}\boldsymbol{x} \leq \boldsymbol{\lambda}_2$ by multiplying the inequalities $\boldsymbol{a}_i \geq \lambda_i$ with -1 if $p_i(K_j) = +$ for $j \in [2]$. For $A_{K_1} = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_j \\ \vdots \\ \boldsymbol{a}_d \end{pmatrix}$ it then holds true that $A_{K_2} = \begin{pmatrix} \boldsymbol{a}_1 \\ \vdots \\ -\boldsymbol{a}_j \\ \vdots \\ \boldsymbol{a}_d \end{pmatrix}$. With Proposition 2.10 the cones are then given by $K_1 = \{\boldsymbol{v}\} + \operatorname{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_j, \ldots, \boldsymbol{r}_d)$ and $K_2 = \{\boldsymbol{v}\} + \operatorname{pos}(\boldsymbol{r}_1, \ldots, -\boldsymbol{r}_j, \ldots, \boldsymbol{r}_d)$. Then it holds by definition that $F(K_1, \boldsymbol{a}) = F(K_2, \boldsymbol{a})$ and $\sigma(K_1, \boldsymbol{a}) = -\sigma(K_2, \boldsymbol{a})$. □

This means that if two vertex cones are only mirrored on one hyperplane, their forward cones are the same and the sum of their growth factors equals 0. This allows us to remove neutralizing vertex cones in the data structure algorithm. We can do this by testing the position vectors in pairs.

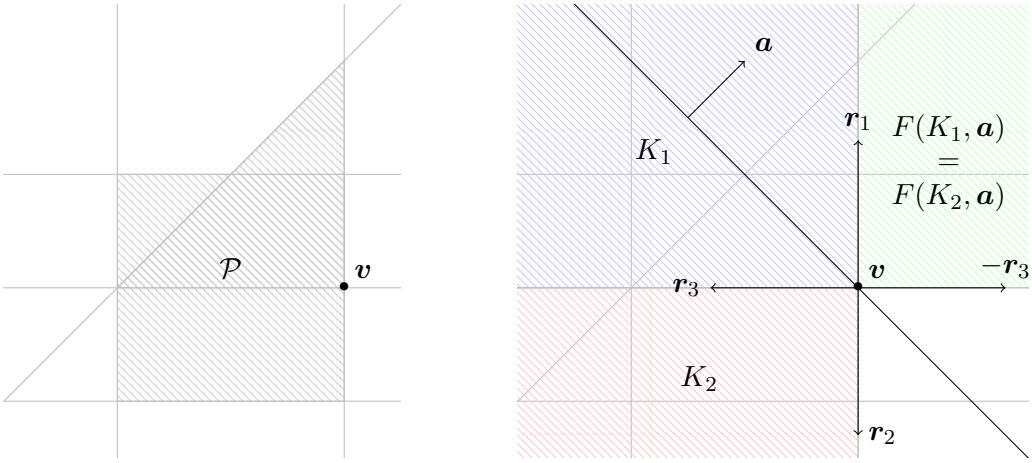
Example 5.4 Let $\mathcal{P}$ be the union of a triangle and a cube as depicted in Figure 5.1. Then the vertex cones at $\boldsymbol{v} := \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$ are given by $K_1 = \{\boldsymbol{v}\} + \operatorname{pos}(\boldsymbol{r}_1, \boldsymbol{r}_3)$ and $K_2 = \{\boldsymbol{v}\} + \operatorname{pos}(\boldsymbol{r}_2, \boldsymbol{r}_3)$, whereby

$$\boldsymbol{r}_1 := \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \boldsymbol{r}_2 := \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \quad \boldsymbol{r}_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

We look at the sweep direction $\boldsymbol{a} := \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Then

$$\langle \boldsymbol{a}, \boldsymbol{r}_1 \rangle = 1, \quad \langle \boldsymbol{a}, \boldsymbol{r}_2 \rangle = -1, \quad \langle \boldsymbol{a}, \boldsymbol{r}_3 \rangle = -1$$

and therefore $\sigma(K_1) = -1$ and $\sigma(K_2) = 1$. Furthermore $F(K_1, \boldsymbol{a}) = F(K_2, \boldsymbol{a}) = \{\boldsymbol{v}\} + \operatorname{pos}(\boldsymbol{r}_1, -\boldsymbol{r}_3)$. We have $\gamma_{K_1} = -\frac{1}{2} = -\gamma_{K_2}$ and therefore $\gamma_{\boldsymbol{v}} = \gamma_{K_1} + \gamma_{K_2} = 0$.



(a) The hyperplane arrangement.

(b) The vertex cones at $\boldsymbol{v}$.

Figure 5.1: The union of a triangle and a cube as an example for neutralizing forward cones.

5.2 Computational Results

In this section we will present the runtimes and the of tests over the unit cube in different dimensions and non-convex bodies in lower dimensions.
We did the calculations on a 64-bit elementary OS machine with a i5-2410M (2 threads, 2.30GHz) processor and 8 GB RAM. The implementation is written in python 2.7 and is using the libraries `numpy 1.13.1` and `pycddlib 2.0.0`.

5.2.1 Unit Cube Runtime

We calculated the volume of the unit cube $C^d := [0, 1]^d$ for dimension d from five to 13. Table 5.1 shows the runtimes of both the data structure algorithm and the sweep-plane volume algorithm. To determine the vertices, $\binom{2d}{d}$ linear equation systems of size $d \times d$ have to be solved if possible. We have $\binom{2d}{d}$ many linear equation systems, since the number of facets of the unit cube and therefore also the number of hyperplanes is $2d$. The number of vertices in the hyperplane arrangement approximately quadruples with each dimension step, whereas the number of vertices of the cube only doubles. The runtime for the sweep-plane volume algorithm seems to be doubling with the dimension, along with the number of vertices of the cube.
Since the unit cube is simple we have as many vertex cones as we have vertices. There are no degenerate vertices and so we do not have to regularize. Table 5.1 shows that most of the complexity lies in the data structure algorithm and the sweep-plane volume functions can be computed efficiently, at least for the unit cube.

Dimension	$\binom{\lvert\text{hyperplanes}\rvert}{d}$	Cube Vertices	Runtime (s)	
			Cell Decomposition	Volume
5	252	32	0.031	0.00019
6	924	64	0.079	0.00016
7	3432	128	0.25	0.00036
8	12870	256	0.93	0.0007
9	48620	512	3.81	0.0015
10	184756	1024	23.42	0.0052
11	705432	2048	82.10	0.0099
12	2704156	4096	372.33	0.0215
13	10400600	8192	1788.34	0.087
14	40116600	16384	39014.7	0.051

Table 5.1: Runtime for the volume computation of the unit cube.

5.2.2 Overlapping Simplices

We have also tested our algorithm on unions of random overlapping simplices in 3 dimensions. This case is interesting since for a small number of polytopes the inclusion-exclusion algorithm is faster than our algorithm, see Figure 5.2. For the inclusion-exclusion based algorithm the leading factor in the computational complexity is 2^P, whereby P denotes the number of polytopes. 2^P is the cardinality of the power set of polytopes. For every possible combination of polytopes the volume of their intersection has to be calculated. For few polytopes the inclusion-exclusion algorithm is faster than a single run of our algorithm. Especially in dimension two and three there are efficient algorithms to calculate the volume of a single polytope.
The runtimes for the sweep-plane volume computation of four overlapping simplices in different dimensions are displayed in Table 5.2. The simplices are randomly generated by taking the convex hull of $d + 1$ uniformly at random sampled points. One can see that the number of relevant vertices explodes with the dimension. The runtimes are way worse than the runtimes for the unit cube with comparable many vertices in higher dimensions. The reason for this is, that to calculate the vertices we solve linear equation systems. For the unit cube these matrices are sparse and therefore the equation systems are solved way faster.

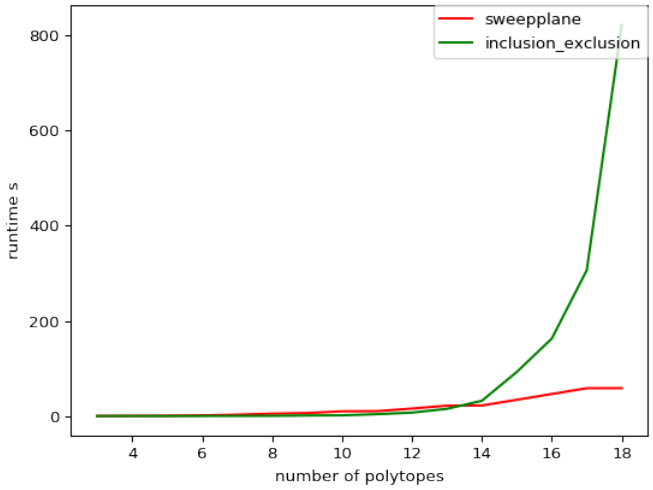


Figure 5.2: The runtime of the sweep-plane algorithm vs. the runtime of the inclusion-exclusion based algorithm on the union of randomly generated tetrahedron.

Dimension	$\binom{\lvert\text{hyperplanes}\rvert}{d}$	Vertices	Runtime (s)	
			Cell Decomposition	Volume
2	66	10	0.0097	0.003
3	560	101	0.1004	0.0088
4	4845	1297	2.915	0.14
5	42504	18093	114.8	3.682

Table 5.2: Runtime for the volume computation of 4 overlapping simplices in different dimensions.

5.2.3 Overlapping Polytopes Cut Analysis

As mentioned before, the motivation for the sweep-plane volume algorithm was in its use for finding approximative convex decomposition (ACD) cuts. The non-convex bodies in Figure 5.3 and Figure 5.4 originate from data corresponding to configurations in a compressor station. We have sampled 500 directions uniformly at random. For every direction we calculated the sweep-plane volume function, as well as the difference of the sweep-plane volumes of the convex hull and the non-convex body, and the derivative of the difference.
For the non-convex $\mathcal{P}$, its convex hull $\text{conv}(\mathcal{P})$ and a sampled sweep direction $\boldsymbol{a}$ we

have looked at the difference of the sweep-plane volume functions

$$\phi(\lambda, \boldsymbol{a}) := \mathrm{Vol}(\mathrm{conv}(\mathcal{P}) \cap H^-(\boldsymbol{a}, \lambda)) - \mathrm{Vol}(\mathcal{P} \cap H^-(\boldsymbol{a}, \lambda))$$

and its derivative ϕ'.
For the set T of sampled sweep-directions we have chosen the sweep-plane $H(\boldsymbol{a}, \lambda)$ at which the maximum $\max_{\boldsymbol{a} \in T} \max_{\lambda \in \mathbb{R}} \phi'(\lambda, \boldsymbol{a})$ is attained. For this sweep-plane the local change in the convexification error is maximal. This means that the convexification and the non-convex body are diverging the most at this sweep-plane. The maximum is attained at a turning point of ϕ. The graphs for the sweep-plane volumes of non-convex bodies and their convexication, the difference of them and the derivative of the difference are depicted on the left at Figure 5.3 and Figure 5.4.
This heuristic often lead to cuts that cut off very small pieces of the body. If the direction gives a supporting hyperplane at the entry vertex of the sweep, it often happens that the maximum $\max_{\lambda \in \mathbb{R}} \phi'(\lambda, \boldsymbol{a})$ is attained at that vertex. We developed the heuristic further by only allowing λ that are sufficiently bigger (smaller) than the $\lambda_{\boldsymbol{v}}$ ($\lambda_{\boldsymbol{w}}$) at which the sweep first (last) meets the bodies. Thereby we demand that both sides of the cut are sufficiently large.

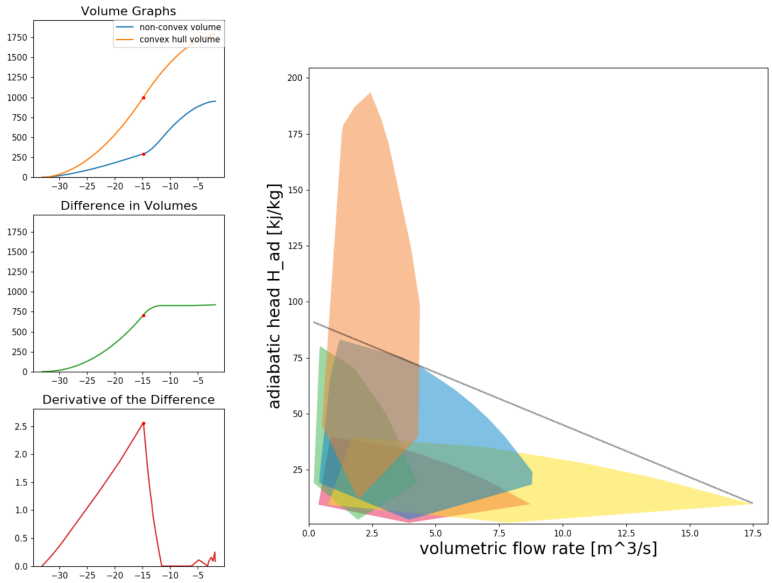


Figure 5.3: Finding an ACD cut for 5 overlapping polytopes obtained from gas net data.

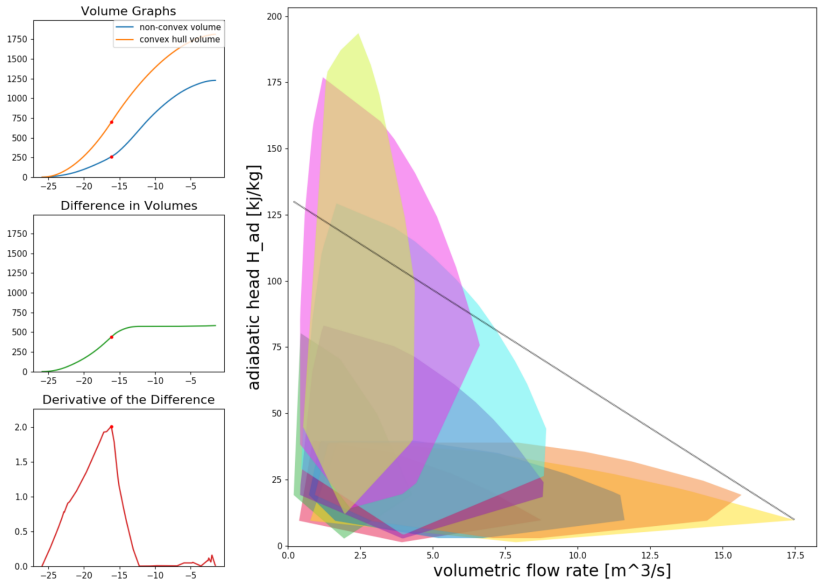


Figure 5.4: Finding an ACD cut for 9 overlapping polytopes obtained from gas net data.

6 Conclusion

We have developed an algorithm that computes the sweep-plane volume function for a union of polytopes in arbitrary dimension. Our algorithm is clearly based on the work of Bieri and Nef [BN83], but for the theoretical background we combined it with more recent theories from [Law] and [GH17].
In Section 2.2 we have learned that we can write the indicator function of a polytope as the (weighted) sum of the indicator functions of its forward cones. In Section 3.1 we have learned that we can write a union of polytopes as the union of cells of the underlying hyperplane arrangement. This way we could sum up the volumes of the active cells by using the conic decomposition. To determine the active cells we used position vectors, which are a powerful tool to put faces of a hyperplane arrangement into relation.
For the computation of the sweep-plane volume of the forward cones we used a well known volume formula for the simplex. In Section 4.1 we have proven that formula and adapted it to our case. This formula can only be used if the cone is simplicial. For the non-simplicial case we developed the regularization method in Section 2.3.
We took the algorithm from Bieri and Nef [BN83] and adapted it to our needs. For us it was especially important that we are able to compute the sweep-plane volume functions for many sweep directions. By separating the algorithm into two parts, we reduced the cost of calculating the volume functions for multiple directions drastically. This way we enabled it to calculate the sweep-plane volume functions for many different directions. We now hope to be able to put the algorithm to good use by finding decent ACD cuts through the heuristics we have developed.

Bibliography

[AG16] José Agapito and Leonor Godinho, *Cone decompositions of non-simple polytopes*, J. Symplectic Geom **14** (2016), no. 3, 737–766.

[BEF00] Benno Büeler, Andreas Enge, and Komei Fukuda, *Exact volume computation for polytopes: a practical study*, Polytopes—combinatorics and computation, Springer, 2000, pp. 131–154.

[BF10] Karl Bringmann and Tobias Friedrich, *Approximating the volume of unions and intersections of high-dimensional geometric objects*, Computational Geometry **43** (2010), no. 6-7, 601–610.

[BN82] Hanspeter Bieri and Walter Nef, *A recursive sweep-plane algorithm, determining all cells of a finite division of r d*, Computing **28** (1982), no. 3, 189–198.

[BN83] ———, *A sweep-plane algorithm for computing the volume of polyhedra represented in boolean form*, Linear Algebra and its Applications **52** (1983), 69–97.

[CV16] Ben Cousins and Santosh Vempala, *A practical volume algorithm*, Mathematical Programming Computation **8** (2016), no. 2, 133–160.

[FP96] Komei Fukuda and Alain Prodon, *Double description method revisited*, pp. 91–111, Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

[FP08] Christodoulos A Floudas and Panos M Pardalos, *Encyclopedia of optimization*, Springer Science & Business Media, 2008.

[GH17] Thomas Gerstner and Markus Holtz, *Algorithms for the cell enumeration and orthant decomposition of hyperplane arrangements*.

[Haa05] Christian Haase, *Polar decomposition and brion's theorem*, Contemporary Mathematics **374** (2005), 91–100.

[Knu05] Donald E. Knuth, *The art of computer programming, volume 4, fascicle 3: Generating all combinations and partitions*, Addison-Wesley Professional, 2005.

[Law] Jim Lawrence, *Polytope volume computation*, Mathematics of Computation.

[MRTT53] Theodore S Motzkin, Howard Raiffa, Gerald L Thompson, and Robert M Thrall, *The double description method.*

[S+04] Richard P Stanley et al., *An introduction to hyperplane arrangements*, Geometric combinatorics **13** (2004), 389–496.

[She67] Geoffrey C Shephard, *An elementary proof of gram's theorem for convex polytopes*, Canad. J. Math **19** (1967), 1214–1217.

[Ste66] P Stein, *A note on the volume of a simplex*, The American Mathematical Monthly **73** (1966), no. 3, 299–301.