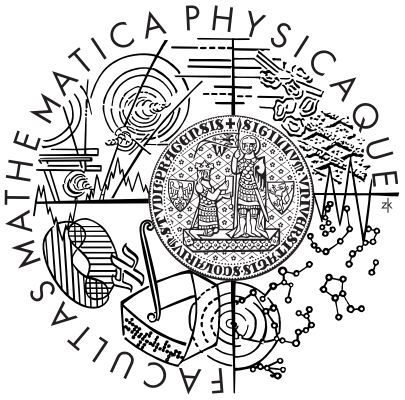
Charles University in Prague

Faculty of Mathematics and Physics
Department of Applied Mathematics




# On Algorithmic Characterization of Functional $D$-convex Hulls

**Doctoral Thesis**

Vojtěch Franěk

Supervisor: Prof. RNDr. Jiří Matoušek, DrSc.

Branch I4: Discrete Models and Algorithms

Charles University in Prague
Faculty of Mathematics and Physics
Department of Applied Mathematics

# On Algorithmic Characterization of Functional $D$-convex Hulls

Doctoral Thesis

Vojtěch Franěk
Supervised by Prof. RNDr. Jiří Matoušek, DrSc.
Prague, 2008

Hereby I declare that I have written this thesis on my own, and the references include all the sources of information I have exploited. I agree with lending of this thesis.

Vojtěch Franěk
May 27, 2008, Prague

# Contents

# Chapter 1

# Introduction

According to etymology dictionary [Ety], the word *convexity* comes from Latin word "convehere" (to bring together) and means "vaulted" or "arched". This meaning comes possibly from the idea of vaults meeting together at the point of a roof. The antonym is *concave* and stems from the Latin word "cavus" (hollow).

In mathematics, the convexity property is discussed in connection with sets and functions. If a set is convex, it means (very roughly spoken) that the set is connected and not too jagged (its surface has no "dips"). If a function is convex, then it forms a cup. To be more precise, a function $g$ is convex if and only if the set of points lying on or above the graph of the function (called the *epigraph* of $g$) is a convex set. A function $f$ is said to be concave if $-f$ is convex.

In this work, we speak about *directional convexity* (or $D$-convexity). This notion is used when considering convexity property in some directions only. Especially, we are interested in *functional $D$-convexity* (and, in particular, *functional $D$-convex hulls* of sets). This is some generalization of the directional convexity. It imposes some special properties which some functions must have on the sets involved. After a short recollection of classical convexity in Chapter 2, this shall be explained in detail in Chapters 3 and 4, where a short introduction to $D$-convexity and functional $D$-convexity is given. This is extended in Chapter 5 where a few more subtleties specific for our approach to the functional $D$-convexity are explained. In Chapter 6, we thoroughly explore the planar case. Chapter 7 states and proves one of the core results of our work — the Cutter Lemma. Chapter 8 exploits the results from previous chapters to complete the goal of this work — the algorithm for finding the functionally $D$-convex hull of a given finite point set in the plane. Complexity issues regarding the algorithm and the planar hull are discussed in Chapter 9. Notes on implementation of this algorithm are given

in Chapter 10. Chapter 11 explains the complications we met when trying to generalize the algorithm for higher dimensions.

### New results

The gist of this work lies in Chapters 7–9. We have found an algorithm computing the functionally $D$-convex hull of a finite set of points $A$ in the plane, and this algorithm finishes in $O(|D|^2|A|\log(|A|))$ time. Although the idea of the algorithm is not bound to planar case, a potential generalization of the algorithm for any dimension seems quite hard. On the other hand, we expect the Cutter Lemma (perhaps after a few modifications) to be a useful tool in studying the problem in general dimension (not only in the plane). Among other results, we would like to point out the linear size of the planar hull and that the size is not dependent on the size of the set of directions $D$.

Some earlier results of this work are available (in a substantially condensed form) in the article [FM]. It contains mostly the material from Chapter 7 (Cutter Lemma) and Chapter 8 (algorithm). The results from Chapter 9 (analysis of the complexity of both the hull ad the algorithm) are not present in the article. We should point out that there is one slight difference between the algorithm given here and the one in the quoted article. This improvement speeds up the algorithm, and simplifies the analysis of the complexity as well. We mention it in Chapter 8 and we explain it more thoroughly in Chapter 9.

# Chapter 2

# Convexity

This chapter recollects some basic notions about convexity. We mention them here, because we will compare them with their analogs in the world of directional convexity later on. The source of material for this chapter is the excellent textbook about discrete geometry written by Jiří Matoušek [Mat02] as well as the Alexander Barvinok's book on convexity [Bar02].

Because we are interested in the Euclidean space mainly, we will give the following definitions with respect to $\mathbf{R}^d$.

A set $C \subseteq \mathbf{R}^d$ is *convex* if for every two points $x, y \in C$ the whole segment $[x, y]$ is also contained in $C$ (cf. Figure 2.1). In other words, for every $t \in [0, 1]$, the point $tx + (1 - t)y$ belongs to $C$.

The *convex hull* of a set $X \subseteq \mathbf{R}^d$, denoted by $\mathrm{co}(X)$, is the intersection of all convex sets in $\mathbf{R}^d$ containing $X$:

$$\mathrm{co}(X) = \bigcap_{Y \in \mathcal{C}_X} Y, \text{ where } \mathcal{C}_X = \{Z | X \subseteq Z \subseteq \mathbf{R}^d \text{ and } Z \text{ is convex}\}.$$

Alternatively, we may define the convex hull of the set $X$ as the inclusion-minimal

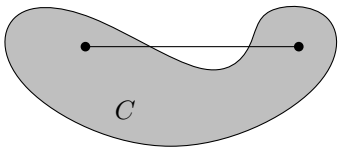



Figure 2.1: Such a situation is not allowed in a convex set.

convex set containing $X$:

$$\text{co}(X) = Y \in \mathcal{C}_X, \text{ where for every } Z \in \mathcal{C}_X : Y \subseteq Z.$$

The direct corollary of these definitions is the fact that a set $C$ is convex if it is the convex hull of itself:

$$C = \text{co}(C).$$

Another definition of the convex hull uses the notion of *convex combination*:

**Definition 2.1 (Convex combination)** *The* convex combination *of the points $x_1, x_2, \ldots, x_n$ is the expression*

$$\sum_{i=1}^{n} \alpha_i x_i$$

*where every $\alpha_i$ is a nonnegative number and $\sum_{i=1}^{n} \alpha_i = 1$.*

Back to the convex hull: a point $x$ belongs to $\text{co}(X)$ if and only if $x$ is a convex combination of some points from $X$:

$$\text{co}(X) = \{x | \exists n > 0 : x = \sum_{i=1}^{n} \alpha_i x_i\},$$

where $\sum_{i=1}^{n} \alpha_i = 1$ and for every $i : \alpha_i \geq 0$ and $x_i \in X$. Carathéodory [Car07] stated that in $\mathbf{R}^d$, it is sufficient to consider convex combinations involving at most $d+1$ points:

**Theorem 2.2 (Carathéodory's Theorem)** *Let $X \subseteq \mathbf{R}^d$. Then each point of* $\text{conv}(X)$ *is a convex combination of at most $d+1$ points of $X$:*

$$\text{co}(X) = \left\{ x \middle| x = \sum_{i=1}^{d+1} \alpha_i x_i \right\},$$

*where $\sum_{i=1}^{d+1} \alpha_i = 1$ and for every $i : \alpha_i \geq 0$ and $x_i \in X$.*

The number $d+1$ in this theorem is called *Carathéodory number*. Formally:

**Definition 2.3 (Carathéodory number)** *Carathéodory number is the minimal number $C_d$ for which*

$$\text{co}(X) = \left\{ x \middle| x = \sum_{i=1}^{C_d} \alpha_i x_i \right\}, \sum_{i=1}^{C_d} \alpha_i = 1, \forall i : \alpha_i \geq 0, x_i \in X$$

*holds for every $X \subseteq \mathbf{R}^d$.*

For example, Carathéodory's Theorem applied in the plane means that $\mathrm{conv}(X)$ is the union of all triangles with vertices at points of $X$.

Another basic result about convex sets is the fact that two disjoint convex sets are separable by a single hyperplane:

**Theorem 2.4 (Separation Theorem)** *Let $C, D \subseteq \mathbf{R}^d$ be disjoint convex sets. Then there is a hyperplane $h$ such that $C$ lies completely in one of the two closed halfspaces determined by $h$ and $D$ lies completely in the opposite one.*

# Chapter 3

# Directional Convexity

In this chapter, we introduce the notion of directional convexity and compare it with standard convexity. We also provide one section on brief history and applications of directional convexity.

## 3.1 Definition and Properties

Directional convexity is a generalization of the usual convexity discussed in the previous chapter. When studying directional convexity, we consider the convexity in some directions only:

**Definition 3.1** *Let $D$ be a set of vectors in $\mathbf{R}^d$. A set $C \subseteq \mathbf{R}^d$ is called a* $D$-convex set *(or* $D$-line-convex*) if for any two points $x_1, x_2 \in C$ holds: if the segment $[x_1, x_2]$ is parallel to some nonzero vector of $D$ then the whole segment $[x_1, x_2]$ is in $C$:*

$$[a, b] = \{\alpha a + (1 - \alpha)b | \alpha \in [0, 1]\} \subseteq C.$$

An alternative definition is:

**Definition 3.2** *Let $D$ be a set of nonzero vectors in $\mathbf{R}^d$. A* $D$-line *is a line parallel to a direction in $D$. A set $B \subseteq \mathbf{R}^d$ is* $D$-convex *(or* $D$-line-convex*) if every $D$-line intersects it in a possibly empty interval.*

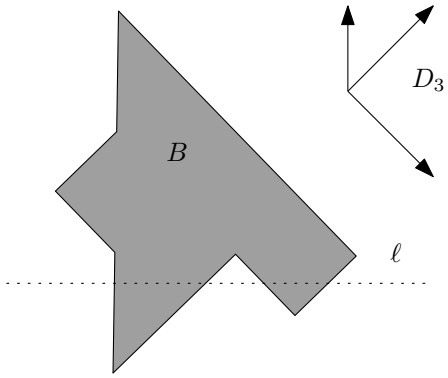



Figure 3.1: An example of a $D_3$-line-convex set.

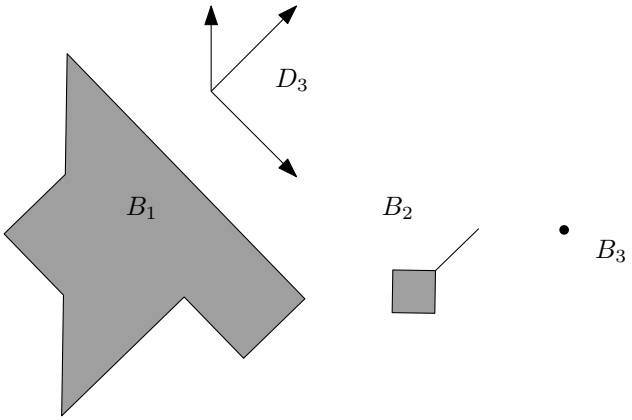



Figure 3.2: Another example of a $D_3$-line-convex set.

For example, let $B$ be the union of all $d$ coordinate axes and let $D$ be a set of $d$ orthogonal vectors parallel to these axes. Obviously, $B$ is not convex, but it is $D$-line-convex. More complex examples (for $d = 2$) are given in Figures 3.1 and 3.2. The set $B$ is not $D_1$-line-convex for any $D_1$ containing any vector parallel to $(1, 0)$ (because the $D_1$-line $\ell$ intersects the set $B$ in *two* intervals). On the other hand, let $D_3 = \{(0, 1), (1, 1), (1, -1)\}$, for example. Then the set $B$ *is* $D_3$-line-convex, as well as the set $B_1 \cup B_2 \cup B_3$. We see that (for general $D$) a $D$-line-convex set can have such strange shapes like $B_2$ ("polygon with a tail") and even more — a $D$-line-convex set can consist of several disjoint ($D$-line-convex) sets.

These were the examples of planar $D$-line-convex sets. To illustrate yet more possible shapes of $D$-line-convex sets, we present the last example, Figure 3.3. This is isometrical image of a three dimensional object consisting of sixteen unit

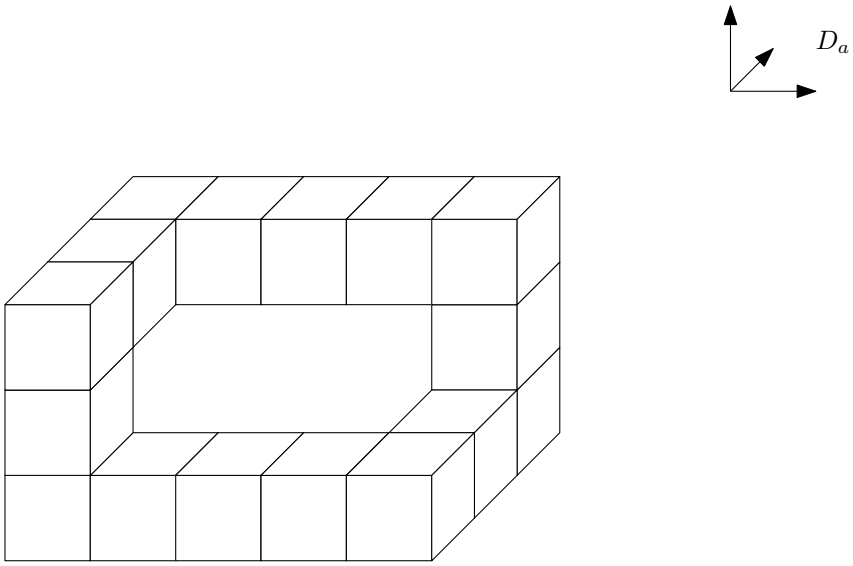



Figure 3.3: $D_a$-line-convex set with a hole.

cubes. Let the set $D_a$ be given by the coordinate axes:

$$D_a = \{(0,0,1),(0,1,0),(1,0,0)\}.$$

It is easy to check that the object in the figure is $D_a$-line-convex, although it has a hole inside.

In the literature, directional convexity is also called *D-convexity*, *restricted-orientation convexity* or *$\mathcal{O}$-convexity* (see, e.g., Schuierer and Wood [SW91a], Fink and Wood [FW96a]), *set-theoretical D-convexity* (Matoušek [Mat01]), and similarly.

For the special case when $D$ consists of $d$ orthogonal vectors, we use the word *separate convexity* (or rectilinear convexity, orthoconvexity, orthogonal convexity, etc.) instead of $D$-convexity. The special planar case of separate convexity is also called *staircase convexity* by some authors (see, e.g., Schuierer et al. [SRW91] or Schuierer and Wood [SW97]).

Let us compare "classical" convexity with $D$-convexity. We see that standard convexity is identical to $D$-convexity for $D$ equal to the whole $\mathbf{R}^d$, or for $D$ equal to the $d$-dimensional (hemi)sphere centered at the origin etc.

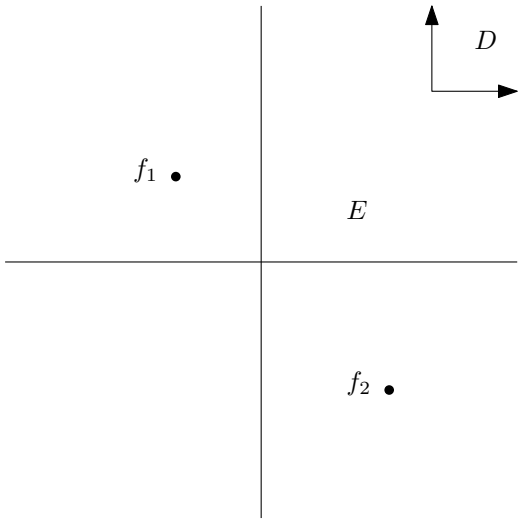



Figure 3.4: Unseparable $D$-line-convex sets.

As for Carathéodory number for the separate $D$-line convexity, Matoušek and Plecháč show in [MP98] a planar example illustrating that it is infinite:

$$A = \left\{ \left( \frac{1}{2i-1}, 1 + \frac{1}{i} \right) \middle| i = 1, 2, \ldots \right\} \cup \left\{ \left( \frac{1}{2i}, \frac{1}{1+i} \right) \middle| i = 1, 2, \ldots \right\} \cup \{(1,1)\}.$$

The separation theorem does not hold any more for $D$-line-convex sets, as can be easily seen from Figure 3.4. Here, the set $D$ is equal to the standard orthonormal basis of $\mathbf{R}^2$:

$$D = \{(0,1), (1,0)\}$$

and the two unseparable $D$-line-convex sets are the cross $E$ created by the two coordinate axes and the set $F$ consisting of two points $f_1$ and $f_2$ lying in opposite quadrants. It is obvious that no line can separate the two sets into different halfplanes.

Eugene Fink and Derick Wood introduced a *generalized halfspace* in connection with $D$-line convexity (in their article [FW96b], they call it an $\mathcal{O}$-halfspace; somewhat stronger definition is given in their report [FW96c]). A closed set is a generalized halfspace if its intersection with every $D$-line is empty, a ray, or a line. Generalized halfspaces have many properties similar to those of "regular" halfspaces. Still, as can be seen from our example, even those generalized halfspaces are not enough for separating some $D$-line-convex sets.

One may object that this is due to the fact that one of the sets is not connected. But as our last example shows, even connectedness does not help. The situation

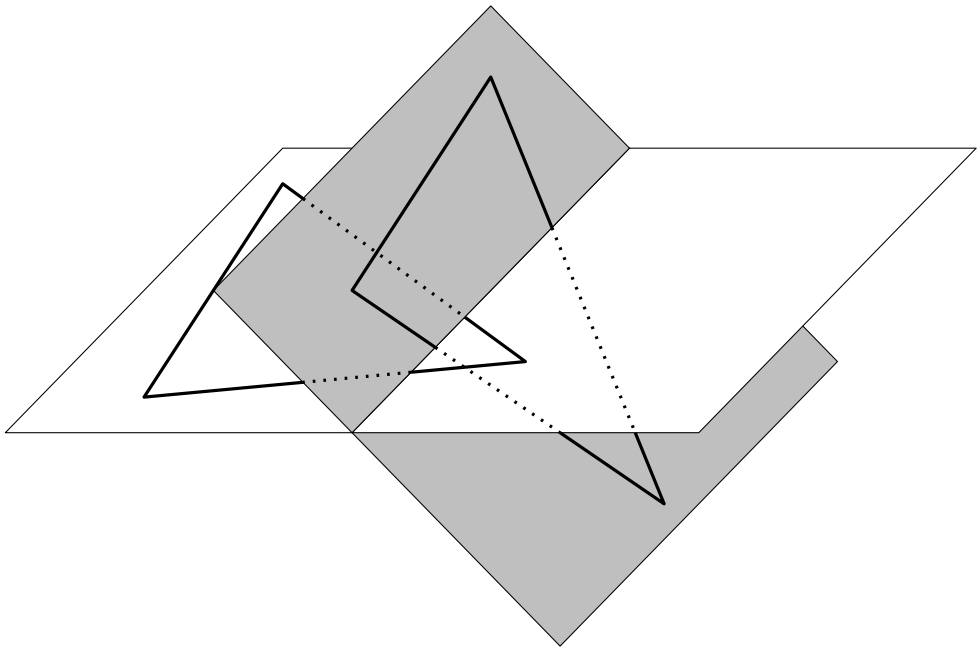


Figure 3.5: Two unseparable connected sets.

is depicted in Figure 3.5. We have two empty polygons, that are intertwined but disjoint. Both of them lie in different planes in $\mathbf{R}^3$. If we choose the set $D$ in such a way that no $D$-line is parallel to any of the planes, then every $D$-line intersects each of the two planes in a point, and hence every subset of any of the planes is $D$-line convex, because it is intersected by any $D$-line in a point, or not at all. On the other hand, there is no surface which could separate the two polygons, clearly.

Since the intersection of two intervals is an interval as well, $D$-line convex sets are closed under intersection. Then the $D$-line-convex hull of a set $X \subseteq \mathbf{R}^d$ (denoted by $\mathrm{co}_D(X)$) can be defined analogously to the convex hull as the intersection of all $D$-line-convex sets containing $X$:

$$\mathrm{co}_D(X) = \bigcap_{Y \in {}_D\mathcal{C}_X} Y, \text{ where } {}_D\mathcal{C}_X = \{Z | X \subseteq Z \subseteq \mathbf{R}^d \text{ and } Z \text{ is } D-\text{convex}\}.$$

It can be also defined as the inclusion-minimal $D$-convex set containig $X$:

$$\mathrm{co}_D(X) = Y \in {}_D\mathcal{C}_X, \text{ where for every } Z \in {}_D\mathcal{C}_X : Y \subseteq Z.$$

Again, a set $C$ is $D$-convex, if it is its own hull:

$$C = \mathrm{co}_D(C).$$

## 3.2 History, Applications, and Modifications

$D$-line convexity is quite favourite topic. It was studied by many authors in different contexts and from various views. Its fundamentals and some comparations with standard convexity are explained in Fink and Wood [FW96a]. They mention that the application of convexity theory to practical problems (see, e.g., Preparata and Shamos [PS85]) led to the exploration of nontraditional notions of convexity. They briefly summarize some of the history of study of this topic as follows: Ralf Hartmut Güting introduced the notion of restricted orientations in two dimensions in his doctoral thesis in 1983. In his later works, he investigated computational properties of polygons whose edges are parallel to the elements of some fixed finite set of lines, called an orientation set. Gregory J. E. Rawlins applied restricted orientations in his definition of two new types of generalized convexity, which he called $\mathcal{O}$-*convexity* and *strong* $\mathcal{O}$-*convexity*, in his doctoral thesis in 1987. $\mathcal{O}$-convexity, also called *restricted-orientation* convexity, was defined in terms of the intersection of a geometric object with lines parallel to the elements of a fixed orientation set $\mathcal{O}$. Together with Wood he showed that the notion of $\mathcal{O}$-convexity generalizes standard convexity and orthogonal convexity, and that the properties of $\mathcal{O}$-convex sets are similar to the properties of standard convex sets. Sven Schuierer continued their exploration and presented an extensive study of geometric and computational properties of $\mathcal{O}$-convex sets in his doctoral thesis in 1991.

In somewhat larger extent, Fink and Wood analyse restricted orientation convexity in their book [FW04]. These authors also give study of $\mathcal{O}$-halfspaces as the analogs of standard halfspaces in the theory of $\mathcal{O}$-convexity in the article [FW98a] and also in the report [FW96c].

There are also many variations of the basic concept of directional convexity. We have already mentioned separation convexity studied in the work [MP98]. *Strong restricted-orientation convexity* (see Fink and Wood [FW95], [FW98b], [FW03]) is similar to standard directional convexity, but is based on the conditions of containing some kind of parallelepipeds rather than simple segments. *NESW convexity* is mentioned as an example in Schuierer [SW91b]. Some other variations are orthogonal convexity (see, e.g., Nicholl et al. [NLLW83]), finitely oriented convexity (see, e.g., Widmayer et al. [WWW85]), or link convexity.

The study of nontraditional notions of visibility and convexity is a fruitful branch of geometry, which has found applications in VLSI design, graphics, motion planning, and other areas. Convexity finds quite natural applications in the field of visibility (e.g., in connection with the Art Gallery Problem, see, e.g., de Berg et al. [dBvKOS00]). Schuierer studied problems regarding re-

stricted orientation convexity and visibility in the report [SW91a] and in the article [SRW91]. Restricted orientation convexity and connectedness is pondered in Fink and Wood [FW96a].

We shall concentrate on yet another kind of $D$-convexity, namely functional $D$-convexity. Since this is the main topic of this work, we will go in more details than before, and this deserves a new chapter.

# Chapter 4

# Functional $D$-Convexity

In this chapter, we discuss the basics of *functional $D$-convexity*. We start with definitions, and after a brief section about motovation and background, we will restate some useful lemmas and theorems that are known on this topic. This will be followed by a talk about the famous four-point configuration. In the end of this chapter, we set the main goal of this work: an algorithm for computing the functional $D$-convex hull of a finite set.

## 4.1 Definitions

This section gives the definitions of functionally $D$-convex set and functionally $D$-convex hull.

Let $B$ be a $D$-line-convex set. A function $f : B \to \mathbf{R}$ is *$D$-convex* if, for any $x \in B$ and $v \in D$, the function $g(t) = f(x + tv)$ is a convex function of the real variable $t$ (the domain of $g$ is an interval in $\mathbf{R}$, because $B$ is $D$-line-convex).

A typical example of a $D$-convex function that is not convex in the usual sense is the function

$$f(x, y) = \max(x, 0) \max(y, 0).$$

For $D = \{(1, 0), (0, 1)\}$, this function is clearly $D$-convex. On the other hand, along the line $x + y = 1$, this function is equal to the concave function $x(1 - x)$ in the positive quadrant, and hence cannot be convex.

We denote by $\mathrm{co}_D(A)$ the $D$-line-convex hull of $A$, while $\mathrm{co}^D(A)$ denotes the

functional $D$-convex hull:

**Definition 4.1** *The functional $D$-convex hull of a compact set $A$ for a given set of directions $D$ is denoted by $\mathrm{co}^D(A)$ and defined as follows:*

$$\mathrm{co}^D(A) = \left\{ x \in \mathbf{R}^d | f(x) \leq \sup_{y \in A} f(y) \text{ for all } D\text{-convex } f : \mathbf{R}^d \to \mathbf{R} \right\}.$$

Once we have defined functionally $D$-convex hull, we may define functionally $D$-convex set in the most natural way:

**Definition 4.2** *A set $B$ is* functionally $D$-convex *if $B = \mathrm{co}^D(B)$.*

## 4.2 Motivation and Background

The motivation for studying functional $D$-convexity explains Matoušek and Plecháč [MP98]. It comes from the field of the calculus of variations. In particular, it is inspired by characterization of rank-one convex functions and quasiconvex functions. The following paragraphs introduce these concepts and explain the motivation. For the sake of brevity, we have decided to omit all technicalities and we do not delve into much detail. Precise formulations and further informations can be found, e.g., in the quoted references.

Let $\mathbf{M}^{m \times n}$ denote the space of all $m \times n$ matrices. A set of matrices $\mathcal{K} \subseteq \mathbf{M}^{m \times n}$ is *rank-one convex* if for every pair of matrices $A, B \in \mathcal{K}$, such that $A - B$ is a rank-one matrix, all convex combinations of these matrices belong to $\mathcal{K}$ as well:

$$\forall t \in [0,1] : tA + (1-t)B \in \mathcal{K}$$

(see, e.g., Cardaliaguet and Tahraoui [CT02].

Analogously, a function $f$ is *rank-one convex* if for every $t \in [0,1]$ the following inequality (known from the usual convexity) holds:

$$f(tA + (1-t)B) \leq tf(A) + (1-t)f(B),$$

provided that $A - B$ is an $m \times n$ matrix of rank-one (see, e.g., Pedegral and Šverák [PŠ98]).

Rank-one convexity can be considered as a special example of $D$-convexity. Indeed, let the set of directions be the cone of all rank-one matrices:

$$D = \{x \in \mathbf{M}^{m\times n} | \operatorname{rank} x \leq 1\}.$$

For this $D$, $D$-convexity equals rank-one convexity, obviously.

A function $f$ is *quasiconvex* (see, e.g., Morrey [Mor66]) if

$$\int_\Omega f(A + \nabla\varphi)dx \geq \int_\Omega f(A)dx$$

for any smooth function $\varphi : \Omega \to \mathbf{R}^m$ compactly supported in $\Omega$.

Quasiconvexity was introduced by Morrey [Mor52] as a condition when studying the closely related questions of lower semicontinuity and existence in the calculus of variations. Quasiconvexity and rank-one convexity have direct applications e.g. to the theory of nonlinear partial differential equations (see, e.g., Müller and Šverák [MŠ99]), but at the same time they are also at the root of many fundamental problems in physics: e.g., Müller [Mül98] refers to quasiconvexity as a characterization of the stored-density energy function that describes the properties of a crystalline microstructure.

Although it was even conjectured that quasiconvexity and rank-one convexity are equivalent notions, Šverák [Šve92] proved by his ingenious counterexample that for $n \geq 2$ and $m \geq 3$, rank-one convexity does not imply quasiconvexity. (For $n = 1$ or $m = 1$, these two notions are equivalent, but for $m = 2$ and $n \geq 2$ the problem is still open.)

Because quasiconvexity plays a similar role in the study of vectorial variational problems (or nonlinear partial differential equations) as usual convexity does for the scalar ones, it is natural to try to use quasiconvexity in similar ways as standard convexity. In particular, quasiconvex envelopes are important tools when studying problems that are not weakly lower semicontinuous, and hence cannot be solved using the standard methods of the calculus of variations. Using quasiconvex functions, one may define the *functionally quasiconvex hull* of a set $A$ analogously as the functionally $D$-convex hull in Definition 4.1:

$$\operatorname{co}^{qc}(A) = \left\{ x \in \mathbf{M}^{m\times n} | f(x) \leq \sup_{y\in A} f(y) \text{ for all quasiconvex } f : \mathbf{M}^{m\times n} \to \mathbf{R} \right\}.$$

In other words, the functionally quasiconvex hull of a set $A$ is the set of those points which cannot be separated from $A$ by quasiconvex functions (see, e.g., Kirchheim et al. [KMŠ02]). Unfortunately, no useful description of quasiconvex functions is known. However, any quasiconvex function is rank-one convex and

hence $\mathrm{co}^D(A) \subseteq \mathrm{co}^{qc}(A)$ for $D$ defined as above as the set of all rank-one matrices. Here is the place where the present investigation of functional $D$-convexity comes into play, since the functionally $D$-convex hull may be readily used as a reasonable first approximation of the quasiconvex hull. Since the computation of rank-one convex hull is a difficult task and we are not aware of any efficient algorithm (even an approximate one), finding inner and outer approximations of quasiconvex hulls is an important task. In the beginning, this work was intended to contribute to this somehow. However, due to the complexity of the problem we restricted ourselves to the finite case and mostly to the plane, which brought the focus into somewhat different area.

## 4.3 Propositions

Functional $D$-convexity was studied by Matoušek and Plecháč ([MP98], [Mat01]). In this section, we recollect some of the interesting theorems proved in [MP98]. We will use most of them in our proofs.

**Proposition 4.3** *A compact set $B \subset \mathbf{R}^d$ is functionally $D$-convex if and only if it is the zero set of a nonnegative $D$-convex function $f : \mathbf{R}^d \to [0, \infty)$, i.e.:*

$$B = f^{-1}(0).$$

Proposition 4.3 is the one of extreme importance. It gives us an alternative definition of functional $D$-convexity that in many situations will be far more convenient than Definition 4.2. We will also refer to this definition very often.

Together with the fact that the pointwise maximum of $D$-convex functions is also a $D$-convex function (formally proved by Corollary 5.2 later), Proposition 4.3 also helps us to show that the functional $D$-convexity property is closed under intersection. Indeed, if $A_1$ and $A_2$ are functionally $D$-convex compact sets and $f_1$, respesctively $f_2$, are nonnegative $D$-convex functions witnessing the functional $D$-convexity of these sets (i.e., $A_1 = f_1^{-1}(0)$ and $A_2 = f_2^{-1}(0)$), their pointwise maximum is a nonnegative $D$-convex function witnessing functional $D$-convexity of the intersection of the sets:

$$(A_1 \cap A_2) = (\max(f_1, f_2))^{-1}(0).$$

This allows us to define the functional $D$-convex hull of a compact set $A \subseteq \mathbf{R}^d$ analogously to convex hull and $D$-line convex hull as the intersection of all functionally $D$-convex sets containing $A$.

Let us continue with the theorems from Matoušek and Plecháč [MP98]:

**Proposition 4.4** *For any $D$ and any $A$, we have*

$$\mathrm{co}_D(A) \subseteq \mathrm{co}^D(A).$$

In other words, functional $D$-convexity implies $D$-line-convexity. This helps us to check whether a set is functionally $D$-convex or not: as soon as we can prove that the set is not $D$-convex, we know that it is not functionally $D$-convex.

**Proposition 4.5** *For $A$ closed and $D = \mathbf{R}^d$ (i.e., for the usual convexity) the $D$-line-convex hull and the functional $D$-convex hull coincide:*

$$\mathrm{co}_D(A) = \mathrm{co}^D(A).$$

**Proposition 4.6** *A point $x$ of a set $A$ is called a* $D$-extremal point *of $A$ if there exists no segment $s \subseteq A$ parallel to some nonzero vector $v \in D$ and containing $x$ as its interior point. Let $A, B$ be compact sets, and suppose that all $D$-extremal points of $B$ belong to $A$. Then $B$ is contained within the functional $D$-convex hull of $A$:*

$$B \subseteq \mathrm{co}^D(A).$$

*In particular, any compact functionally $D$-convex set is the functional $D$-convex hull of the set of all its $D$-extremal points.*

Proposition 4.6 is so called *Krein–Milman type theorem*. The Krein–Milman theorem says that compact convex set is the closed convex hull of its extreme points (see, e.g., Royden [Roy88]; for the original version of the statement refer to Krein and Milman [KM40]).

**Proposition 4.7** *Let $B_1, B_2$ be disjoint compact sets with $B_1 \cup B_2$ being a functionally $D$-convex set. Then both sets $B_1$ and $B_2$ are functionally $D$-convex as well:*

$$B_1 = \mathrm{co}^D(B_1), B_2 = \mathrm{co}^D(B_2).$$

**Proposition 4.8** *Let $A \subset \mathbf{R}^d$ be contained in a functionally $D$-convex set $B$, which is a disjoint union of compact sets $B_1, \ldots, B_k$. Then the functional $D$-convex hull of $A$ can be constructed from the functional $D$-convex hulls of these subsets in the following way:*

$$\mathrm{co}^D(A) = \bigcup_{i=1}^{k} \mathrm{co}^D(A \cap B_i).$$

To be precise, Proposition 4.8 is claimed in Matoušek and Plecháč [MP98] as Corollary 2.9 without any further explanation. But as was pointed out by Bernd Kirchheim, this result probably cannot be considered as a immediate consequence of Proposition 4.7 (Proposition 2.8 in the original paper). Nevertheless, this proposition holds and a full proof is given in Matoušek [Mat01] (the proposition was independently proved by Kirchheim [Kir99]).

## 4.4 The Four-Point Configuration

Matoušek and Plecháč [MP98] also mention an important example (discovered indepenently in various contexts by Scheffer [Sch74], Aumann and Hart [AH86], Nesi and Milton [NM91], Tartar [Tar93b], and Tarabusi [Tar93a]). The example shows that the functional $D$-convex hull may be much larger than the $D$-line-convex hull in general. As this example is crucial, we show it here as well.

The example is called the *Four-Point Configuration* and we denote it by $\mathcal{C}_4$. In this example, $d = 2$, $D = \{(0,1),(1,0)\}$ and $A = \{a_1, a_2, a_3, a_4\}$ as in Figure 4.1. The $D$-line-convex hull $\mathrm{co}_D(A)$ is equal to the four-point set $A$, while the functional $D$-convex hull $\mathrm{co}^D(A)$ consists of the four segments $a_ib_i$ and the square $b_1b_2b_3b_4$. Let us denote this "square with four tails" by $C$:

$$C = \mathrm{co}^D(A).$$

Although the same proof is given in Matoušek and Plecháč [MP98], we repeat the proof for the inclusion $C \subseteq \mathrm{co}^D(A)$, since it is a nice exercise introducing the concept of functional $D$-convex hull.

**Proof of $C \subseteq \mathrm{co}^D(A)$:** Let $f$ be a $D$-convex function, let $M_A = \max_i f(a_i)$ and $M_B = \max_i f(b_i)$. If $M_B > M_A$, let $i$ be such that $M_B = f(b_i)$. Looking at the $D$-line $a_ib_{i-1}$ (where $b_0 = b_4$), we see a contradiction with the $D$-convexity of $f$, since

$$f(a_i) \le M_A < M_B = f(b_i) = M_B \ge f(b_{i-1}).$$

We have shown that for all $i \in \{1,2,3,4\}$:

$$f(b_i) \le M_B \le M_A = \max_i f(a_i) = \sup_{y\in A} f(y).$$

Since this is true for every $D$-convex function $f$, we have (see Definition 4.1)

$$B = \{b_1, b_2, b_3, b_4\} \subset \mathrm{co}^D(A).$$

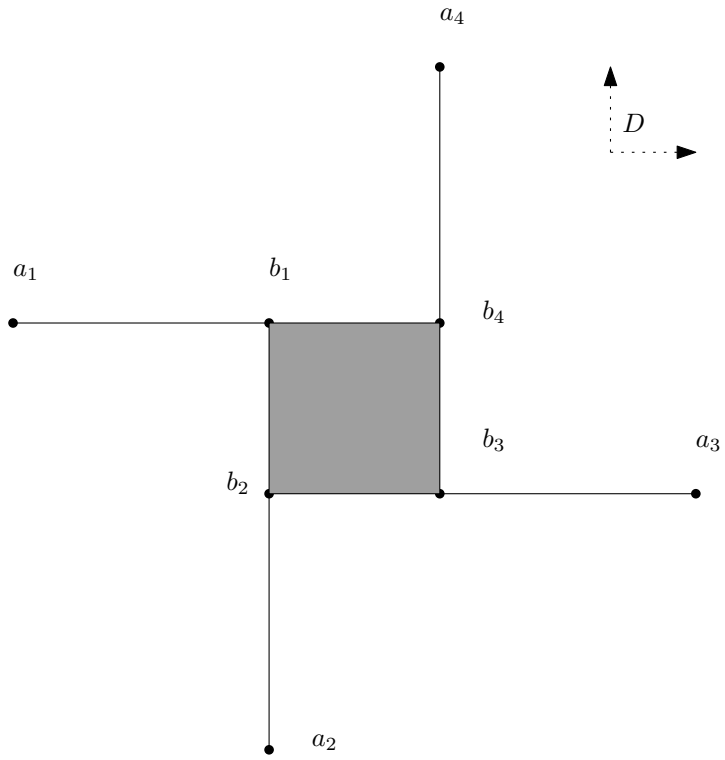



Figure 4.1: The $\mathcal{C}_4$ configuration.

For the following inclusion, we use Proposition 4.4 and we are done:

$$C = \mathrm{co}_D(A \cup B) \subseteq \mathrm{co}^D(A \cup B) = \mathrm{co}^D(A).$$

□

The $\mathcal{C}_4$ configuration is interesting since it is a generic nontrivial set without $D$-connection. Let us explain this. Let $D$ be fixed. Let us call a (finite) set $A$ *nontrivial* if it is not its functionally $D$-convex hull, i.e., $\mathrm{co}^D(A) \neq A$. A simple reason for nontriviality of $A$ can be the fact that it contains two distinct points $x, y$ such that the line $xy$ is parellel to a vector in $D$. In this case we say that $A$ *has a $D$-connection*. The four point configuration $\mathcal{C}_4$ is an example of a nontrivial set without any $D$-connection. Note that it is *generic nontrivial*, meaning that any sufficiently small perturbation of the points of the set yields a nontrivial set, again.

The $\mathcal{C}_4$ configuration has also a unique role in the theory of *separate convexity*, i.e., the situation where $D$ consists of $d$ orthogonal vectors. Since the separate convexity property is preserved by an affine transformation, separate convexity

handles all the cases of $|D| = d$, where the vectors of $D$ are linearly independent. Matoušek and Plecháč [MP98] showed that for functional separate convexity in the plane, any nontrivial set without a $D$-connection contains a copy of $\mathcal{C}_4$ or its mirror reflection, and that the Carathéodory number for functional separate convexity in the plane is 5 (which contrasts with the fact that the Carathéodory number for separate $D$-line convexity is infinite as we already mentioned before). On the other hand, Matoušek [Mat01] proved that in dimensions 3 and higher, there is no such simple description of nontrivial sets for functional separate convexity. Inclusion-minimal generic nontrivial configurations can be arbitrarily large (he also gives an inductive method for their construction), and, as a consequence, Carathéodory number for functional separate convexity in dimension 3 and higher is infinite.

## 4.5 Functional $D$-convex Hull of a Finite Set

In our work, we focus on algorithm for computing the functional $D$-convex hull of a finite point set $A$. The one given in Matoušek and Plecháč [MP98] covers the situation for separate convexity, that is, for $D$ consisting of exactly $d$ linearly independent vectors. Because it is not too complicated and because we will use somewhat similar ideas later, we describe the algorithm here as well, namely in the end of this chapter.

Our goal is to find an algorithm for any finite $D$ consisting of *at least* $d$ vectors, where none of them is a multiple of another:

$$|D| \geq d.$$

The alternative definition of the functionally $D$-convex hull of a set $A$ talks about all nonnegative $D$-convex functions vanishing on $A$, but it is known (see Matoušek [Mat01]) that it suffices to consider one particular $D$-convex function. Namely, $\mathrm{co}^D(A)$ equals the zero set of the $D$-convex envelope of the function $\delta_A$, where $\delta_A(x)$ is the (Euclidean) distance of $x$ from $A$, and the *$D$-convex envelope* (or *$D$-convexification*) $C_D f$ of a function $f$ is the pointwise supremum of all $D$-convex functions $g$ that satisfy $g \leq f$. Formally, the $D$-convexification $C_D f$ of a function $f : \mathbf{R}^d \to \mathbf{R}$ is given by

$$C_D f(x) = \sup_{g \in \mathcal{G}_f} \{g(x)\}, \text{ where}$$

$$\mathcal{G}_f = \{g | g : \mathbf{R}^d \to \mathbf{R} \text{ is } D-\text{convex and } \forall x \in \mathbf{R}^d : g(x) \leq f(x)\}.$$

The functionally $D$-convex hull $\mathrm{co}^D(A)$ of a set $A$ can be defined as

$$\mathrm{co}^D(A) = (C_D\delta_A)^{-1}(0),$$

where $\delta_A(x) = \inf_{y\in A} \|x - y\|$.

This suggests a possible approach for computing an approximation of $\mathrm{co}^D(A)$: compute the $D$-convexification of $\delta_A$ approximately by an iterative algorithm, and look where it is very close to 0. Unfortunately, an example in Matoušek [Mat01] shows that in order to approximate $\mathrm{co}^D(A)$ in this way, we need to compute the $D$-convexification of $\delta_A$ with unrealistically high precision (exponential in $|A|$) in some cases. Hence a different kind of algorithm is needed, based on better understanding of the properties of the D-convex hull. It will be given in Chapter 8 for the planar case.

### 4.5.1 Simple Set of Directions

To get rid of the sometimes annoying condition of not being a multiple of another vector, we will use the following definition:

**Definition 4.9** *Let $D$ be a set of directions. If $D$ consists of nonzero vectors and none of them is a multiple of another, we call it a* simple set of directions.

**Observation 4.10** *Let $D$ be a set of directions. Let us create a simple set of directions $D_s \subseteq D$ by reducing the set $D$ in the following way:*

- *If there is the zero vector in the set, remove it.*
- *While there are parallel vectors in the set, remove any one of them.*

*The directional convexity restrictions given by this reduced set $D_s$ are exactly the same as the ones given by the initial set of directions $D$.*

**Proof:** The proof follows directly from the Definitions 3.1 and 3.2. □

### 4.5.2 The Separate Case

In this subsection, we describe the algorithm for computing functional separately convex hull of a finite set, as it is given in Matoušek and Plecháč [MP98].

**Definition 4.11** *For any finite set $A \subset \mathbf{R}^d$, let its* grid *be defined as the set of all the "coordinate intersections" of all its points:*

$$\text{grid } (A) = x_1(A) \times x_2(A) \times \ldots \times x_d(A),$$

*where $x_i(A)$ stands for the set of all the $i$-th coordinates of all the points from the set $A$:*

$$x_i(A) = \{x_i(a) | a \in A\},$$

*where $x_i(a)$ denotes the $i$-th coordinate of $a$.*

**Definition 4.12** *Let $G = \text{grid } (A)$ be the grid of a finite set $A \subset \mathbf{R}^d$ and let $a \in G$ be a point of the grid. We denote by $a^{i+}$ and $a^{i-}$ its successor (resp. predecessor) along the $i$-th coordinate in $G$. More formally: if $a = (x_1(a), x_2(a), \ldots, x_d(a))$, then*

$$a^{i+} = (x_1(a), x_2(a), \ldots, x_{i-1}(a), z^+, x_{i+1}(a), \ldots, x_d(a)),$$

$$a^{i-} = (x_1(a), x_2(a), \ldots, x_{i-1}(a), z^-, x_{i+1}(a), \ldots, x_d(a)),$$

*where $z^+$ is the successor and $z^-$ is the predecessor of $x_i(a)$ in $x_i(A)$. The "border" points of $G$ are missing these $a^{i-}$ or $a^{i+}$ for some or all $i$, of course.*

**Definition 4.13** *Let $G = \text{grid } (B)$ be the grid of a finite set $B \subset \mathbf{R}^d$ and let $e \in G$ be a point of the grid. This point $e$ is called* grid-extremal *point of $B$ if for each $i = 1, 2, \ldots, d$, at least one of the points $e^{i+}, e^{i-}$ does not exist or does not belong to $B$.*

**Definition 4.14** *An* elementary box *on a grid $G$ is a Cartesian product of the form*

$$I_1 \times I_2 \times \ldots \times I_d,$$

*where for each $i = 1, 2, \ldots, d$, the set $I_i$ has either the form $\{x_i\}$ for some $x_i \in x_i(G)$, or the form $[x_i(a), x_i(a^{i+})]$ for an $a \in G$. Again, this is a natural definition: the grid dissects the space $\mathbf{R}^d$ into some $d$-dimensional bounded boxes (and some unbounded sets, which are of no importance now). Elementary box is any face (in the polyhedral sense) of any such box. Thus, elementary box may have dimension zero (if it is a point), one (if it is a segment), etc. up to $(d-1)$ (being a facet of such a box) or even $d$ (if it is equal to such a box).*

**Definition 4.15** *Let $B \subseteq G$ be a subset of a grid $G$. The* box complex union *of $B$ in $G$, denoted by $|\mathcal{BC}(B)|$, is the union of all elementary boxes on $G$ with all corners belonging to $B$.*

**Theorem 4.16 (The Algorithm)** *Let $A \subset \mathbf{R}^d$ be a finite set. The following alogrithm computes its functional separately convex hull.*

1. *Let $B_0$ be a subset of the grid $G = \text{grid } (A)$ containing $A$:*

$$A \subseteq B_0 \subseteq G$$

   *such that its box complex union $|\mathcal{BC}(B_0)|$ is functionally separately convex. (The most simple way — but probably not the most efficient one in general — how to achieve this is taking the whole grid: $B_0 = G$.)*

2. *Suppose that some $B_j$ has already been computed. If $B_j$ has an grid-extremal point $e \notin A$ outside of $A$, put*

$$B_{j+1} = B_j \setminus \{e\}$$

   *and repeat this step.*

   *If all extremal points of $B_j$ belong to $A$, then the box complex union $|\mathcal{BC}(B)|$ of $B = B_j$ is the desired functional separately convex hull of $A$.*

A simplified description (and implementation aspects as well) of the algorithm is given in diploma thesis by Boris Letocha [Let99]:

1. Mark all the points of $A$ by black color.
2. Mark all the remaining points of $G$ by white color.
3. If there is a white point having at most one neighbour along all coordinate axes, remove it and repeat this step.

In the end, the set of all the black or white points corresponds to the final set $B$ from the Algorithm 4.16. The process of the algorithm is illustrated in Figure 4.2 on a planar case.

There are six points of the set $A$ marked black on every diagram. The remaining points of the set $B_j$ are marked white. The diagram (a) shows the initial state for $B_0 = G$. There are many ways how to proceed, all lead to the same (correct)

result. As an example, the order in which the first six points could be removed is indicated by the numbers. The second diagram (b) shows the situation after several (21, to be precise) iterations of the algorithm have been made. There are still some white points to remove, for example the one marked by the arrow. Diagram (c) is the final stage and diagram (d) shows the functional separately convex hull of $A$ reconstructed from the box complex union of the set of all the ten points which still remained marked after the algorithm had finished.

If we look at the algorithm closely, we see that in every step we "bite off a corner". In other words, we position a box-shaped "cutter" at a "free" convex vertex and start removing the matter from the set $|\mathcal{BC}(B_j)|$ until the cutter is "blocked". As one can see, e.g., on the diagram (b) of Figure 4.2, this biting is performed locally, i.e. the cutting off of the vertex marked by the arrow does not affect for example the point lying to the right and above of it.

This approach we will use in our planar algortihm coping with any finite set of directions $D$, too. In this case, we do not have any reasonable analogy of the grid used for separate convexity. Thus, in every step, we shall find a free corner, construct a suitable wedge-shaped cutter, and decide how much of the current approximation of the functional $D$-convex hull we can safely remove in this step. All the details of this procedure are explained thoroughly in Section 8.

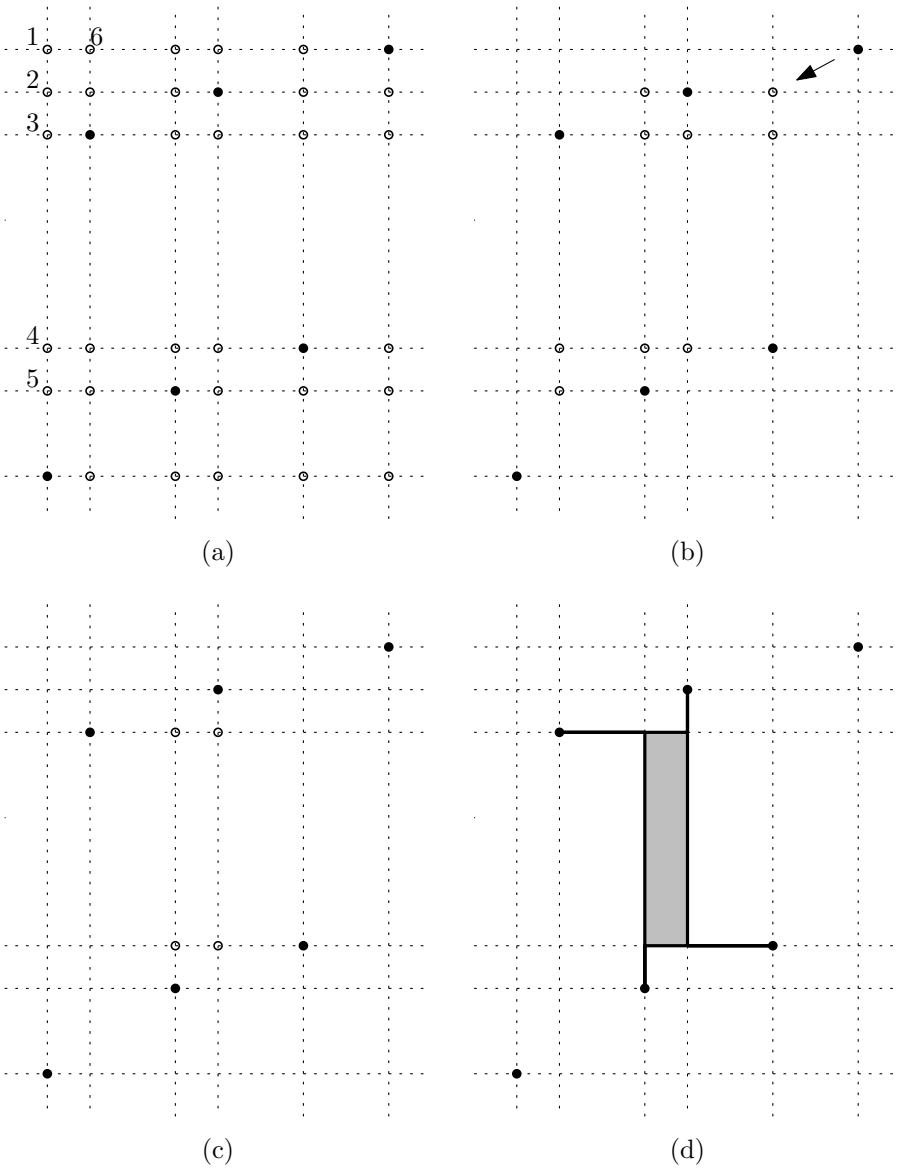



Figure 4.2: The planar algorithm for computing functional separately convex hull.

# Chapter 5

# Preliminaries

In this chapter, we define a few notions and state and prove a few simple facts regarding $D$-convexity and functional $D$-convexity. We will use them in the following chapters.

## 5.1 Definitions

Here we define $(D, A)$-polygon and all the notions that lead to this definition or are related to it.

Let $D$ be a finite set and suppose that its linear span is $\mathbf{R}^d$. A *$D$-hyperplane* is a translation of the linear span of some $d-1$ linearly independent members of $D$.

As before, a *$D$-line* is a line parallel to some nonzero vector from $D$. A *$D$-ray* is a ray parallel to a $D$-line.

A *cell* of an arrangement of a finite set of hyperplanes means the closure of a face of the arrangement. Cells can be $0, 1, \ldots, d$ dimensional. Following the usual notation, we call the 0-dimensional cells *vertices*, 1-dimensional cells are *edges* and $d$-dimensional cells are *facets*.

A *polyhedron* is the union of a finite number of cells of the arrangement of a finite set of hyperplanes. A *polytope* is a bounded polyhedron. (Note that every polyhedron is closed.) A *polygon* is a planar polytope.

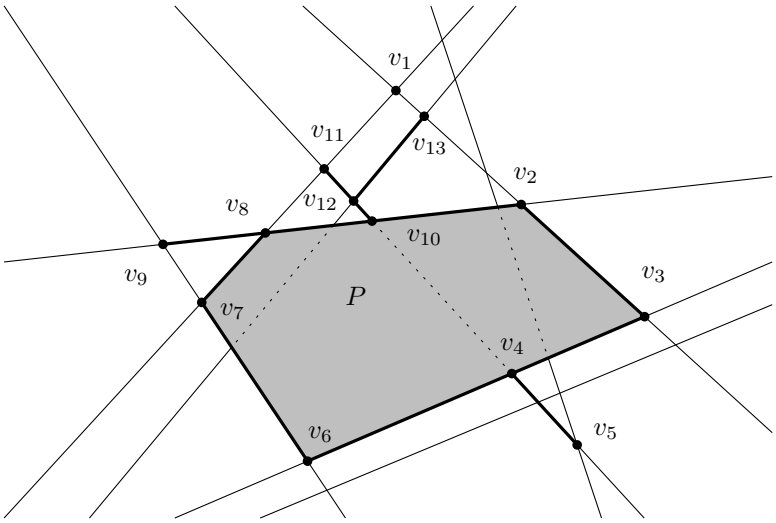



Figure 5.1: An illustration to the definitions of edge atoms, edges, and vertices of a polyhedron.

An *edge atom* of polyhedron $P$ is an edge $e$ of the underlying arrangement belonging to $P$ such that at least one of the arrangement facets incident to $e$ does not belong to $P$. An *edge* of $P$ is the union of one or more edge atoms of $P$ forming a segment $s$ which is inclusion-maximal (i.e., no more edge atoms of $P$ adjacent to $s$ lie on the same line as $s$) and the intersection with the union of all the remaining edge atoms of $P$ is a subset of the set of both endpoints of $s$ (it contains either both of them, or only one of them, or none). The endpoints of edges of $P$ as well as the isolated vertices of the underlying arrangement that belong to $P$ are called *vertices of* $P$.

As an example, on Figure 5.1 we have polygon $P$ consisting of one isolated vertex $v_1$, the gray area and the four fat segments $[v_4v_5], [v_8v_9], [v_{10}v_{11}]$ and $[v_{12}v_{13}]$. The segments $[v_4v_5], [v_8v_9]$ and $[v_{12}v_{13}]$ are edges of $P$. The edge $[v_4v_5]$ consists of two edge atoms, the other two edges are formed by one single edge atom. The segment $[v_{10}v_{11}]$ is not an edge of $P$, because the edge atom $[v_{12}v_{13}]$ intersects it in its interior. The segment $v_4v_{10}$ is not an edge atom of $P$, because both the incident facets belong to $P$. All in all, the polygon $P$ has thirteen vertices and twelve edges.

One special case of a polyhedron is *$D$-polyhedron*. We define it similarly to polyhedron. $D$-polyhedron is the union of some cells of the arrangement of a finite set $G$ of $D$-hyperplanes, a *$D$-polytope* is a bounded $D$-polyhedron, and a *$D$-polygon* is a planar $D$-polytope.

Let $A \subset \mathbf{R}^d$ be a finite set. A $(D, A)$-*hyperplane* is a $D$-hyperplane passing through a point of $A$. A $(D, A)$-*line* is a line parallel to a nonzero direction from $D$ and passing through a point from $A$. A $(D, A)$-*ray* is a ray parallel to a $(D, A)$ line such that it starts in an intersection with another $(D, A)$-line. A $(D, A)$-*polyhedron* is a $D$-polyhedron such that all the hyperplanes in the set $G$ are $(D, A)$-hyperplanes. (Simply, a $(D, A)$-polyhedron is the union of some cells of a $(D, A)$-arrangement, i.e., the arrangement of a finite set of $(D, A)$-hyperplanes.)

Again, a $(D, A)$-*polytope* is a bounded $(D, A)$-polyhedron, and a $(D, A)$-*polygon* is a planar $(D, A)$-polytope.

On Figure 5.2, we see some examples of planar $(D, A)$-polyhedra. The sets generating the $(D, A)$-arrangement (an arrangement defined by all the $(D, A)$-lines) are $D = \{(0, 1), (1, 0), (1, -1)\}$ and $A = \{a_1, a_2, a_3, a_4\}$. The point $a_2$ itself is $(D, A)$-polygon, as well as the gray set $Z$ in the middle or the three thick segments $S$ on the left. Any union of any of these sets is $(D, A)$-polygon, too. The (unbounded) cell $P$ in the upper right corner of the figure is a planar $(D, A)$-polyhedron, and so is the union of this planar $(D, A)$-polyhedron with any of the $(D, A)$-polygons mentioned before.

Let $x$ be a vertex of a polyhedron $P$ (i.e., $x \in P$ is a 0-dimensional cell of the underlying arrangement). The vertex $x$ is *convex* if there is an open neighbourhood $\Omega(x)$ and a hyperplane $h \ni x$ such that $(P \cap \Omega(x)) \cap h^- = \{x\}$ , where $h^-$ is one of the closed halfspaces given by $h$.

## 5.2 Facts

In this section, we start with a simple fact about convex functions. This will be followed by a corollary regarding $D$-convex functions. It will be used in a theorem concerning limit behaviour of the functionally $D$-convex hull. The chapter will be concluded by an easy topological observation.

**Lemma 5.1** *Let $f_1, f_2$ be any convex functions. Then their pointwise maximum is also convex:*

$$h(x) = \max(f_1(x), f_2(x)).$$

*In particular, for $f_2(x) = C$ being a constant function, the function*

$$\max(f_1(x), f_2(x)) = \max(f_1(x), C)$$

*is convex.*

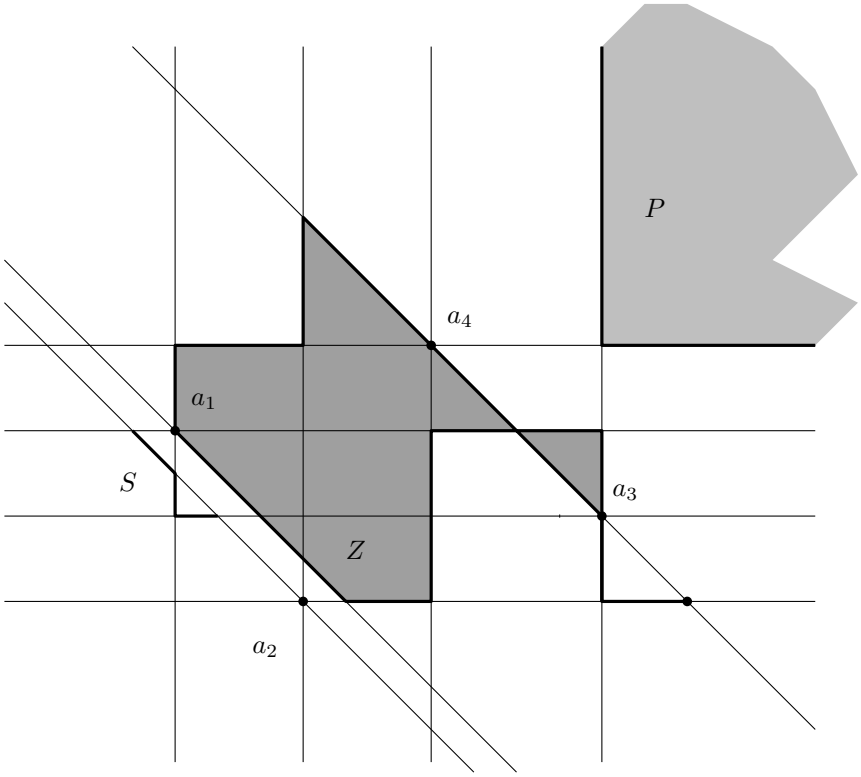



Figure 5.2: Some examples of planar $(D, A)$-polyhedra.

**Proof:** We start with the definition of convex function (see, e.g., Rudin [Rud76]): function $f$ is convex if

$$\forall x_1, x_2 : \forall 0 \leq \alpha \leq 1 : f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2).$$

We know that for every $y : f_1(y) \leq h(y) \geq f_2(y)$. Let $x = \alpha x_1 + (1-\alpha)x_2$. If $h(x) = f_i(x)$ then

$$h(x) = f_i(x) \leq \alpha f_i(x_1) + (1-\alpha)f_i(x_2) \leq \alpha h(x_1) + (1-\alpha)h(x_2).$$

It is clear that the function $h$ satisfies the convexity condition for every choice of $x_1, x_2$, and $\alpha$. □

**Corollary 5.2** *Let $f_1, f_2$ be any $D$-convex functions. Then the following function is also $D$-convex:*

$$h(x) = \max(f_1(x), f_2(x)).$$

*In particular, for $f_2(x) = C$ being a constant function, the function*

$$\max(f_1(x), f_2(x)) = \max(f_1(x), C)$$

*is $D$-convex.*

**Theorem 5.3** *Let $A_1, A_2, \ldots$ be a sequence of sets. Let $A$ be the limit of the sequence with the following property: every neighbourhood of the set $A$ contains infinitely many sets of the sequence, formally:*

$$A = \lim_{i\to\infty} A_i \text{ if } \forall\varepsilon > 0 : \exists N : \forall n \geq N : A_n \subset \Omega_\varepsilon(A).$$

*Let* $\mathrm{Clust}_i(A_i)$ *be the set of the cluster points of the sequence $A_1, A_2, \ldots$:*

$$\forall x \in \mathrm{Clust}_i(A_i) : \forall\varepsilon > 0 : \exists N : \forall n \geq N : A_i \cap \Omega_\varepsilon(x) \neq \emptyset.$$

*Then*

$$\mathrm{Clust}_i(\mathrm{co}^D(A_i)) \subseteq \mathrm{co}^D(A).$$

**Proof:** Let $x \notin \mathrm{co}^D(A)$. Then there is a nonnegative $D$-convex function $f$ for which $f^{-1}(0) = \mathrm{co}^D(A)$ and $f(x) = a > 0$. Let us define another nonnegative function $f' = \max(0, f - \frac{a}{2})$. It is $D$-convex due to Corollary 5.2. This function vanishes on a neighbourhood of $A$ and, on the other hand, it is positive on a neighbourhood of $x$:

$$\exists\varepsilon > 0 : \forall y \in \Omega_\varepsilon(A) = \Omega_A : f'(y) = 0 \text{ and } \forall y \in \Omega_\varepsilon(x) = \Omega_x : f'(y) > 0.$$

From the definition of functional $D$-convex hull, it follows that

$$\forall i : A_i \subset \Omega_A \Rightarrow \mathrm{co}^D(A_i) \cap \Omega_x = \emptyset.$$

Because there are infinitely many sets $A_i$ in $\Omega_A$, the point $x$ cannot be in $\mathrm{Clust}_i(\mathrm{co}^D(A_i))$. □

The following general observation has nothing to do with directional convexity, but we will need it later.

**Observation 5.4** *Let $\partial X$ denote the boundary of a subset $X$ of a topological space $Y$. For any two sets $B, C \subseteq Y$ holds:*

$$\partial(B \cap C) \subseteq \partial B \cup \partial C.$$

**Proof:** Let $\overline{X}$ denote the closure of a set $X$. From the closure axiom $X \subseteq \overline{X}$ follows

$$B \cup C \subseteq \overline{B} \cup \overline{C}, B \cap C \subseteq \overline{B} \cap \overline{C}.$$

Using another closure axiom $X \subseteq Y \Rightarrow \overline{X} \subseteq \overline{Y}$, and the fact that both the union and the intersection of two closed sets are closed, we have

$$\overline{B \cup C} \subseteq \overline{\overline{B} \cup \overline{C}} = \overline{B} \cup \overline{C}, \overline{B \cap C} \subseteq \overline{\overline{B} \cup \overline{C}} = \overline{B} \cap \overline{C}.$$

Following the definition

$$\partial X = \overline{X} \cap \overline{\neg X}$$

and De Morgan duality, we have

$$\partial(B \cap C) = \left(\overline{B \cap C}\right) \cap \left(\overline{\neg\left(B \cap C\right)}\right) = \left(\overline{B \cap C}\right) \cap \left(\overline{\neg B \cup \neg C}\right).$$

Applying the facts just proved, we see that

$$\left(\overline{B \cap C}\right) \cap \left(\overline{\neg B \cup \neg C}\right) \subseteq \left(\overline{B} \cap \overline{C}\right) \cap \left(\overline{\neg B} \cup \overline{\neg C}\right) =$$

$$= \left(\overline{B} \cap \overline{C} \cap \overline{\neg B}\right) \cup \left(\overline{B} \cap \overline{C} \cap \overline{\neg C}\right).$$

Again, using the definition of $\partial X$, we get

$$\begin{array}{ccccccc} \left(\overline{B} \cap \overline{C} \cap \overline{\neg B}\right) \cup \left(\overline{B} \cap \overline{C} \cap \overline{\neg C}\right) & \subseteq & \left(\partial B \cap \overline{C}\right) & \cup & \left(\overline{B} \cap \partial C\right) & \subseteq \\ & \subseteq & \partial B & \cup & \partial C. & \end{array}$$

□

# Chapter 6

# Some Basic Facts for $d = 2$

Throughout this chapter, we will explore the planar case ($d = 2$). We are given a finite set $A$ of points in the plane and a finite set $D$ being a simple set of directions spanning the plane.

## 6.1 The Basics

Here we state and prove a few simple but important basic facts about planar $(D, A)$-polyhedra.

**Observation 6.1** *Let $H$ be the arrangement of all $(D, A)$-lines. Let $Z$ be a planar polyhedron such that its boundary lies on the lines from $H$ and all of its vertices are vertices of $H$. Then $Z$ is planar $(D, A)$-polyhedron.*

**Proof:** Since both endpoints of every edge of $Z$ are vertices of $H$ and every edge lies completely on a line from $H$, we see that every edge of $Z$ is the union of one or more edges of $H$. It follows that every two-dimensional cell $C$ of $H$ is either a subset of $Z$, or it is disjoint with $Z$:

$$C \subseteq Z \Leftrightarrow (C \cap Z \neq \emptyset).$$

□

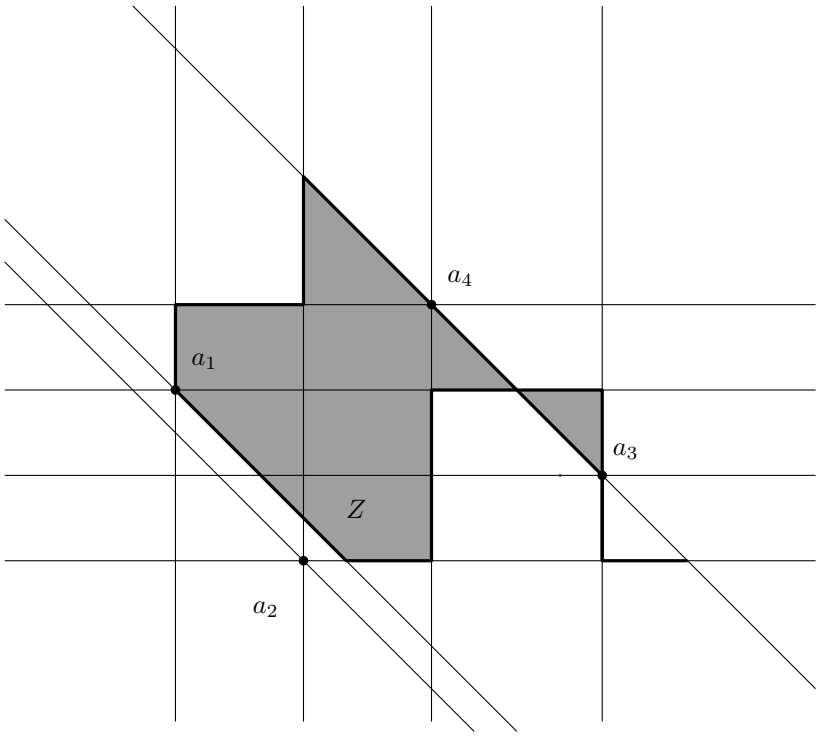



Figure 6.1: Illustration to Observation 6.1.

Figure 6.1 illustrates the situation. The sets are

$$D = \{(0, 1), (1, 0), (1, -1)\}$$

and

$$A = \{a_1, a_2, a_3, a_4\}.$$

The observation says that planar $(D, A)$-arrangement induces planar $(D, A)$-polyhedrons.

**Corollary 6.2** *The intersection of any two planar $(D, A)$-polyhedra is a planar $(D, A)$-polyhedron.*

**Proof:** Let $H$ be the arrangement of all $(D, A)$-lines. Let $X, Y$ be planar $(D, A)$-polyhedra and $Z = X \cap Y$. Thus $Z$ is a planar polyhedron. Observation 5.4 shows that all the boundaries of $Z$ lie on the lines from $H$. Due to the construction, every vertex of $Z$ is an intersection of two $(D, A)$-lines, and hence it is a vertex of $H$. The rest follows from Observation 6.1. □

**Observation 6.3** *Let $B$ be a $(D, A)$-polygon. Then every $D$-extremal point of $B$ is a vertex of $B$. We will call them* $D$-extremal vertices *of $B$.*

**Proof:** Let $x$ be a non-vertex point of $B$. If it lies in the interior of $B$, it is clearly not $D$-extremal. If it lies on an edge, the edge is a part of a $D$-line (by the definition of $(D, A)$-polyhedra). Because $x$ is not a vertex of $B$, it is not an endpoint of the edge and hence it is not $D$-extremal. □

## 6.2 The Tripod

In this section, we introduce a phenomenon which we encountered when studying functionally $D$-convex hulls. We call it the tripod. It is a $D$-convex set which has non-convex $D$-extremal vertex.

Let $|D| = 3$, let $B \supset A$ be a $D$-convex $(D, A)$-polygon. If there are some three edge atoms $e_1, e_2, e_3$ of $B$ adjacent to a non-convex $D$-extremal vertex $x \in B$ (this implies that $x$ lies inside the convex hull of the endpoints of the edge atoms), we call the union of these edge atoms an *atomic tripod* and $x$ is the *center* of this atomic tripod.

Let us explore atomic tripods more thoroughly.

**Lemma 6.4** *Let $D = \{d_1, d_2, d_3\}$, let $B \supset A$ be a $D$-convex $(D, A)$-polygon. Let $x$ be the center of an atomic tripod formed by the edge atoms $e_1, e_2, e_3$ of $B$, where each segment $e_i$ is parallel to the direction $d_i$. Let $y \neq x$ be the other endpoint of $e_1$. Then the following two observations hold:*

1. *In $B$, there is no segment $yz \nparallel e_1$.*
2. *$y \in A$.*

**Proof:** We prove it by contradiction.

1. This part of the proof is illustrated on Figure 6.2. Let there be a segment $yz \subset B$ such that $yz \nparallel e_1$. Then any $D$-line parallel to $e_1$ crossing the interior of $yz$ and the interior of $e_2$ or $e_3$ witnesses the non-$D$-convexity of $B$ — a contradiction.

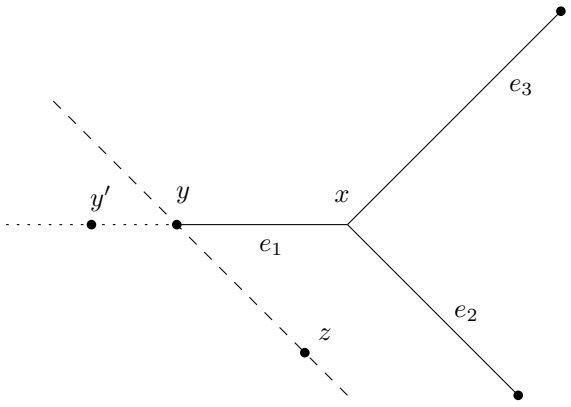



Figure 6.2: An atomic tripod with center $x$.

2. Let $y \notin A$. Since $y$ is a vertex of the underlying $(D, A)$-arrangement, it must be an intersection of some two non-parallel $(D, A)$-lines. It follows that there is a point $a \in A \subset B$ and a $D$-line $ay \nparallel xy$ witnessing the non-$D$-convexity of $B$: The segment $ya$ cannot be a part of $B$ according to the first part of this lemma. Then, the segment $ya$ is parallel to a $D$-line, it has both endpoints in $B$, but it is not a subset of the $D$-convex set $B$ — a contradiction.

□

In $B$, there can be an edge atom $yy' \parallel xy, y' \neq x$ (cf. Figure 6.2). Using the same argument as above, we see the following:

1. In $B$, there is no segment $y'z' \nparallel e_1$.
2. $y' \in A$.

It is time to define the *tripod* with center $x$ for $|D| = 3$, say $D = \{d_1, d_2, d_3\}$. Let $H$ be the arrangement of all $(D, A)$-lines. Let

$$Y = \bigcup_{j=1}^{3} \{y_{j,0}, y_{j,1}, \ldots, y_{j,k_j}\}$$

be a set of vertices of $H$ such that $k_j > 0$ for all $j$ and $y_{1,0} = y_{2,0} = y_{3,0} = x$ lies in the interior of the triangle $y_{1,1}y_{2,1}y_{3,1}$. Let $C$ be the set of segments

$$C = \bigcup_{j=1}^{3} \{[y_{j,0}, y_{j,1}], [y_{j,1}, y_{j,2}], \ldots, [y_{j,k_j-1}.y_{j,k_j}]\}.$$

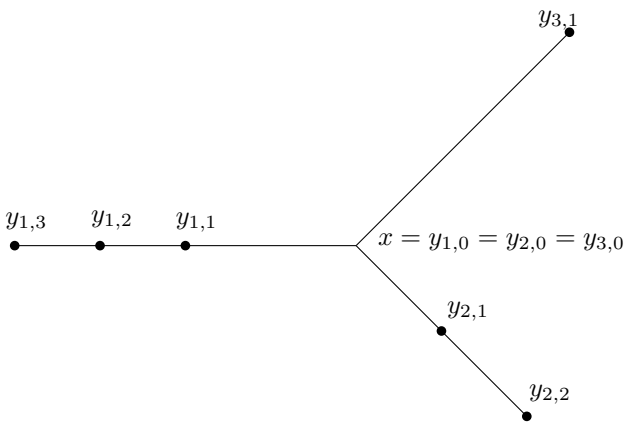



Figure 6.3: A tripod with center $x$.

If all the segments $[y_{j,i-1}, y_{j,i}]$ are parallel to $d_j$, then $C$ is clearly $D$-convex. Moreover, if $C \cup A$ is $D$-convex, then all of the endpoints of the segments are in $A$ (with the exception of the point $x$, which need not to be in $A$): $Y \setminus \{x\} \subseteq A$, and $C$ is called a tripod with center $x$ (cf. Figure 6.3). In particular, any atomic tripod is a tripod as well. For $|D| \neq 3$, there are no tripods defined.

From the previous analysis, it is clear that if there is an atomic tripod in a $D$-convex $(D, A)$-polygon $B$ then the connected component of $B$ containing this atomic tripod is a tripod formed by three edges of $B$.

**Lemma 6.5** *Let $B$ be a $D$-convex $(D, A)$-polygon. Then for every $D$-extremal point $x$ of $B$ holds:*

1. *For $|D| \neq 3$, $x$ is a convex vertex of the polygon.*
2. *For $|D| = 3$, $x$ is either a convex vertex of the polygon, or the center of a tripod.*

**Proof:** We will prove it by contradiction. Let $x$ be a non-convex $D$-extremal vertex of $B$. Let $e_1, \ldots, e_k$ be all the edges around $x$ in clockwise order. Because $x$ is $D$-extremal vertex of a $D$-polygon, no two of these edges are parallel and $k \geq 3$ and $|D| \geq 3$.

Let $e'_1, \ldots, e'_k$ be the rays opposite to these edges, ie. $e'_j || e_j$ and $e'_j \cap e_j = \{x\}$, for all $j = 1, 2, \ldots, k$. Because $x$ is $D$-extremal and $B$ is functionally $D$-convex, $x$ is the only point of the rays that lies in $B$. Let us take the two edges $e_i, e_{i+1}$

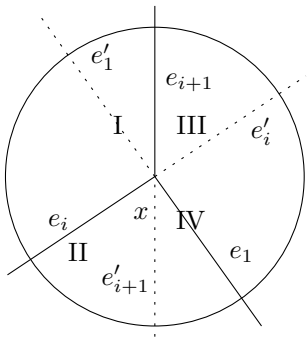



Figure 6.4: Inspecting the center of a tripod.

that have the ray $e'_1$ between them. Because the vertex $x$ is not convex, we know that $1 < i < k$.

Let $\bar{\Omega}(x)$ be a closed neighbourhood of $x$ so small that $x$ is the only vertex of $H$ lying in the neighbourhood. The situation around $x$ is as on Figure 6.4. Let the Roman numbers I to IV denote the four closed regions in the neighbourhood, given by the two lines supporting $e_i, e_{i+1}$:

- I contains $e'_1 \cap \bar{\Omega}(x)$;
- II is the convex region between $e'_{i+1}$ and $e_i$;
- III is the convex region between $e_{i+1}$ and $e'_i$;
- IV contains $e_1 \cap \bar{\Omega}(x)$.

We have selected the index $i$ in such a way that the interior of the region I is empty (meaning that it does not contain any point of the set $B$). Because $x$ is not convex and it is $D$-extremal, the region I spans less than half of the neighbourhood (the angle between $e_i$ and $e_{i+1}$ is less than $\pi$).

Let us inspect the possible shape of the $D$-convex $(D, A)$-polygon $B$ in the neighbourhood $\bar{\Omega}(x)$. Let us suppose that there is an edge $e_{i-1}$, that lies strictly between $e'_{i+1}$ and $e_i$ (thus $e_{i-1}$ goes through the interior of II). Then the situation is as in Figure 6.5. Because $e_{i-1}$ is parallel to a $D$-line, we see that in this case $B$ is not $D$-convex (remember that I is empty) — a contradiction. So the region II is empty except for $e_i$. The situation for the region III is analogous; it is empty except for $e_{i+1}$.

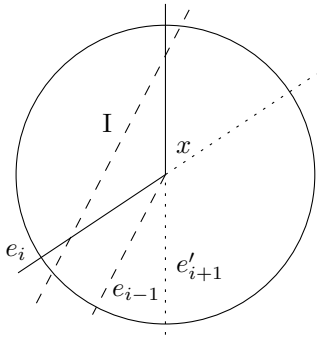



Figure 6.5: An edge through the interior of II.

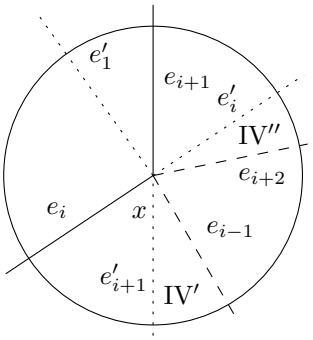



Figure 6.6: The interiors of I, II, and III are empty.

Now we know that $B \cap (\mathrm{I} \cup \mathrm{II} \cup \mathrm{III}) \subset e_i \cup e_{i+1}$. Let us put $e_{k+1} = e_1$ and let us have a look at the edges $e_{i-1}$ and $e_{i+2}$. The situation is depicted in Figure 6.6 (the region IV′ between $e_{i-1}$ and $e'_{i+1}$ is empty and so is the region IV″ between $e'_i$ and $e_{i+2}$). If these edges do not coincide, we see that $B$ is not $D$-convex (for example, due to a $D$-line parallel to $e_{i+2}$ crossing both $e_{i-1}$ and $e_i$ in $\bar{\Omega}(x)$ and thus crossing also the interior of the empty region IV′).

But if the edges coincide (i.e., the case when $i = 2$ and $k = 3$), for $|D| > 3$ the polygon $B$ would not be $D$-convex due to any direction from $D$ other than those of $e_1, e_2$ and $e_3$, and thus the interior of IV must be empty as well — a contradiction (we stated that $e_1$ goes through the interior of IV). For $|D| = 3$, the edges $e_1, e_2, e_3$ form a tripod with the center in $x$. □

**Corollary 6.6** *Let $B \supseteq A$ be a $D$-convex $(D, A)$-polygon. Any $D$-extremal vertex of $B$ which is not the center of a tripod is convex.*
*In particular, if all the convex vertices of $B$ are in $A$ then every component of $B$ which is not a tripod has all its $D$-extremal points in $A$.*

# Chapter 7

# The Cutter Lemma

In this chapter, we establish *Cutter Lemma*, one of the key tools proving the correctness of the algorithm for computing functionally $D$-convex hull of a finite set, which is given in Section 8. Although the algorithm is given for the planar case, we expect this lemma to be useful for higher dimensions, too, and we formulate and prove it for arbitrary dimension $d$.

**Lemma 7.1 (Cutter Lemma)** *Let $D$ be an arbitrary set of directions in $\mathbf{R}^d$, let $B \subset \mathbf{R}^d$ be a compact functionally $D$-convex set, and let $T \subseteq \mathbf{R}^d$ (the "cutting tool") be closed and such that $\mathbf{R}^d \setminus \operatorname{int} T$ is the zero set of a nonnegative $D$-convex function. Let $R$ be a subset of $B \cap T$ such that the distance of $R$ from its complement in $B \cap T$, i.e. from $(B \cap T) \setminus R$, is strictly positive (see Figure 7.1). Then the set*

$$\check{B} = B \setminus (R \cap \operatorname{int} T)$$

*is functionally $D$-convex.*

Note that the requirement for $\mathbf{R}^d \setminus \operatorname{int} T$ is stated as being the zero set of a nonnegative $D$-convex function. This is caused by the fact that this set is unbounded, and since we have defined only *compact* functionally $D$-convex sets, we cannot speak about functional $D$-conexity of unbounded sets. Indeed, the usual definition of $D$-convexity, used for unbounded sets, is in general not equivalent to being a zero set of a nonnegative $D$-convex function.

**Proof:** Let $f\colon \mathbf{R}^d \to [0, \infty)$ be a nonnegative $D$-convex function witnessing the functional $D$-convexity of $B$ (i.e., $B$ is the zero set of $f$) and let $g\colon \mathbf{R}^d \to [0, \infty)$

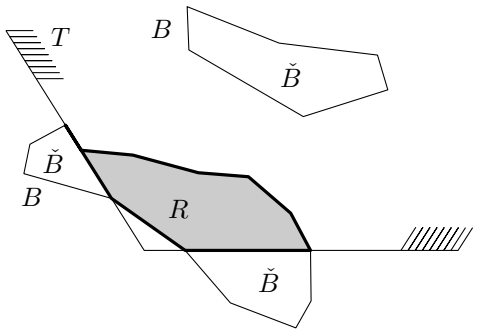



Figure 7.1: The Cutter Lemma.

be a nonnegative $D$-convex function whose zero set is $\mathbf{R}^d \setminus \operatorname{int} T$. We may also assume that $f$ and $g$ are continuous (see, e.g., Matoušek and Plecháč [MP98]).

For a set $R \subset \mathbf{R}^d$ and $\delta > 0$, let $R^\delta$ denote the closed $\delta$-neighborhood of $R$.

Let $\delta > 0$ be half of the distance between $R$ and $(B \cap T) \setminus R$, and let $\eta > 0$ be a sufficiently small real number, to be determined later. We define a function $h\colon \mathbf{R}^d \to [0, \infty)$ by

$$h(x) = \begin{cases} \max(f(x), \eta \cdot g(x)) & \text{for } x \in R^\delta \cap T \\ f(x) & \text{otherwise.} \end{cases}$$

*Claim I:* $\check{B}$ is the zero set of $h$.

*Proof of Claim I.* The function $h$ is nonzero on the complement of $B$ since $f > 0$ there, and it is nonzero on $R \cap \operatorname{int} T$ since $g > 0$ there. Thus $h(x) > 0$ for all $x \notin \check{B}$. Now let $x \in \check{B}$. Then $x \in B \setminus \operatorname{int} T$ or $x \in (B \cap T) \setminus R$. In the former case both $f(x) = 0$ and $g(x) = 0$, and so $h(x) = 0$ as well. For the latter case we get, by the condition on $R$ in the lemma, that $x \notin R^\delta \cap T$, and hence $h(x) = f(x)$ and $f(x) = 0$ since $x \in B$. This proves Claim I.

*Claim II:* Let $S$ be the closure of the set $(R^\delta \setminus R^{\delta/2}) \cap T$. Then the minimum of $f$ on $S$ is strictly positive.

*Proof of Claim II* (cf. Figure 7.2). Since $S$ is compact, it suffices to show $f(x) > 0$ for all $x \in S$, and this is the same as $S \cap B = \emptyset$. Any point of $S$ has distance at least $\delta/2$ to $R$, and hence an $x \in S \cap B$ would have to lie in $(B \cap T) \setminus R$, but this set is disjoint from $R^\delta$. This proves Claim II.

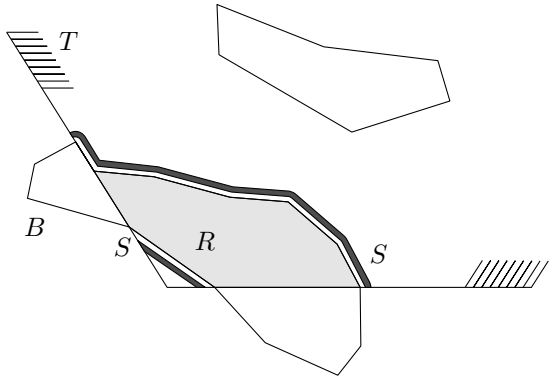



Figure 7.2: An illustration to Claim II.

Let us set

$$\mu = \min_{x \in S} f(x)$$

and

$$M = \max_{x \in S} g(x).$$

We have $\mu > 0$ by Claim II and $M < \infty$. We put

$$\eta = \mu / M.$$

Then $f \geq \eta \cdot g$ on $S$ and hence $h(x) = f(x)$ for $x \in S$.

Let $\ell$ be a $D$-line. For a more convenient notation let us identify the line $\ell$ isometrically with the real axis, so that, for example, $a + \varepsilon$ is a well-defined point of $\ell$ for $a \in \ell$ and $\varepsilon \in \mathbf{R}$. Let us write $h_\ell$ for $h$ restricted to $\ell$, and similarly for $f_\ell$ and $g_\ell$.

*Claim III:* Suppose that $\eta > 0$ is as above and $\ell$ is an arbitrary $D$-line. Then at least one of the following holds for every point $a \in \ell$:

(i) There exists $\varepsilon = \varepsilon(a) > 0$ such that $h_\ell(x) = f_\ell(x)$ for all $x \in (a - \varepsilon, a + \varepsilon)$.

(ii) There exists $\varepsilon = \varepsilon(a) > 0$ such that $h_\ell(x) = \max(f_\ell(x), \eta \cdot g_\ell(x))$ for all $x \in (a - \varepsilon, a + \varepsilon)$.

(iii) $f(a) = g(a) = h(a) = 0$.

*Proof of Claim III.* We distinguish several cases, depending on the position of the point $a$.

- For $a \notin R^\delta \cap T$ there is a neighborhood of $a$ avoiding $R^\delta \cap T$, and so (i) applies.
- For $a$ lying on the boundary of $T$ we have $g_\ell(a) = 0$. If (i) doesn't apply, then arbitrarily small neighborhoods of $a$ on $\ell$ contain points $x$ with $h_\ell(x) \neq f_\ell(x)$. It follows from the definiton of $h$ that $f_\ell(x) \neq h_\ell(x) = \max(f_\ell(x), \eta \cdot g_\ell(x))$ and hence $f_\ell(x) < \eta \cdot g_\ell(x)$. Then we get $f_\ell(a) = 0$ by the continuity of $f$ and $g$, and (iii) applies.
- For $a \in \operatorname{int}(T \cap R^\delta)$ there is a neighborhood of $a$ contained in $\operatorname{int}(T \cap R^\delta)$, and hence (ii) applies.
- The remaining case is $a$ lying in $\operatorname{int} T$ and on the boundary of $R^\delta$. If we choose $\varepsilon = \delta/2$ then each $x \in (a - \varepsilon, a + \varepsilon)$ lies either in $S$ or outside $R^\delta$, and in both cases we have $h_\ell(x) = f_\ell(x)$. Hence (i) applies.

Claim III is proved.

Claim III shows that the function $h_\ell$ has the following property: For every point $a \in \ell$ there exists $\varepsilon = \varepsilon(a) > 0$ and an affine function $\sigma_a \colon \ell \to \mathbf{R}$ that is supporting for $h_\ell$ on $(a - \varepsilon, a + \varepsilon)$; that is, $\sigma_a(a) = h_\ell(a)$ and $\sigma_a(x) \leq h_\ell(x)$ for all $x \in (a - \varepsilon, a + \varepsilon)$. Indeed, in cases (i) and (ii) $h_\ell$ equals to a convex function on $(a - \varepsilon, a + \varepsilon)$ and hence it has a supporting tangent at $a$, and in case (iii) $\sigma_a(x) = 0$ is supporting since $h_\ell$ is nonnegative.

To conclude the proof, we will need the following lemma:

**Lemma 7.2** *Let $f : [a, b] \to \mathbf{R}$ be a continuous function such that for each $x \in (a, b)$, there exists a linear function $T_x : \mathbf{R} \to \mathbf{R}$ and a neighborhood $U_x$ of zero in $\mathbf{R}$ such that $f(x + t) \geq f(x) + T_x(t)$ for all $t \in U_x$. Then the function $f$ is convex.*

This is almost a verbatim quotation of Lemma 3.1 in Matoušková and Reich [MR03], where it is also proved.

This implies that $h_\ell$ is convex (consider Lemma 7.2 and $\sigma_a(x)$ as $h_\ell(a) + T_a(a - x)$) and hence $h$ is $D$-convex. Lemma 7.1 is proved. □

# Chapter 8

# The Cutting Algorithm

In this chapter, we present the algorithm, which, for a given set of vectors $D$, yields the functional $D$-convex hull of the given point set $A$. Throughout this chapter, we suppose that $d = 2$.

## 8.1 The Algorithm

We have a finite set $A \in \mathbf{R}^2$ and a finite simple set of directions $D \in \mathbf{R}^2$. Using the following algorithm, we want to find the functional $D$-convex hull of $A$. In each iteration $s = 1, 2, \ldots$ of the algorithm, we have a current set $B = B_{s-1}$ which is the approximation of the hull.

1. Find a convex vertex $v$ of $B = B_{s-1}$ not in $A$. If there is no such vertex, the algorithm ends.

2. **Cutting Tool** Find a cutting tool $T_0$ (a planar $D$-polyhedron with $\mathbf{R}^2 \setminus \operatorname{int} T_0$ being the zero set of a nonnegative $D$-convex function) such that

   $$R_0 = \{v\},$$

   where $R_0$ is the connected component of $B \cap T_0$ containing $v$.

3. **Translation** Translate $T_0$ to an "extremal position" $T$ so that $T$ is a planar $(D, A)$-polyhedron and these two conditions hold:

   (a) $v \in \operatorname{int} T$;

(b) $A \cap (R \cap \operatorname{int} T) = \emptyset$ for $R$ being the connected component of $B \cap T$ that contains $v$.

Later on in this chapter, we are going to explain in detail, how such a translation can be achieved. We call the set $T$ a *cutter for $v$ in a fixed position.*

4. **Cut** Put
$$B_s = B \setminus (R \cap \operatorname{int} T)$$
and continue with the next iteration.

In the following paragraphs, we show the following lemma:

**Lemma 8.1** *In every iteration $s$ of the algorithm, $B = B_{s-1}$ is*

1. *a $(D, A)$-polygon*
2. *a superset of $A$*
3. *functionally $D$-convex.*

To make this true from the very beginning ($s = 1$), we construct $B_0$ by the following procedure:

For every direction $d_i$ from $D$, let us consider all the halfplanes given by all the $(D, A)$-lines parallel to $d_i$. Among them, there are two different halfplanes (say $h_{i1}$ and $h_{i2}$) containing all points from $A$. Let us put (as illustrated on Figure 8.1)
$$B_0 = h_{11} \cap h_{12} \cap h_{21} \cap h_{22}.$$
This set is an intersection of halfplanes and hence it is convex and hence functionally $D$-convex. All the halfplanes are supersets of $A$, hence the intersection is a superset of $A$ as well. All the halfplanes are given by $(D, A)$-lines and thus $B_0$ is a $(D, A)$-polygon.

We just have proved that $B_0$ is a functionally $D$-convex $(D, A)$-polygon that is a superset of $A$.

**Proof of part 1 of Lemma 8.1:** One can write
$$B_s = B \setminus (R \cap \operatorname{int} T) = (B \cap (\mathbf{R}^2 \setminus \operatorname{int} T)) \cup ((B \cap T) \setminus R).$$

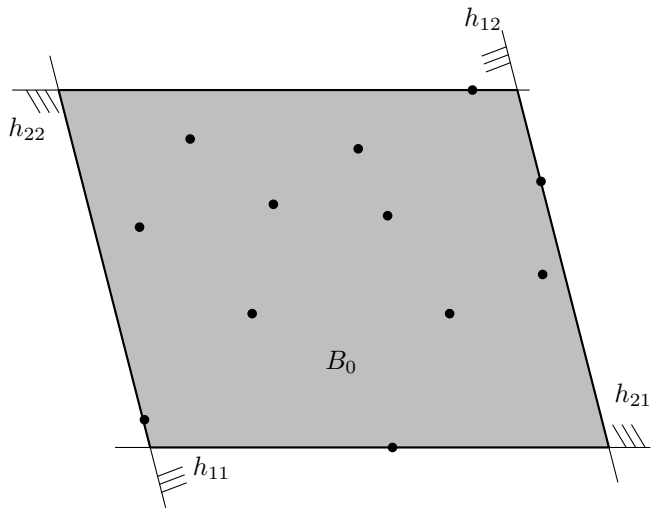



Figure 8.1: The initial $(D, A)$-polygon $B_0 = h_{11} \cap h_{12} \cap h_{21} \cap h_{22}$.

The first set

$$V = B \cap (\mathbf{R}^2 \setminus \operatorname{int} T)$$

containing the untouched parts of $B$ not in the interior of the cutter $T$ is a $(D, A)$-polygon due to Corollary 6.2: $B$ is a $(D, A)$-polygon by induction, $\mathbf{R}^2 \setminus \operatorname{int} T$ is a $(D, A)$-polygon by definition, hence their intersection $V$ is also a $(D, A)$-polygon.

Since $R$ is a connected component of $T \cap B$, the set

$$U = (B \cap T) \setminus R$$

contains all of the (finitely many) components of $B$ in $T$ except for $R$. Note that, due to $D$-convexity of $B$, the $(D, A)$-lines bounding $T$ cannot intersect any component of $B$ outside $R$. Hence, $U$ is a $(D, A)$-polygon disjoint from $V$. $U$ and $V$ are $(D, A)$-polygons consisting of finitely many disjoint components, hence their union $B_s$ is a $(D, A)$-polygon as well. □

To prove part 2 of Lemma 8.1 for every $s$, it is enough to show that the algorithm does not cut off any point of $A$, that is:

**Lemma 8.2** *In every iteration, for $R, T$ defined as above: $A \cap (R \cap \operatorname{int} T) = \emptyset$.*

To prove part 3 of Lemma 8.1 for every $s$, it is enough to show that we can apply the Cutter Lemma while cutting:

**Lemma 8.3** *For $T, B, R$ defined as above, in every iteration of the algorithm we have:*

1. *$T$ is closed*
2. *$\mathbf{R}^2 \setminus \operatorname{int} T$ is the zero set of a nonnegative $D$-convex function*
3. *$B$ is a compact functionally $D$-convex set*
4. *$R \subseteq B \cap T$*
5. *the distance of $R$ from $(B \cap T) \setminus R$ is strictly positive.*

The parts 1, 4 of this lemma are satisfied from the definition immediatly. The set $B = B_{s-1}$ is functionally $D$-convex by induction, and, being a closed subset of the compact set $B_0$, it is compact as well. It is also clear that the distance of $R$ from $(B \cap T) \setminus R$ is strictly positive, because there are only finitely many components.

Let us explore the particular steps of the algorithm and, from the results, we prove the rest, namely Lemma 8.2 (and hence part 2 of Lemma 8.1) and part 2 of Lemma 8.3 (and hence complete the proof of Lemma 8.3 and hence the proof of part 3 of Lemma 8.1).

## 8.2 The Cutting Tool

Before moving on to the definition of the cutting tool, we establish two more notions.

Suppose that we have a *separating line* $h$ in $v$, i.e.,

$$v \in h \text{ and } \exists \varepsilon > 0 : (\Omega_\varepsilon(v) \cap B) \cap h^- = \{v\},$$

where $h^-$ is one of the closed halfplanes given by $h$. Moreover, let there be no direction in $D$ parallel to $h$. (This can be easily achieved by a small rotation of $h$ around $v$.) Such a line always exists, because $v$ is a convex vertex of $B$. We may, for example, take the line supporting one of the edges incident to $v$ and rotate it a little around $v$, so that it becomes a separating line and is not parallel to any direction in $D$.

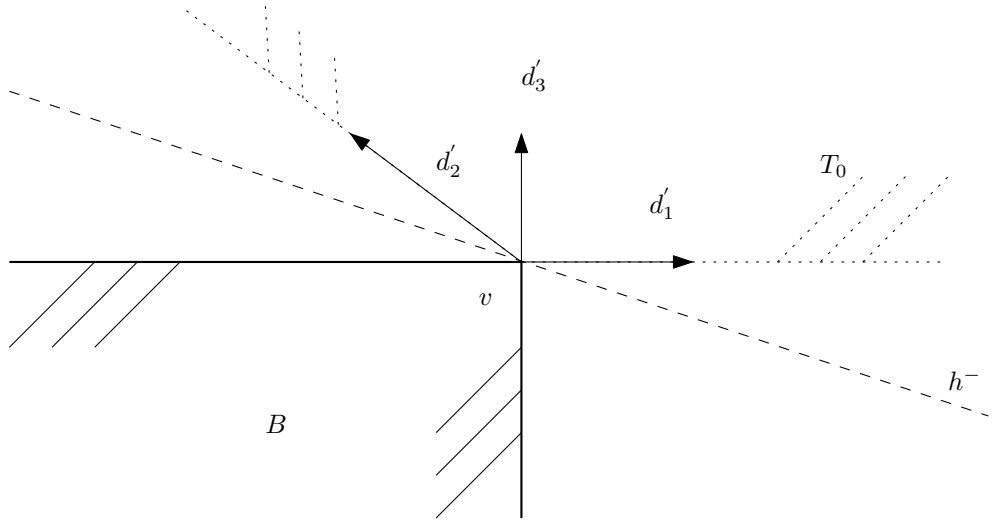



Figure 8.2: An example of construction of the cutting tool ($D = \{-d_1', d_2', d_3'\}$).

Based on this, let us define good separating halfplane. A closed halfplane $H$ is a *good separating halfplane* of a convex vertex $v$ of a $(D, A)$-polygon $B$ if its boundary line goes through $v$ and it is not a $D$-line, and for sufficiently small neighbourhood $\Omega(v)$ of $v$ we have $(\Omega(v) \cap B) \cap H = \{v\}$. The halfplane $h^-$ from the previous paragraph is a good separating halfplane of $v$.

Now we are ready to define the cutting tool: let us define the set

$$T_0 = \mathrm{co}((v + \beta c) \cap h^- : \beta \in \mathbf{R}, c \in D)$$

as the convex hull of all the $D$-rays through $v$ lying in the closed halfplane $h^-$. Clearly, this set is a convex cone with apex $v$. The cone is given by two $D$-rays

$$(v + \alpha d_i') : 1 \leq i \leq 2, \alpha \geq 0,$$

where $d_i' \in D \cup (-D)$, and hence $T_0$ is a planar $D$-polyhedron. Because $h$ was chosen in such a way that it is not a $D$-line, we have $d_1', d_2' \nparallel h$ and thus $T_0 \subset h^-$.

The construction is illustrated on Figure 8.2.

**Lemma 8.4** *Let us denote the rays bounding the cone $T_0$ by $F_1, F_2$. Let $n_i$ be the normal vector of the ray $F_i$ pointing to the interior of the cone. Then the function*

$$f(x) = \prod_{k=1}^{2} \max(0, \langle n_i, x - v \rangle)$$

*vanishes exactly on $\mathbf{R}^2 \setminus \mathrm{int}\, T_0$.*

**Proof:** For every $i$, $\langle n_i, x - v\rangle$ is strictly positive (and hence $\max(0, \langle n_i, x - v\rangle)$, too) exactly for all $x$ lying in the open halfplane given by the line supporting the $i$-th ray and lying on the same side as $v + n_i$. Hence, $f(x)$ is strictly positive exactly for every $x$ lying in the intersection of both these open halfplanes. This intersection equals to $\operatorname{int} T_0$. The rest follows from the fact that, being the product of nonnegative functions, $f$ is also nonnegative. □

**Lemma 8.5** *The function $f$ is $D$-convex.*

**Proof:** We show that $f$ is convex along any $D$-line $\ell$. If $\ell$ does not intersect $\operatorname{int} T_0$, then $f|_\ell$ is zero and hence convex. Let $\ell$ intersect $\operatorname{int} T_0$. Let

$$t_\ell = \ell \cap (T_0 \setminus \operatorname{int} T_0)$$

be the intersection of $\ell$ with the boundary of $T_0$. Then there is a vector $d'$ parallel to the $D$-line $\ell$ and hence to a direction from $D$ for which

$$\ell \cap T_0 = \{t_\ell + \alpha d' | \alpha \geq 0\}.$$

Along the line

$$\ell = \{t_\ell + \beta d' | \beta \in \mathbf{R}\},$$

we have

$$f|_\ell(x) = f|_\ell(t_\ell + \beta d') = \prod_{i=1}^{2} \max(0, g_i^\ell(\beta)),$$

where for both $i$:

$$g_i^\ell(\beta) = \langle n_i, t_\ell + \beta d' - v\rangle = \langle n_i, (t_\ell - v) + \beta d'\rangle = \langle n_i, t_\ell - v\rangle + \langle n_i, \beta d'\rangle.$$

Without loss of generality, let $\ell$ intersect the ray $F_1$:

$$t_\ell = \ell \cap F_1 \neq \emptyset.$$

The situation is depicted on Figure 8.3. We have $n_1 \perp (t_\ell - v)$ and hence

$$g_1^\ell(\beta) = \langle n_1, \beta d'\rangle.$$

Due to construction, the span of the angle between $n_1$ and $d'$ is in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$. Thus we have

$$\begin{array}{rcl} \beta < 0 & : & g_1^\ell(\beta) < 0; \\ \beta = 0 & : & g_1^\ell(\beta) = 0; \\ \beta > 0 & : & g_1^\ell(\beta) > 0. \end{array}$$

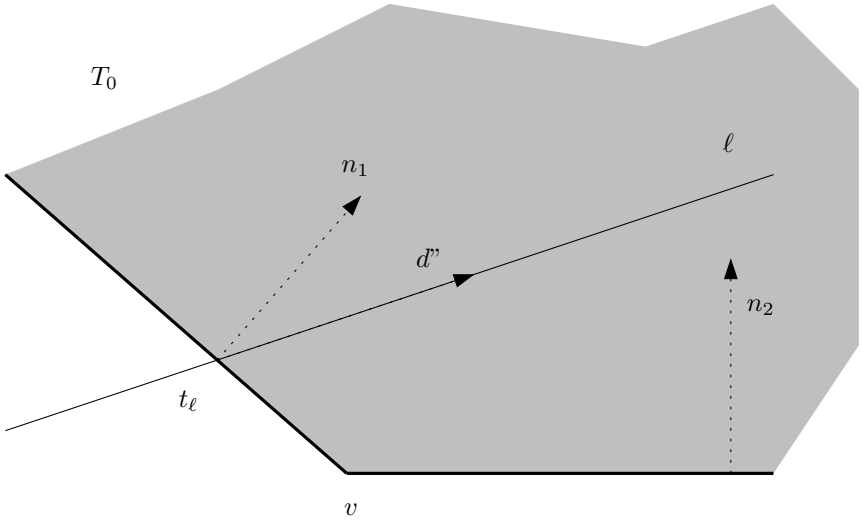



Figure 8.3: A $D$-line $\ell$ passing through the cone.

We see that $g_1^\ell(\beta)$ is an increasing linear function:

$$g_1^\ell(\beta) = p\beta, p > 0.$$

Let us examine the function $g_2^\ell$. So far we know:

$$g_2^\ell(\beta) = \langle n_2, t_\ell - v\rangle + \langle n_2, \beta d'\rangle.$$

Because $T_0$ does not span the whole halfplane $h^-$, the span of the angle between $n_2$ and $(t_\ell - v)$ is in the interval $(-\frac{\pi}{2}, \frac{\pi}{2})$. Thus we have

$$\langle n_2, t_\ell - v\rangle = C > 0.$$

On the other hand, the span of the angle between $n_2$ and $d'$ is in the interval $\langle -\frac{\pi}{2}, \frac{\pi}{2}\rangle$. (If it was not the case, then one of the $D$-rays

$$\{v + \alpha d' | \alpha \geq 0\}, \{v - \alpha d' | \alpha \geq 0\}$$

would belong to $h^-$ but would not belong to $T_0$ — a contradiction with the definition of $T_0$ as the convex hull of all the $D$-rays through $v$ lying in $h^-$. See Figure 8.4.) We see that $\langle n_2, \beta d'\rangle$ is either an increasing linear function, or 0 (for $n_2 \perp d'$).

We have finished our analysis of the function $g_2^\ell$ and we know that

$$g_2^\ell(\beta) = C + q\beta, C > 0, q \geq 0.$$

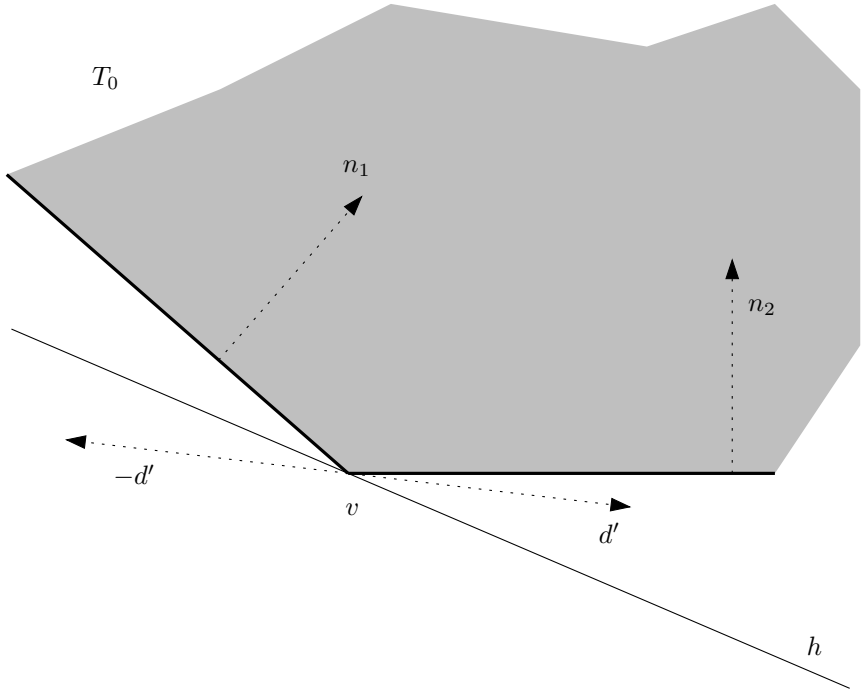



Figure 8.4: Span of the angle between $n_2$ and $d'$ is in the interval $\langle -\frac{\pi}{2}, \frac{\pi}{2} \rangle$.

Putting this together, we have

$$\begin{array}{rrcl} & f|_\ell(x) & = & f|_\ell(t_\ell + \beta d') = \prod_{k=1}^{2} \max(0, g_k^\ell(\beta)) \\ \beta \leq 0 & : & = & 0 \cdot \max(0, g_2^\ell(\beta)) = 0 \\ \beta > 0 & : & = & g_1^\ell(\beta) g_2^\ell(\beta) = p\beta(C + q\beta), \end{array}$$

with $p, C > 0$ and $q \geq 0$. Let

$$h^\ell(\beta) = g_1^\ell(\beta) g_2^\ell(\beta).$$

For $\beta \geq 0$, we see that it is either a linear function (for $q = 0$), or a quadratic convex function. In any case, the function $h^\ell$ is convex on $\mathbf{R}^+$.

Finally, we show that for every $a \in \ell$, there is an affine function that is supporting for $f|_\ell$. Let

$$a = t_\ell + \beta d'.$$

For $\beta \leq 0$, we see that $f|_\ell(a)$ is zero and the zero function is supporting, since $f|_\ell$ is nonnegative. For $\beta > 0$, we have

$$f|_\ell(a) = h^\ell(t_\ell + \beta d'),$$

and $h^\ell$ has a supporting tangent at $a$, because $h^\ell$ is convex.

Together with Lemma 7.2, this implies that $f|_\ell$ is convex. □

So $f$ witnesses that $\mathbf{R}^2 \setminus \operatorname{int} T_0$ is the zero set of a nonnegative $D$-convex function, and hence for every translation $T$ of $T_0$ the difference $\mathbf{R}^2 \setminus \operatorname{int} T$ is the zero set of a nonnegative $D$-convex function, too, which concludes the proof of part 2 of Lemma 8.3.

To show that we really have found the cutting tool desired in the description of the algorithm, one more small thing is to be pointed out:

**Lemma 8.6** *Let $R_0$ be the connected component of $B \cap T_0$ containing $v$. In this initial case, we have*

$$R_0 = \{v\}.$$

**Proof:** The proof follows immediatly from the definition of the separating line $h$. □

## 8.3 The Translation

Let $M$ be the connected component of $B$ containing $v$. (Note that $R \subseteq M$.) We will translate the cone $T_0$ twice. After the first translation, we denote the cone by $T_1$, and $T_2 = T$ after the second one. These translations will be done in the following way ($v_i$ denotes the apex of the cone $T_i$):

$$T_i = T_{i-1} - \alpha_i d_i' = v_i + T',$$

where $\alpha_i > 0$ is defined by

$$\alpha_i = \min_{\mathbf{R}^+}\{\alpha | \exists a \in ((A \cap M) \setminus T_{i-1}) \cap (T_{i-1} - \alpha d_i')\}$$

and $T' = T_0 - v$ is the cone $T_0$ translated to the origin.

In other words, we move the apex of the cone along the ray opposite to one boundary of the cone, until the moved cone hits a new point from $A$ in $M$[1].

[1]Note that here is a slight difference between this algorithm and the one given in the article [FM]. Here we are concerned in the points in the intersection $A \cap M$, whereas we considered *any* point of $A$ in the earlier version of the algorithm given in the article. This minor enhancement has a quite substantial impact on the analysis of complexity of the algorithm as we will see in Chapter 9.

Then we will do the same with this new cone $T_1$ and the other boundary. If both the rays bounding the cone $T = T_2$ contain a point of $A$, then $T$ is clearly a $(D, A)$-polyhedron.

If there is no such point, then $\alpha_i = \infty$. Thus $T_i$ degenerates to a halfplane bounded by a $D$-line, which is a planar $(D, A)$-polyhedron, still. If there were $\alpha_1 = \alpha_2 = \infty$ then $T_2$ would degenerate to the whole plane, but this situation cannot happen:

**Lemma 8.7** *In the translation just described, we have at least one coefficient bounded:*

$$\min(\alpha_1, \alpha_2) < \infty.$$

**Proof:** Because $v \notin A$ is a convex $D$-extremal vertex of the functionally $D$-convex $(D, A)$-polygon $B$, there must be an edge $[v, v_1]$ of the polygon $B$ ($v$ cannot be an isolated vertex of $B$, since $B$ is $D$-convex and $v \notin A$ lies on a $(D, A)$-line). Because $B$ is a $(D, A)$-polygon, this edge lies on a $(D, A)$-line, hence on this line $vv_1$, there must be a point from $A$ — say, $a_1$. Because $B$ is $D$-convex and $v \notin A \subseteq B$, this point $a_1$ must lie in the intersection of the ray $vv_1$ and the connected component $M$ of $B$ containing $v$. This way we have

$$A \supseteq \{a_1\} \subset M.$$

To translate the cone in such a way that it does not hit this point, one must avoid the component $M$ during translation as well. But this is not possible: Let $\alpha_1 = \infty$. We have $T_1$ being a halfplane not containing $a_1$:

$$T_1 \cap \{a_1\} = \emptyset.$$

It follows that $\{a_1\} \subset (A \setminus T_1)$, hence during the second translation, we must hit a point from $A \cap M$ (because there is at least $a_1$ in this intersection), and hence $\alpha_2 < \infty$. □

The steps of the translation are shown on Figures 8.5–8.7. The degenerated case (where $\alpha_2 = \infty$) is shown on Figures 8.8–8.10.

**Lemma 8.8** *During translation, the set $R_i$ (the connected component of $B \cap T_i$ that contains $v$) cannot grow into $T_0$. Formally, for every $i \in \{0, 1, 2\}$:*

$$R_i \cap T_0 = \{v\}.$$

*In particular, note that for $i = 2$, we have $R = R_i$ and*

$$R \cap T_0 = \{v\}.$$

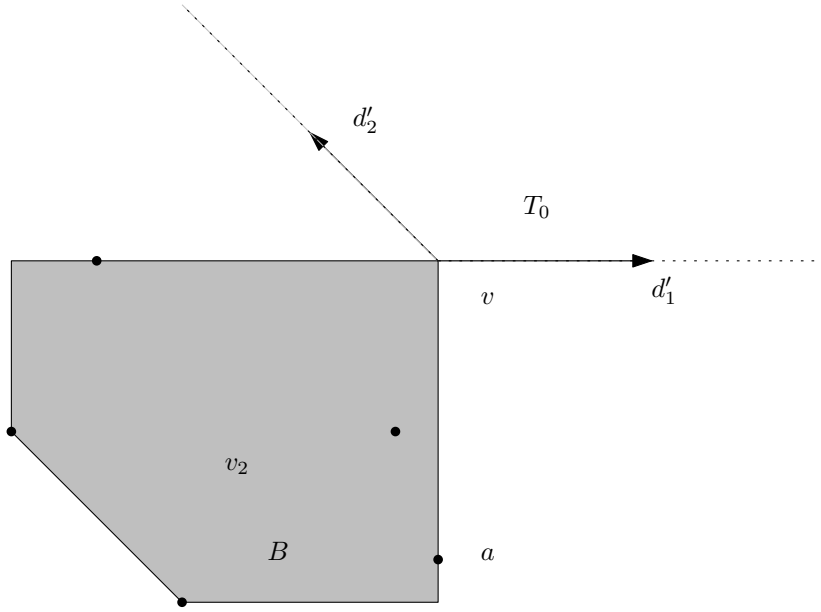



Figure 8.5: Translation of the cone: the initial state. The cone $T_0$ is given by $d_1', d_2'$, the set $A$ by the black points, the set of directions is $D = \{d_1', d_2', va\}$.

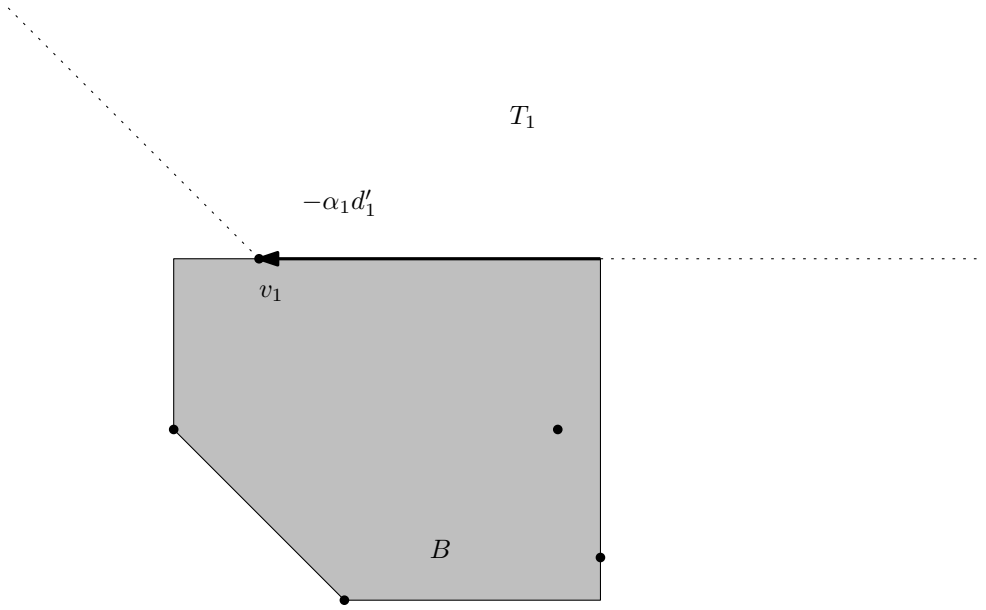



Figure 8.6: Translation of the cone: the first translation to $v_1$.

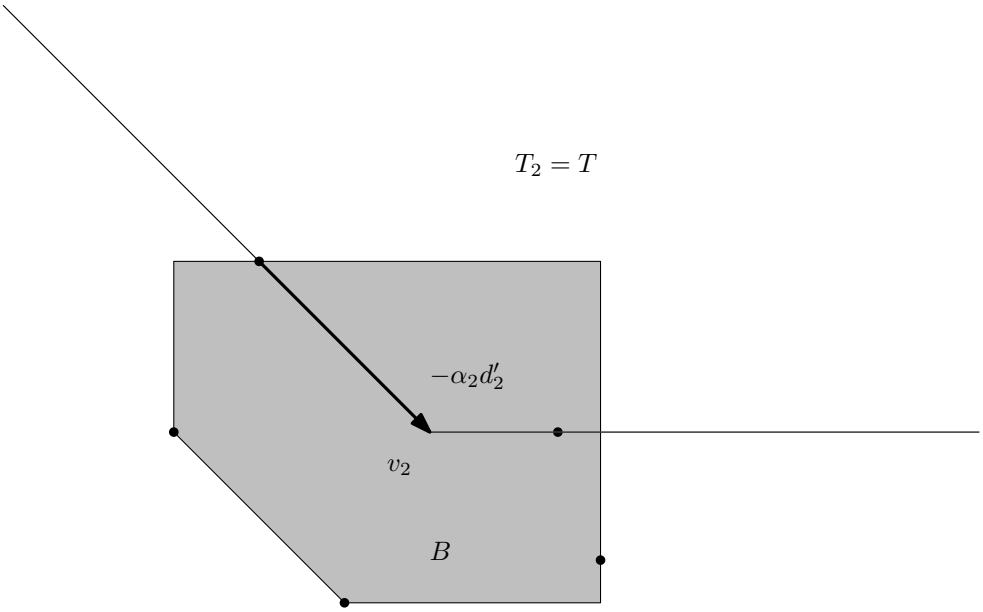



Figure 8.7: Translation of the cone: the second (i.e. the last) translation to $v_2$.

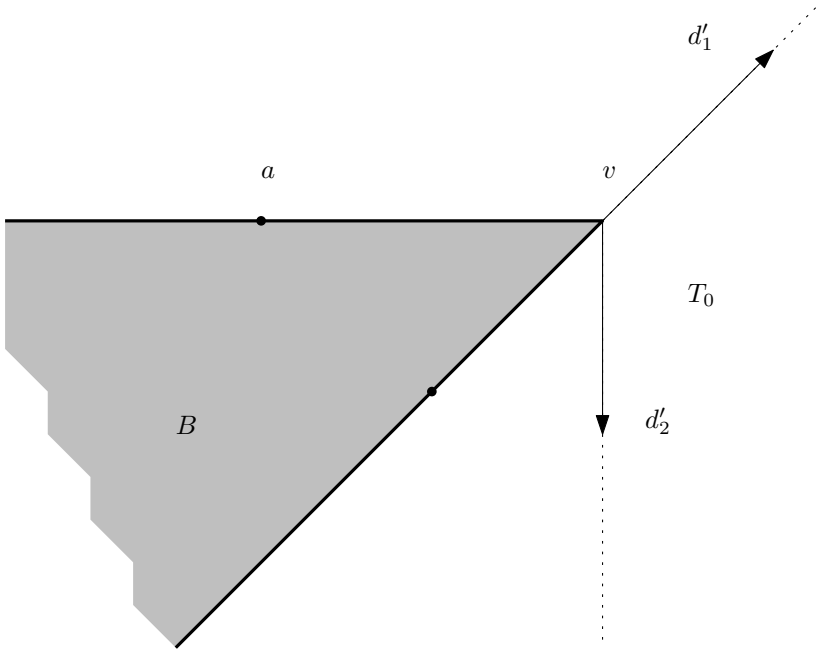



Figure 8.8: Translation of the cone: the initial state. The cone $T_0$ is given by $d'_1, d'_2$, the set $A$ by the black points, the set of directions is $D = \{d'_1, d'_2, va\}$.

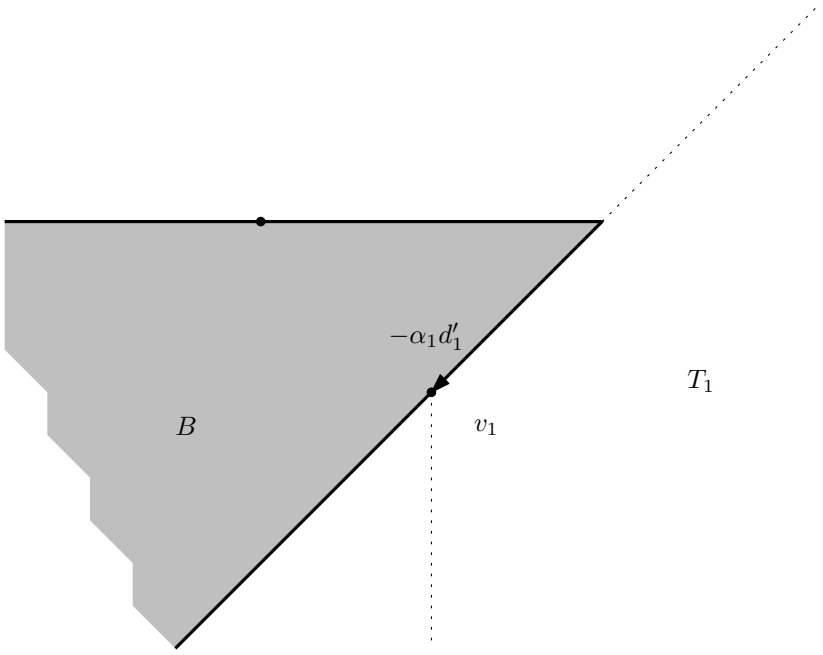



Figure 8.9: Translation of the cone: the first translation to $v_1$.

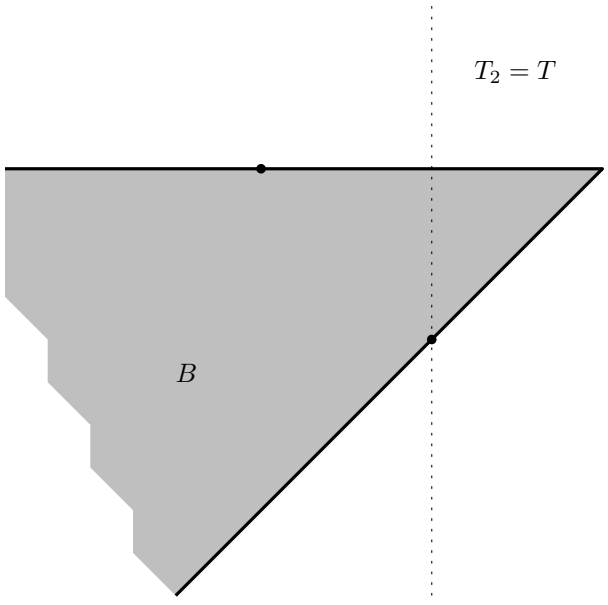



Figure 8.10: Translation of the cone: the second (i.e. the last) translation. The degenerated cone $T_2$ is the whole halfplane to the right of the dotted line.

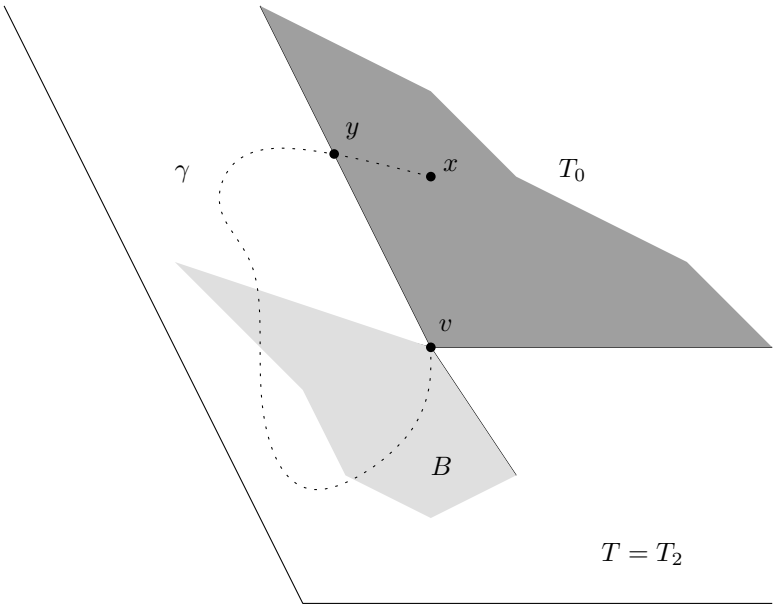



Figure 8.11: The considered path $\gamma$ from $v$ to $x$ through a boundary point $y$.

**Proof:** For $i = 0$, the claim of the lemma follows immediately from Lemma 8.6. For $i > 0$, we construct the proof by contradiction.

Let $i = 1$ or $i = 2$ and let us suppose that there is a point $x$ of $R_i$ in $T_0$ different from $v$:

$$\exists x \in (R_i \cap T_0) \setminus \{v\}.$$

Because both $x$ and $v$ belong to the same connected component $R_i$, it follows that there must be a (non-self-intersecting) path $\gamma$ in $B$ leading from $v$ to $x$. Due to construction of the cutter $T_0$ and due to the fact that $v$ is a convex vertex of $B$, we see that there is no path in $B \cap T_0$ leading from $v$ anywhere. Hence, the path $\gamma$ leads from $v$ somewhere out of $T_0$ instantly. Because it ends in $T_0$, the path must cross the boundary of $T_0$ in a point $y$ different from $v$:

$$\exists y \in (B \cap \partial T_0) \setminus \{v\}.$$

The situation for $i = 2$ is depicted on Figure 8.11.

Due to the $D$-convexity of $B$, the segment $[v, y]$ (being a part of a $D$-line with both endpoints in $B$) must be in $B$, but this is in contradiction with the fact that there is no path in $B \cap T_0$ leading from $v$ anywhere. It follows that in $(R_i \cap T_0)$, there is no other point than $v$. □

**Corollary 8.9** $A \cap (R \cap \operatorname{int} T) = \emptyset$.

**Proof:** Let us denote the difference between the interior of the cone in the final position and the cone in initial position by $T_\Delta$ :

$$T_\Delta = \operatorname{int} T \setminus T_0.$$

Due to the process of translation $T_0$ to $T_2 = T$ described above, we see that the set $T_\Delta$ does not contain any point from $A \cap R$ (because such a point would be hit during translation — remember that $R \subseteq M$):

$$((A \cap R) \cap (\operatorname{int} T \setminus T_0)) = A \cap (R \cap T_\Delta) = \emptyset.$$

On the other hand, we have $R \cap T_0 = \{v\}$ due to Lemma 8.8. It follows that

$$A \cap (R \cap T_0) = \emptyset.$$

The claim of the lemma is obvious now:

$$A \cap (R \cap \operatorname{int} T) = (A \cap (R \cap (\operatorname{int} T \setminus T_0))) \cup (A \cap (R \cap T_0)) = \emptyset.$$

□

Lemma 8.2 is proved by this corollary.

**Lemma 8.10**

*For both* $i, 1 \leq i \leq 2 : T_i \supset T_{i-1}$.

*In particular,* $v \in \operatorname{int} T_2$.

**Proof:** We know that the apex $v_i$ of the cone $T_i$ is a translation of $v_{i-1}$:

$$v_i = v_{i-1} - \alpha_i d'_i.$$

Every $x \in T_{i-1}$ can be expressed as

$$x = v_{i-1} + \sum_{j=1}^{2} \beta_j d'_j,$$

where every $\beta_j$ is nonnegative. So is $\alpha_i$, and hence

$$x = v_i + \alpha_i d'_i + \sum_{j=1}^{2} \beta_j d'_j = v_i + \sum_{j=1}^{2} \beta'_j d'_j,$$

where every $\beta'_j$ is nonnegative, and hence $x \in T_i$.

The inequality between $T_i$ and $T_{i-1}$ follows from the fact that $T_i = T_{i-1} - \alpha_i d'_i$ and $\alpha_i > 0$.

In particular,

$$v = v_0 = v_2 + \sum_{j=1}^{2} \alpha_j d'_j,$$

where $\alpha_1, \alpha_2 > 0$ and hence $v \in \operatorname{int} T_2$. □

## 8.4 The Cut

Let $R$ be the connected component of $B \cap T$ that contains $v$. Due to Lemma 8.3, we can apply the Cutter Lemma. We obtain the functionally $D$-convex $(D, A)$-polytope $B \setminus (R \cap \operatorname{int} T)$ and continue with the next iteration.

After the algorithm finishes, we have a functionally $D$-convex $(D, A)$-polygon $B \supseteq A$ and we know that all the convex vertices of $B$ are in $A$. Let

$$C = \{B_1, \ldots, B_k\}$$

be the set of all the connected components of $B$. Due to Proposition 4.7, we know that every $B_i$ is functionally $D$-convex. Due to Corollary 6.6, every $B_i$ is a tripod, or all the $D$-extremal points of $B_i$ are in $A$. In this latter case, we have

$$B_i = \operatorname{co}^D(A \cap B_i)$$

due to Proposition 4.6. If $B_i$ is a tripod, the equality

$$B_i = \operatorname{co}^D(A \cap B_i)$$

is a direct corollary of the following lemma:

**Lemma 8.11** *Let $B \supset A = \{a_1, a_2, a_3\}$ be a tripod. Then $B = \operatorname{co}^D(A)$.*

**Proof:** It is easy to see that the algorithm can yield the tripod in three steps and hence $B \supseteq \operatorname{co}^D(A)$. (The three corresponding cutters are denoted $T, U, V$ in Figure 8.12, $T$ is given by the convex angle $\langle a_3 c a_1 \rangle$, $U$ by the convex angle $\langle a_1 c a_2 \rangle$, $V$ by the convex angle $\langle a_2 c a_3 \rangle$.) For the opposite inclusion, it is enough

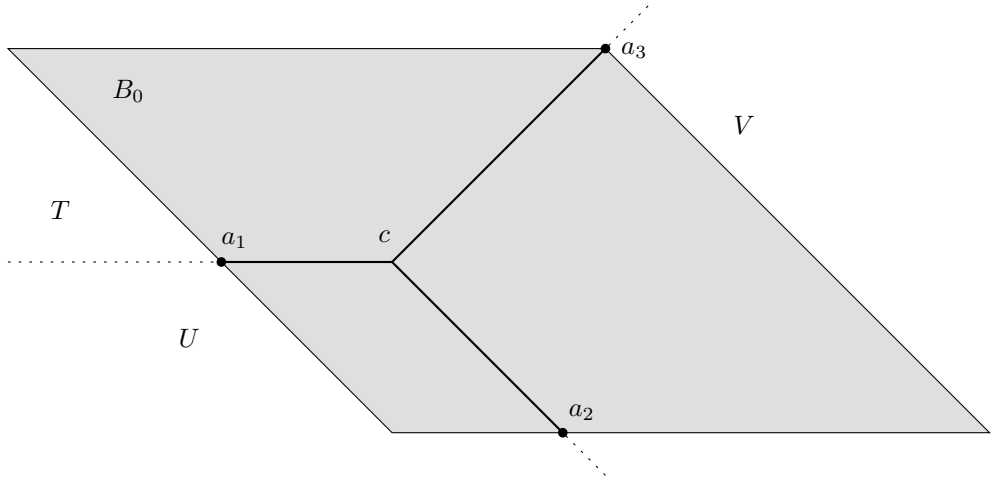



Figure 8.12: The creation of the tripod.

to show that every nonnegative $D$-convex function, that vanishes in $A$, vanishes in the center of the tripod, too. For this, we have two proofs. We start with the elementary one.

Let us consider the situation as in Figure 8.13. Let $f$ be a nonnegative $D$-convex function with

$$f(a_1) = f(a_2) = f(a_3) = 0.$$

For contradiction, let us suppose that $f(c) > 0$, where $c$ is the center of the tripod $B$. Let $Z = \bar{\Omega}_\varepsilon(a_2)$ be a closed neighbourhood of $a_2$ so small that

$$\forall z \in Z : f(z) < f(c).$$

Such a neighbourhood must exist, since $f(a_2) = 0 < f(c)$. Let us consider a $D$-line $\ell || ca_2$ crossing the interior of the segment $ca_3$ in a point $x$ and crossing the line $ca_1$ in a point $y$. This $y$ must lie on the ray opposite to $ca_1$. Let $z$ be a point in $Z \cap \ell$. We know that $f(z) < f(c)$. Due to the convexity of $f|_{cy}$ (the function $f$ restricted to the line $cy$) and with respect to the fact $f(a_1) = 0 < f(c)$, we also know that $f(y) > f(c)$. Similarly, $f(a_3) = 0 \leq f(x) < f(c)$ due to the convexity of $f|_{xc}$. Putting this all together, we have

$$f(z) < f(c) < f(y) > f(c) > f(x)$$

which is a contradiction with the convexity of $f|_\ell$.

We have shown that every nonnegative $D$-convex function $f$, vanishing in $A$, vanishes in $c$, too. Due to $D$-convexity, it must vanish on all the segments $ca_i$, and hence $B \subseteq \mathrm{co}^D(A)$. □

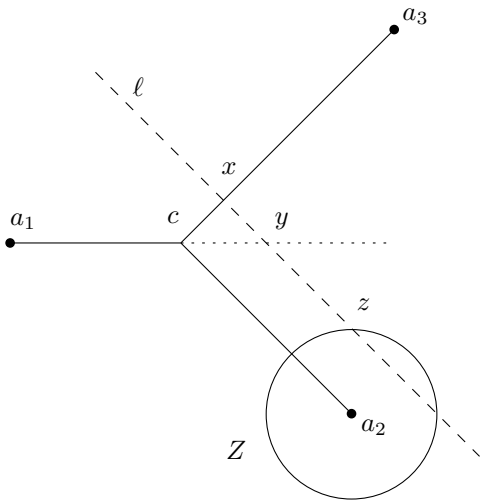



Figure 8.13: The tripod is functionally $D$-convex.

**Proof (another one):** In fact, we prove a slightly different (more general) lemma now.

Let $|D| = 3$. Suppose that we have three point set

$$A = \{t, u, v\}$$

and such a point $x$ lying inside the convex hull of $A$, that all the lines $xt, xu, xv$ are different $D$-lines. We show that

$$x \in \mathrm{co}^D(A).$$

Let us have a sequence of sets

$$A_i = \{t, u, v_i\},$$

where $v_i \to v$ as in Figure 8.14, that is,

$$\lim A_i = \{t, u, v\} = A.$$

Let $\ell_i$ be the $D$-line which is parallel to $xv$ and goes through $v_i$. Let us define the following points:

$$x_i = \ell_i \cap tx,$$
$$y_i = \ell_i \cap ux.$$

It is easy to check (e.g. using Proposition 4.6) that $\mathrm{co}^D(A_i)$ consists of the segments $tx, v_ix_i, uy_i$ and the triangle $xx_iy_i$. That implies

$$x \in \mathrm{Clust}_i(\mathrm{co}^D(A_i)).$$

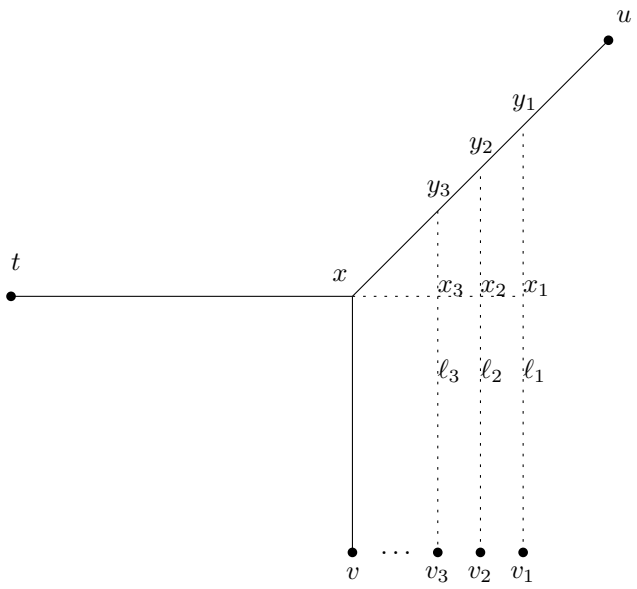



Figure 8.14: The sequence with limit in the shape of a tripod.

Due to Theorem 5.3, we see that $x \in \mathrm{co}^D(A)$. □

We have just shown

$$B_i = \mathrm{co}^D(A \cap B_i)$$

for every component $B_i$, and hence

$$\mathrm{co}^D(A) = \bigcup_{i=1}^{k} \mathrm{co}^D(A \cap B_i) = \bigcup_{i=1}^{k} B_i = B$$

due to Proposition 4.8.

**Corollary 8.12** *The algorithm given in the beginning of this chapter yields the functionally D-convex hull of the set A:*

$$B = \mathrm{co}^D(A).$$

# Chapter 9

# Complexity

In this chapter, we will discuss some complexity issues regarding the planar functionally $D$-convex hull and the described algorithm for its construction.

As before, we suppose that we have a finite set $A \in \mathbf{R}^2$ and a finite simple set of directions $D \in \mathbf{R}^2$ throughout this chapter.

## 9.1 Complexity of the Hull

Before we move on to the analysis of the complexity of the hull, we give one more definition — the void side of an edge.

Let $P$ be a $(D, A)$-polygon. Let us look at an edge $e$ of $P$. It is a segment formed by one or more edge atoms of $P$, i.e., edges of the underlying arrangement such that at least one of the two adjacent facets does not belong to $P$. Moreover, no other edge of $P$ crosses the interior of $e$. It follows that along at least one side of $e$, all the adjacent facets do not belong to $P$. Let us call the union of these facets along one side *void side* of $e$. Any edge of $P$ has one or two void sides.

As an example, on Figure 9.1, there is a $(D, A)$-polygon consisting of the union of segment $e_1$ and the grey tetragon. The set $A$ consists of the three points marked by disks and the set $D$ consists of three directions. The arrangement lines are dotted on the figure. Edge $e_1$ is the only one having both sides void: void side $S_1$ consists of one triangular facet of the underlying arrangement and void side $S_2$ consists of one tetragonal facet. The edge $e_2$ has only one void side,

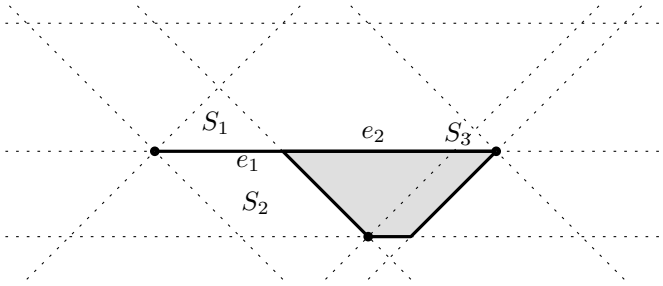



Figure 9.1: An example illustrating void sides.

it is denoted by $S_3$ and consists of two facets.

Now we start with the analysis of the complexity of the hull. First, we prove the following lemma.

**Lemma 9.1** *Let $B$ be a functionally $D$-convex $(D, A)$-polygon and let $\ell$ be a $(D, A)$-line. Let us go along this line and index all the vertices of $B$ lying on $\ell$ by their occurence: let the first one be $v_1$ and the last one $v_k$. Then both of the following statements are true:*

1. *$k \leq 6$, i.e., every $(D, A)$-line contains at most six vertices of $B$.*
2. *If there are two vertices $v_m, v_{m+1}$ forming an edge of $B$ then at least one of the vertices $v_1, v_k$ is convex.*

**Proof:** Let us suppose that $k \geq 2$. Due to $D$-convexity of $B$, we know that the set $B$ contains the whole segment $[v_1, v_k]$. Let

$$V = \{v_1, v_2, \ldots, v_k\}$$

be the set of all vertices of $B$ that lie on $\ell$. Let $\ell^+$ be one of the closed halfplanes given by $\ell$ and let $\ell^-$ be the opposite one.

Let $v_i$ be an vertex from $V$ such that there is an edge $[v_i, u_i]$ such that $u_i \notin V$. (If there is no such vertex then $k = 2 < 6$ and both the vertices $v_1$ and $v_2$ are convex.) Without loss of generality, we may suppose that $[v_i, u_i] \subset \ell^+$. If the edge $[v_i, u_i]$ has both sides void, then there is no edge $[v_j, t_j] \subset \ell^+, t_j \notin (V \cup \{u_i\})$, due to $D$-convexity of $B$. If the edge $[v_i, u_i]$ has only one side void, then there is exactly one edge $[v_j, t_j] \subset \ell^+, t_j \notin (V \cup \{u_i\})$, due to $D$-convexity of $B$, again.

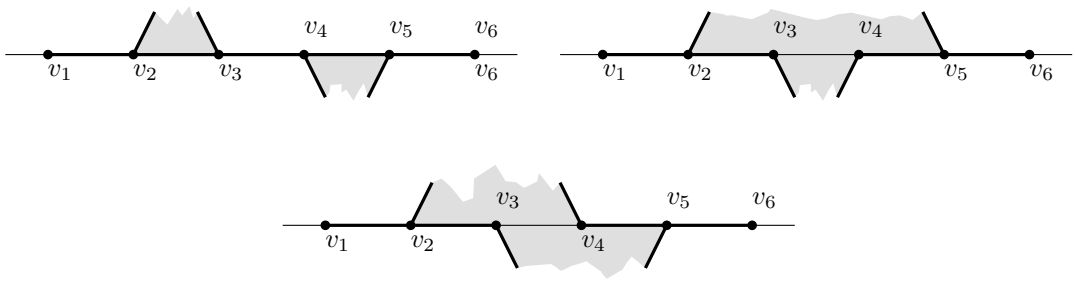



Figure 9.2: The three situations where a $(D, A)$-line contains six vertices of a $D$-convex $(D, A)$-polygon.

We see that each of the halfplanes $\ell^+$ and $\ell^-$ contains at most two edges of the form $[v, u]$, where $v \in V$ and $u \notin V$. Because for every vertex $v \in V \setminus \{v_1, v_k\}$ such an edge must exist, we see that between the points $v_1$ and $v_k$, there lie at most four other vertices of $B$ on $\ell$. Up to symmetry, there are three orderings of these edges $[v, u]$ possible, all of them are illustrated on Figure 9.2. This finishes the proof of part 1 of this lemma.

Let us finish the proof of part 2. Let $e = [v_m, v_{m+1}]$ be an edge of $B$. Without loss of generality, we may assume that $\ell^+$ contains a void side of the edge $e$.

Let us suppose that $v_1$ is non-convex. Then there is an edge $e_1 = [u_1, v_1]$ of $B$ such that $u_1 \notin V$ and one of the following is true:

(i) $e_1 \subset \ell^+$ and the intersection of the interior of the triangle $(u_1, v_1, v_2)$ with $B$ is empty:
$$\operatorname{int} \triangle(u_1, v_1, v_2) \cap B = \emptyset.$$

(ii) $e_1 \not\subset \ell^+$ and the edge $[v_1, v_2]$ has no void side in $\ell^+$.

Analogously, let us suppose that $v_k$ is non-convex. Then there is an edge $e_k = [u_k, v_k]$ of $B$ such that $u_k \notin V$ and one of the following is true:

(iii) $e_k \subset \ell^+$ and the intersection of the interior of the triangle $(u_k, v_k, v_{k-1})$ with $B$ is empty:
$$\operatorname{int} \triangle(u_k, v_k, v_{k-1}) \cap B = \emptyset.$$

(iv) $e_k \not\subset \ell^+$ and the edge $[v_k, v_{k-1}]$ has no void side in $\ell^+$.

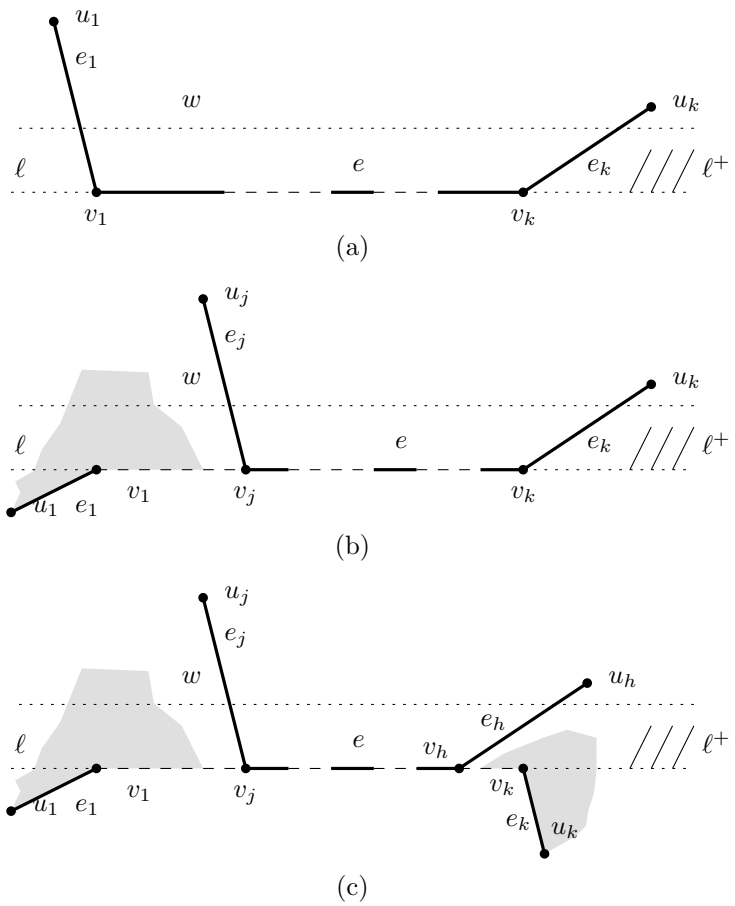



Figure 9.3: The dotted $D$-line $w$ always witnesses that $B$ is not $D$-convex.

Now if (i) and (iii) were true, then any $D$-line crossing the interior of both $e_1$ and $e_k$ would witness that $B$ is not $D$-convex (see Figure 9.3(a)) — a contradiction.

If (ii) and (iii) were true, then there would have to be an edge $e_j = [v_j, u_j] \subset \ell^+, u_j \notin V, j \leq m$, because the edge $e$ has a void side in $\ell^+$ but $[v_1, v_2]$ has not. Then any $D$-line crossing the interior of both $e_j$ and $e_k$ would witness that $B$ is not $D$-convex (see Figure 9.3(b)) — a contradiction. An analogous argument can be used for the case when (i) and (iv) were true.

Similarly in the last case, if (ii) and (iv) were true, then there would have to be an edge $e_j = [v_j, u_j] \subset \ell^+, u_j \notin V, j \leq m$ (because the edge $e$ has a void side in $\ell^+$ but $[v_1, v_2]$ has not), as well as there would have to be an edge $e_h = [v_h, u_h] \subset \ell^+, u_h \notin V, h \geq m + 1$ (because the edge $e$ has a void side in $\ell^+$ but $[v_k, v_{k-1}]$ has not). Again, any $D$-line crossing the interior of both $e_j$ and $e_h$

would witness that $B$ is not $D$-convex (see Figure 9.3(c)) — a contradiction.

Note that the discussed cases also cover the situations where $m = 1$ (and, in particular, where $k = 2$ and hence $m = 1$, too). □

**Lemma 9.2** *Let $B$ be a functionally $D$-convex $(D, A)$-polygon and let $v_c$ denote the number of convex vertices of $B$. Then the number $v_n$ of non-convex vertices of $B$ is linear in $v_c$:*

$$v_n = O(v_c).$$

**Proof:** Every non-convex vertex of $B$ lies on a $(D, A)$-line containing an edge of $B$. According to Lemma 9.1, there are at most six vertices on every such line, and at least one of them is convex. It follows that

$$v_n \leq 5v_c.$$

Performing an easy routine analysis of all the possible cases, one could improve this result even to $v_n \leq 3v_c$. We do not present the details proving this improved bound here, because the desired linear dependence is already proved. □

**Corollary 9.3** *In the plane, the number $v$ of vertices of the functionally $D$-convex hull of the set $A$ is linear in $|A|$:*

$$v = O(|A|).$$

*In particular, it does not depend on the size of $D$.*

**Proof:** According to Corollary 8.12, the algorithm finishes with $B = \mathrm{co}^D(A)$. The algorithm ends as soon as there are no more convex vertices of $B$ outside of $A$. Hence the number $v_c$ of convex vertices of $B$ is at most $|A|$:

$$v_c \leq |A|.$$

The rest is a direct corollary of Lemma 9.2. □

With a little extra effort, we can prove the linear bound for every approximation of the hull that our algorithm produces:

**Lemma 9.4** *Let $B = B_{s-1}$ be the approximation of the functionally $D$-convex hull $\mathrm{co}^D(A)$ before the $s$-th iteration of the algorithm. Then the $D$-convex $(D, A)$-polygon $B$ has $O(|A|)$ vertices.*

**Proof:** Let us define an *edge segment* for the purposes of this proof. Let $[v_1, v_2]$, $[v_2, v_3]$, ..., $[v_n, v_{n+1}]$ be an inclusion-maximal chain of $n$ consecutive edges of $B$ such that all of them lie on the same line. Then the segment $[v_1, v_{n+1}]$ is called an *edge segment*. Every edge lies on an edge segment, clearly. As an example, let us have a look at Figure 9.2 and the line $v_1v_6$. On the diagram in the upper left corner, there is one edge segment $[v_1, v_6]$. On each of the two remaining diagrams, there are two edge segments: $[v_1, v_3]$ and $[v_4, v_6]$.

Note that in every edge segment, there is a point from $A$ on at least one of the edges (due to construction; it is the point hit when translating a cutter, or one of the points on the boundary of $B_0$).

Every vertex of $B$ is either an isolated vertex, or it is an intersection of two or more edges. We say that a vertex $v$ of $B$ *belongs* to a point $a \in A$ if $v = a$ or $a$ lies on an edge which lies in the same edge segment as $v$. Clearly, every vertex of $B$ belongs to a point from $A$. Every non-isolated vertex belongs to more than one point.

Now, let us compute how many of the vertices of $B$ can belong to a point $a$ from $A$. If $a$ is an isolated vertex then only $a$ belongs to $a$. If $a$ lies in the interior of $B$ then no vertex belongs to it. If $a$ lies in the interior of an edge then all the vertices belonging to $a$ lie on the edge segment containing this edge. Lemma 9.1 shows that there are at most six vertices on every $(D, A)$-line (and on every edge segment, too), and hence at most six vertices belong to $a$ in this case. In the last case, if $a$ is a vertex of $B$ then all the vertices belonging to $a$ lie on the edge segments passing through $a$. Exploring all the possible cases, one can find out that the degree of any vertex of any $D$-convex polygon is at most four. Hence, there are at most four edge segments passing through $a$. (In fact, there are at most three such edge segments, because if the degree is four, then the vertex is the intersection of two edge segments.)

It follows that there is at most constant number of vertices belonging to any point from $A$. Since every vertex belongs to a point from $A$, there are at most $O(|A|)$ vertices in $B$. □

## 9.2 Complexity of the Algorithm

The previous section was concerned in complexity of the hull. In this section, we discuss complexity of the algorithm. In particular, from the maximum number of its iterations and the running time of one iteration, we compute the time complexity of the algorithm.

### 9.2.1 The Number of Iterations

We start with an easy and straighforward estimate. The finiteness of the algorithm follows from the fact that the number of vertices of the underlying arrangement $H$ that belong to the current approximation $B$ of the hull strictly decreases in each iteration. In the worst case, we remove only one vertex $v$ of $H$ from $B$ in one step. The arrangement $H$ consists of at most $|A||D|$ lines. Every $(D, A)$-line given by a point $a$ from $A$ and a direction from $D$ intersects all the $(D, A)$-lines given by the remaining $|A| - 1$ points and $|D| - 1$ directions, thus yielding at most $(|A| - 1)(|D| - 1)$ vertices of $H$ apart from $a$. Every vertex is given by at least two lines:

$$\#(\text{vertices in } H) \leq \frac{|A||D|(|A| - 1)(|D| - 1)}{2} + |A|.$$

We cannot remove the points from $A$, hence in the worst case, the number of iterations of the algorithm is

$$\frac{|A||D|(|A| - 1)(|D| - 1)}{2} = O(|A|^2|D|^2).$$

We are going to improve this crude estimate in the following paragraphs, but this needs deeper analysis of the algorithm.

Let us look at the wedge-shaped cutters used in the algorithm. Every such cutter is given by one or two closed halfplanes. Any of these halfplanes is given by a point from $A$ hit during the translation of the cutter, a $(D, A)$-line $\ell$ through this point given by a direction from $D$, and a boolean value indicating which one of the two opposite halfplanes given by $\ell$ we mean. This may be defined as the indicator whether a fixed point lies in the halfplane or not. Let us formalize this in the following paragraph.

Let $p$ be a fixed point not lying on any $(D, A)$-line. Let $\mathcal{H}$ be the set of all $(D, A)$-halfplanes that may come into play when translating the cutters in the

algorithm:

$$\mathcal{H} = \bigcup_{a\in A, e\in D, f_p\in\{0,1\}} (a, e, f_p)$$

where the halfplane is given by the $(D, A)$-line through $a$ and parallel to $e$ and the indicator $f_p$ is one if the point $p$ belongs to the halfplane and zero if not. (Especially note that this set $\mathcal{H}$ is not only a plain set of all $(D, A)$-halfplanes, but it is also important which point from $A$ defines the $(D, A)$-line. In other words, if some two different points $a, b$ from $A$ lie on the same $(D, A)$-line parallel to a direction $e \in D$ then $(a, e, f_p)$ and $(b, e, f_p)$ define the same halfplane in the usual sense, but for our purposes they are different.)

**Lemma 9.5** *The algorithm uses every halfplane from the set $\mathcal{H}$ at most once, i.e., in every iteration, the cutter $T$ in a fixed position is defined by one or two halfplanes such that none of them has defined any cutter in a fixed position in any of the previous iterations.*

**Proof:** Let $p_1, p_2$ be very large numbers such that the point $p = [p_1, p_1+p_2]$ is not lying on any $(D, A)$-line and it is very distant from any vertex of the arrangement of all $(D, A)$-lines. The exact distance and position is not important, it is enough to know that it is somewhere "far above and even further to the right" of the set $A$.

Let $a$ be a point from $A$ and let $e$ be a direction from $D$. Without loss of generality, we may assume that this direction is horizontal. Let $h$ be the $(D, A)$-line parallel to $e$ that goes through $a$. Then $h^+ = (a, e, 1)$ is the halfplane from $\mathcal{H}$ that lies above the $(D, A)$-line $h$. We are going to show that in the algorithm, this halfplane is used at most once.

Using the same notation as in the previous chapter, let $B_s$ denote the approximation of the hull after the $s$-th iteration of the algorithm. Similarly, let $T_s$ denote the cutter in a fixed position in the $s$-th iteration of the algorithm. If $h^+$ is opposite to one of the four halfplanes whose intersection is $B_0$, then it is surely not used for any cutter in a fixed position, because the cutters cut the interior of (what remains of) $B_0$.

Let $h^+$ be used in the $s$-th iteration, $s > 0$. Due to Lemma 8.10, if it were used in a subsequent iteration $r > s$ again, then there would have to be a convex vertex $v_r$ of $B_{r-1}$ lying on or above the line $h$ and in the same connected component $M_{r-1}$ of $B_{r-1}$ as $a$. We show that if there is such a vertex, no cutter for it in a fixed position is defined by $h^+$.

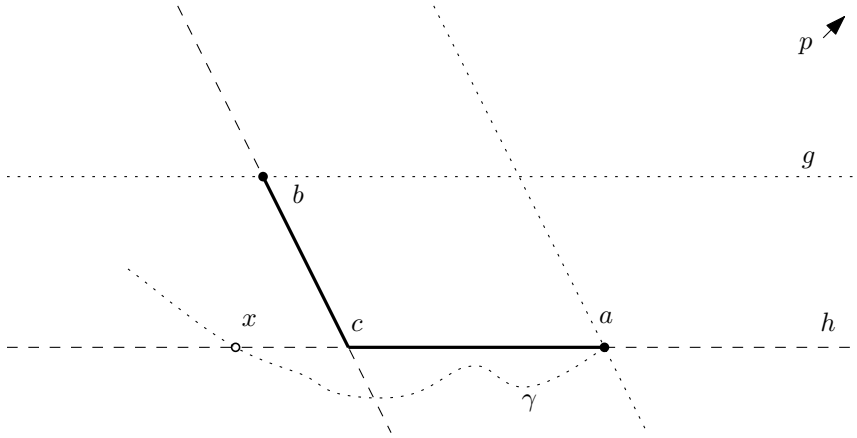



Figure 9.4: The situation regarding the $s$-th iteration of the algorithm.

If $T_s$ is degenerated, i.e., it is equal to the halfplane $h^+$, then above $h$, there is no point of the connected component $M_s$ of $T_s$ that contains $v_s$. Because $M_s \supseteq M_{r-1}$, we are done.

Let us suppose that $T_s$ is not degenerated, i.e., that it is the intersection of two halfplanes. One of them is $(a, e, 1)$, let $(b, f, 1)$ be the other one, let $g$ be the $(D, A)$-line through $b$ parallel to $e$, and let $c$ be the apex of the cutter $T_s$. Then the situation is as illustrates Figure 9.4. Due to $D$-convexity of $B_s$, the interior of the intersection of the three halfplanes

$$\operatorname{int}\left((a, e, 1) \cap (b, f, 1) \cap (b, e, 0)\right)$$

contains no point of $B_s \supseteq B_{r-1}$ and so does the interior of the intersection of three halfplanes

$$\operatorname{int}\left((a, e, 1) \cap (b, f, 1) \cap (a, f, 0)\right).$$

It follows that there is no point of $M_s \supseteq M_{r-1}$ in the interior of the intersection of two halfplanes

$$\operatorname{int}\left((a, e, 1) \cap (b, f, 1)\right).$$

Thus the only place for $v_r$ remains the intersection of the halfplanes $(a, e, 1)$ and $(b, f, 0)$.

In the connected component $M_{r-1} \supset \{a, v_r\}$, there must be a connected path $\gamma$ leading from $a$ to $v_r$, and this path must intersect the ray opposite to $ca$ in a point $x$. Due to $D$-convexity of $M_{r-1}$, the segment $[x, a]$ must be in $M_{r-1}$, and it follows that the segment $[a, c]$ belongs to $M_{r-1}$ as well. Now when we know that $c$ belongs to $M_{r-1} \subseteq B_{r-1} \supset A \ni b$, due to $D$-convexity the segment $[b, c]$ must belong to $M_{r-1}$, too.

Let $U$ be the cutter cutting $v_r$ in initial position, i.e., before we start translate it to the fixed position $T_r$. There are two possibile shapes of $U$. Either the bounding ray of $U$ parallel to $e$ intersects the line $bc$ (we "cut from right") or it does not (we "cut from left").

In the former case, the vertex $v_r$ must lie above the line $g$ (due to $D$-convexity of $M_{r-1}$). Then the bounding ray of $U$ parallel to $e$ can never hit the point $a$ during translation, because it stops at point $b$ if not sooner.

In the latter case, let $u$ be the bounding ray of $U$ not parallel to $e$. Note that the line supporting $u$ crosses the interior of the ray opposite to $ca$. Because the translation should stop at $a$ from above, we must first move the ray $u$ to the right in the direction parallel to $e$. If $v_r$ lies above the line $g$, then the second translation of the cutter either stops at point $b$ if not sooner (by the same reasons as in the previous case), or misses $a$ because the cutter was not moved to the right far enough. If $v_r$ lies on the line $g$ or below then during the first translation, the moved ray $u$ hits $b$ and still, the line supporting this moved ray crosses the interior of the ray opposite to $ca$. Hence the translated cutter can never hit the point $a$. □

Note that in this proof, we used the fact that the stopping point lies in $M$ (i.e., in the same connected component as the convex vertex being cut). In the earlier version of the algorihtm given in the article [FM], we considered all the points from $A$ and not only those lying in $M$. Without the present modification, Lemma 9.5 would not be true in general. We give an counterexample to this.

Figure 9.5 shows the counterexample. There are illustrated five iterations of the algorithm on the figure. The set $A$ is given by the four black points $a_1, a_2, a_3, a_4$. The set of directions is $D = \{(1,0),(1,1),(1,-1)\}$. In each iteration $s$, the set $B_{s-1}$ (i.e., the current approximation of $\mathrm{co}^D(A)$ *before* the cut) is the union of the gray polygon with the set $A$, the vertex being cut is denoted by $v_s$ and the cutter in a fixed position $T_s$ is marked by the dotted wedge. As in the previous proof, let the reference point $p$ be far above and even further to the right of the set $A$. We see that $T_2$ and $T_5$ share the same halfplane $(a_1,(1,0),1)$ — a counterexample.

Now we are ready to state the following lemma regarding the number of iterations of the algorithm:

**Lemma 9.6** *The algorithm given in previous chapter finishes in at most* $2|D||A|$ *iterations.*

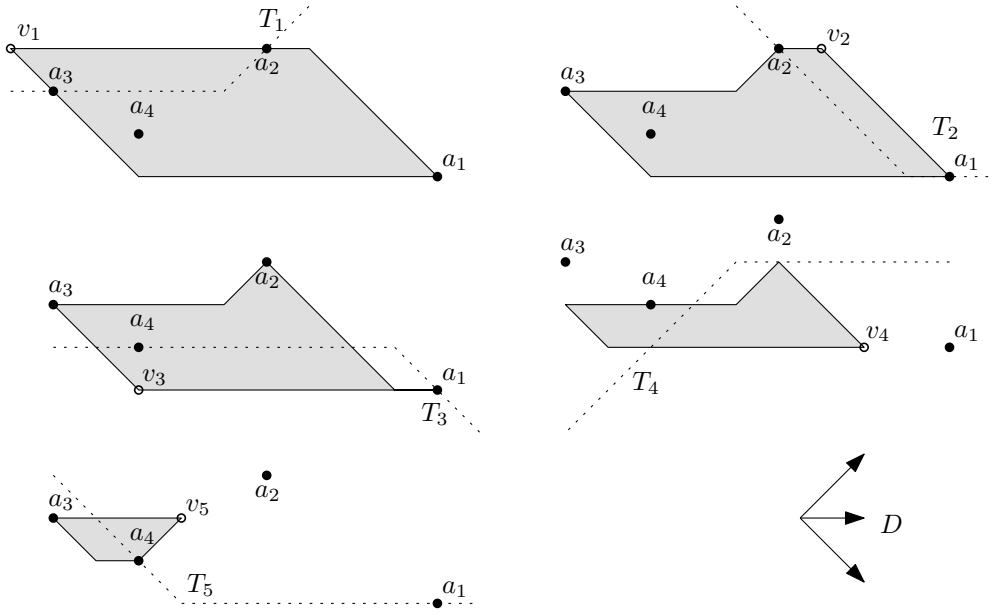



Figure 9.5: The earlier version of the algorithm uses the same halfplane $(a_1, (1, 0), 1)$ twice.

**Proof:** Again, consider the set $\mathcal{H} = \bigcup_{a \in A, e \in D, f_p \in \{0,1\}} \{a, d, f_p\}$ defined in this section. It is easy to compute its size:

$$|\mathcal{H}| = 2|D||A|.$$

The statement of this lemma follows from the fact that in every iteration of the algorithm, we use at least one halfplane from the set $\mathcal{H}$, and we never use the same halfplane twice. □

It follows that for a fixed set of directions $D$, the number of steps of the algorithm is linear in the number of points of the set $A$.

### 9.2.2 The Running Time of one Iteration

To finish the analysis of the running time of the algorithm, it remains to compute the time complexity of one iteration. Each iteration consists of these steps:

**Vertex** Find a convex vertex to cut.

**Cutter** Find the corresponding cutter.

**Translation** Translate the cutter to a fixed position.

**Cut** Apply the cutter.

In the following paragraphs, we propose an implementation of one iteration of the algorithm and of each of these steps such that its running time is reasonably short.[1] We do not provide a proof that our implementation is optimal, but we believe that it is close to optimum.

First, we will prepare some data structures[2] from the input. They stay immutable throughout the whole run of the algorithm. Here we just say what properties these structures should have. We postpone the explanation of the reason why to create such structures and how to use them to the analysis of each particular step that follows the description of the initialization phase.

We create a set $D^* = D \cup (-D)$ containing all the vectors from the set $D$ and all the opposite vectors. This set will be sorted clockwise and this takes $O(|D| \log |D|)$ time.

Note that there are exactly $2|D|$ possible shapes of the cutters. The shape of the cutter depends solely on the good separating halfplane. Let us create two-dimensional layered range trees (or two-dimensional range trees with fractional cascading) for the set $A$. Every such tree will be constructed with respect to a different shape of the cutter, i.e., the search ranges are paralellepipeds (instead of the traditional axis-parallel rectangles) with their edges parallel to the pair of lines supporting the rays of the corresponding cutter shape. Because every shape shares the same supporting lines with its point-symmetrical counterpart, there are exactly $|D|$ such trees. Every such tree can be built in $O(|A| \log |A|)$ time. These trees will be called *point trees*.

Next, we intialize some dynamic data structures that will reflect the current state of the algorithm.

Of course, we must prepare the initial polygon $B$. This means to determine (any) direction from $D$ and find the boundary $(D, A)$-lines parallel to that direction. This procedure is done twice. This can be easily done in one passage through $A$, and hence in $O(|A|)$ time. Using the point trees mentioned before, this can be sped up to $O(1)$ (by simply reporting the leftmost node and the rightmost node in the associated structure of the root of the corresponding tree).

[1] For practical reasons (given in Chapter 10), we have not converted this description of the implementation to a program and we rather used another one based on Nef polyhedra. Nevertheless, there should be no obstacles for implementing this as proposed.

[2] To find the appropriate structures, we often referred to de Berg et al. [dBvKOS00].

The current $(D, A)$-polygon $B$ can be represented as a list of its connected components and every component can be represented as a sequence of its edges such that one can decide in constant time whether a vertex of the component is convex or not. An isolated vertex $a$ can be represented as single edge $e_a = [a, a]$. Every edge should know which connected component it belongs to (for example, there can be a pointer from the edge to the component).

For every direction $f \in D$, we will need to find edges of the polygon $B$ intersected by $(D, A)$-lines parallel to $f$. To this end, we will create a self-balancing binary search tree for every direction. Because we start with four edges only, this could be done in $O(|D|)$ time. We will call these trees *edge trees*.

Next, we would like to have the *set of candidates* $C$, i.e., the set of convex vertices of $B$ that are not in $A$. This can be a set of pointers into $B$, and since we want to remove the vertices from $B$ and $C$ simultaneously, the corresponding points in $B$ should point backwards to $C$. This set could be initialized during the construction of $B$ in constant time.

Putting this all together, we see that the initial phase takes $O(|D| \log |D| + |D||A| \log |A|)$ time.

This concludes the analysis of the initial phase. Now, let us analyse the time complexity of every step of one iteration.

### Vertex

Because we have a set of candidates $C$ an we choose arbitrary element from the set, this step can be done in constant time.

### Cutter

To find the good separating halfplane, we can use the idea described in Chapter 8. We take the line supporting one of the edges incident to the vertex $v$ being cut and rotate it a little around $v$ so that it becomes separating line and is not parallel to any direction in $D$. Based on this idea, we may determine the shape of the cutter as follows: let $d_e$ be the direction of the edge entering $v$. Then the shape of the cutter is given by the two directions $d_n$ and $-d_e$, where $d_n$ is the vector following $d_e$ in $D^*$ and $-d_e$ is the vector opposite to $d_e$. Both these vectors can be retrieved from $D^*$ in constant time.

### Translation

Moving the cutter means finding appropriate point from $A$ and check it against the current connected component $M$ (and all of this is done twice). Note that if we translate the cutter along the ray as described in Chapter 8 and we first hit a point from $A$ that is not in $M$ (the same connected component as the vertex being cut off), then the cutter will never stop in this direction. (The argument for this is similar as in the proof of Lemma 8.8: if the first point hit $a_1$ is not in $M$ and if some more distant point $a$ was in $M$, then there would be a $D$-line passing through $a_1 \notin M$ and intersecting the connected path between $v$ and $a$ in a point $x \in M$ — a contradiction.) Thus it is enough to find the closest point from $A$ and check it against $M$.

Using the brute-force approach, the set $A$ can be searched through in $O(|A|)$ time. Using the point trees, this can be done faster. The point tree corresponding to the shape of the cutter can be searched through in $O(\log |A| + p)$ time, where $p$ is the number of reported points. The query algorithm in such a tree can be adapted easily so that it reports just the point that is the closest one to a specific line, and the running time will be $O(\log |A|)$.

One may find the connected component which a point $a \in A$ belongs to by finding an intersection of the polygon $B$ with arbitrary $(D, A)$-line passing through $a$. Because of the $D$-convexity of $B$ and because $A \subseteq B$, every $(D, A)$-line intersects exactly one connected component of $B$. Thus we choose arbitrary direction from $D$ and in the corresponding edge tree, we find any edge intersecting the line given by the chosen direction and passing through $a$. Because we have $O(|A|)$ edges in the tree, this can be done in $O(\log(|A|))$ time.

### Cut

The cut includes updating all the dynamic data structures to reflect the new state. We have three structures: the forest consisting of $|D|$ edge trees, the set of candidates $C$, and the current polygon $B$.

Before we start the analysis of this step, we prove the following lemma:

**Lemma 9.7** *Every cut removes at most two vertices from $B$.*

**Proof:** Every vertex $v$ of $B$ originates as the intersection of two $(D, A)$-lines.

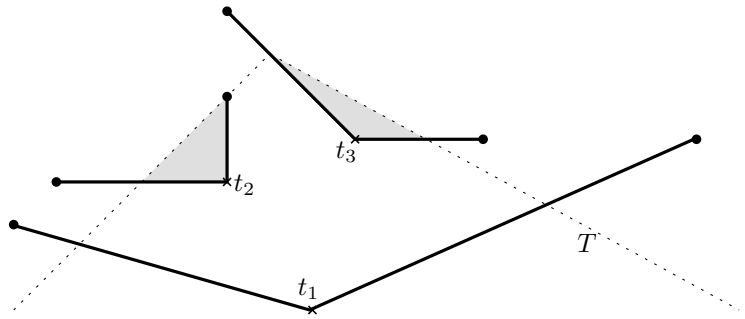



Figure 9.6: The three types of removed vertices (marked by crosses) $t_1$, $t_2$ and $t_3$ with their incident segments. The support points are the black dots. Note that the latter two types force the triangle spanned by the incident segments and the cutter (marked by the dotted rays) also belong to $B$ (due to $D$-convexity).

Each line goes through a point from $A$ which *supports* the line; this is one of the points which stopped the cutter during translation (even in the initialization phase when $B$ is a paralellepiped, the four lines bounding $B$ have their support points). Note that, due to $D$-convexity, segment $[v, a_1^v]$ belongs to $B$ and so does segment $[v, a_2^v]$, where $a_1^v$ and $a_2^v$ are the points supporting the lines that formed $v$. Let us call these segments *incident* to $v$.

Let $T$ be the cutter in a fixed position for a cut that removes a vertex $t$. Note that the supporting points lying on the segments incident to $t$ cannot lie in the interior of $T$, otherwise the translated cutter would stop at them. Let us consider all the possible types of removed vertices. There are three types: the incident segments intersect the interior of different bounding rays of the cutter, or they both intersect the interior of the same bounding ray of the cutter, or one of the segments intersects the apex of the cutter (cf. Figure 9.6). Exploring all the possible combinations of these types, we could find out that if we want to remove two vertices in one cut, there are just two eligible cases (both of them combining two vertices of the first type), see Figure 9.7. All the other combinations are not admissible, because they either spoil $D$-convexity, or the incident segments are intersecting in such a way that a vertex loses support point for its ray (and hence such a combination is invalid since such a vertex could not be created by the algorithm), or for other resons.

Now we can try to combine three vertices, i.e., add a third vertex (of the first type) to the two allowed combinations. It is easy to check that no such combination is possible, and the lemma is proved. □

Now we know that the cut removes at most two vertices from $B$ and adds

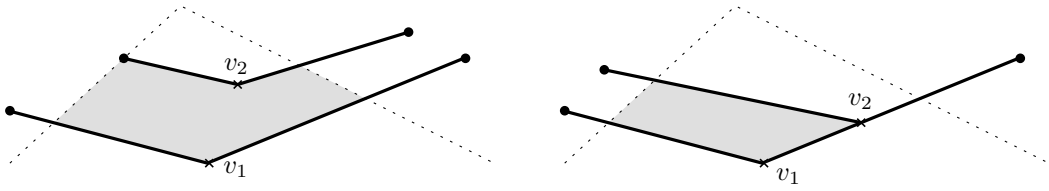



Figure 9.7: The only two possible cases of removing two vertices in one cut. The support points are the black dots, the removed vertices are marked by crosses, the incident segments by full segments, the cutter is marked by the dotted rays. The gray area must belong to $B$ due to $D$-convexity.

at most 12 new vertices to $B$ and hence we update (or delete) $O(1)$ edges in $B$. The new vertices are defined as the intersection of two $(D, A)$-lines with the edges of $B$ and thus can be found using the edge trees. So we perform $O(1)$ search operations on two edge trees and $O(1)$ insert and delete operations on all $|D|$ edge trees. Every such operation takes $O(\log(|A|))$ time. Because we have localized the cut edges in $B$, we may update $B$ (and $C$, simultaneously) in $O(1)$ time.

The last thing remaining is to update the mapping from the edges to connected components. This must be done whenever an existing edge gets into another component. In our algorithm, this happens if and only if a component is split into two parts. In such a case, we must travel around the new component and map every edge to it. Since the component can contain almost (up to one) all edges of $B$ in general, this one remapping takes $O(|A|)$ time. If there are $k$ connected components in the final hull $\mathrm{co}^D(A)$ then we perform this remapping $(k-1)$ times and hence all the remappings take $O(k|A|)$ time. (Note that this is the total figure for the whole algorithm, not for just one iteration.)

But this could be improved. If we first decide which of the two parts the old component split into is smaller then we can remap this smaller one and thus spare some time. Let us describe how to do it in detail.

We can decide which one of the two components is smaller by lock-step, i.e., we travel around both components alternately edge by edge. In time $t$, we reach the point where we started in one of the components. This is the smaller one and its size is $s_2 = t/2$. Its remapping takes $s_2$ time (we remap every edge). We see that the total time for one splitting is $3s_2$, where $s_2$ is the size of the smaller piece the original component split into. The following lemma gives the total time for all splittings.

**Lemma 9.8** *Using the described strategy, the total time complexity of all splittings is*

$$O(|D||A| \log |A|).$$

**Proof:** Let us imagine that whenever we add an edge to a component, the edge pays an entrance fee. Let this amount be $\log_2(4(c|A|+1))$ credits, where $c > 0$ is a suitable constant to be specified later. Whenever a component of size $s$ is split into two parts of sizes $s_1$ ans $s_2$, $s_1 \geq s_2$, it must pay $s_2$ credits. If the total credit paid for all splittings is $S$ credits then the total time complexity of all splittings is $3S = O(S)$. Because in each of the $O(|D||A|)$ iterations we add $O(1)$ edges, the total credit involved is $O(|D||A| \log_2(4(c|A|+1)) = O(|D||A| \log |A|)$ and, because the total credit paid $S$ cannot be greater, it is also the time complexity of all splittings.

It remains to show that the credit of each component is high enough to pay for its splitting. To this end, it is enough to show that every component's credit is at least $s \log_2 s$, where $s$ is the size of the component. When splitting, the split component's credit is high enough to pay $s_2$ for the splitting and still it is able to give $s_1 \log_2 s_1$ credits to the bigger component and $s_2 \log_2 s_2$ credits to the smaller one, because we have:

$$s_1 \log_2 s_1 + s_2 \log_2 s_2 + s_2 = s_1 \log_2 s_1 + s_2(1 + \log_2 s_2) \leq$$

$$\leq s_1 \log_2 s + s_2(1 + \log_2 s/2) = s_1 \log_2 s + s_2 \log_2 s = s \log_2 s$$

(the last inequality follows from $s_1 \geq s_2$ and hence $s = s_1 + s_2 \geq 2s_2$).

We have just proved that if a component of size $s$ has at least $s \log_2 s$ credits, it is enough for every split. It remains to show that every component has at least this credit. We prove it by induction. We start with $B_0$ consisting of one component of size 4. Its credit is $4 \log_2(4(c|A|+1))$ and this is of course not less than $4 \log_2 4$. Now let us have a component of size $s$, we assume that its credit is $s \log_2 s$. We need to show that after adding an edge with credit $\log_2(4(c|A|+1))$, the credit will be at least $(s+1) \log_2(s+1)$. To this end, let us compute this difference:

$$\begin{aligned} \Delta &= s \log_2 s + \log_2(4(c|A|+1)) - (s+1) \log_2(s+1) = \\ &= s \log_2 s + 2 + \log_2(c|A|+1) - (s+1) \log_2(s+1). \end{aligned}$$

Lemma 9.4 shows that there is a constant $C > 0$ such that the size of the approximation of the hull is at most $C|A|$ in every iteration of the algorithm. If we put $c = C$ we have $s \leq c|A|$ and we may continue:

$$\Delta \geq s \log_2 s + 2 + \log_2(s+1) - s \log_2(s+1) - \log_2(s+1) =$$

$$= 2 - s(\log_2(s+1) - \log_2 s) = 2 - s\log_2\frac{s+1}{s} = 2 - s\frac{\ln(1+\frac{1}{s})}{\ln 2}.$$

Following the fact $1 + x \leq e^x$, we have

$$\Delta \geq 2 - s\frac{\frac{1}{s}}{\ln 2} = 2 - \frac{1}{\ln 2} = \log_2 4 - \log_2 e$$

which is a positive number and the proof is finished. □

Adding all the steps together, we see that every iteration of the algorithm (not including the time needed to restore the edge-component mapping) could be done in $O(|D|\log(|A|))$ time, and all the remappings in $O(|D||A|\log|A|)$ time.

Let us summarize the results of this analysis:

**Corollary 9.9** *Let $D$ be a finite set of vectors in the plane. The functionally $D$-convex hull of a finite set $A$ of points in the plane can be computed in*

$$O(|A||D|^2\log(|A|))$$

*time. In particular, for a fixed size of the set $D$, the time complexity of the computation is*

$$O(|A|\log(|A|)).$$

**Proof:** According to Lemma 9.6, the algorithm has $O(|D||A|)$ iterations. Because the initial phase takes only $O(|D|\log|D| + |D||A|\log|A|)$ time, the algorithm finishes in

$$O(|D|\log|D| + |D||A|\log|A| + |D||A||D|\log(|A|) + |D||A|\log(|A|))$$

time (including the initial phase). □

# Chapter 10

# The Program

In this chapter, we present the results of our effort to implement the algorithm given in Chapter 8. We start with some implementation notes and then show some pictures from the output of the program.

## 10.1 Notes on Implementation

We have chosen to write the code in C++ because of general availability of compilers of this language.

### 10.1.1 The Early Stages

First, we tried to write the whole implementation of the planar algorithm from scratch. After many failures, which were caused mainly by the subtleties of implementing various geometric algorithms, we had a version that was able to compute the functional $D$-convex hull of a given finite point set $A$ for given finite set of directions $D$. It worked well for general sets, but sometimes failed or even crashed if there were some special configurations, for example two points of the set $A$ on the same $D$-line etc. The data structures for representation of various geometric objects involved were derived from the well known ones. We just simplified them (a lot). This simplified the implementation quite well. On the other hand, the non-standard representation prevented us from porting of the resulting data structure to some visualisation apparatus. For these reasons, we

decided to use some standard, robust and well proven geometric library instead.

### 10.1.2 CGAL

In the end, we have developed a C++ implementation of the planar algorithm, which is quite different from the early versions. The program is based on the Computational Geometry Algorithms Library (CGAL). To describe this library, we use the characterization from CGAL web page [CGA]:

The goal of the CGAL Open Source Project is to provide easy access to efficient and reliable geometric algorithms to users in industry and academia in the form of a C++ library. CGAL is used in various areas needing geometric computation, such as: computer graphics, scientific visualization, computer aided design and modeling, geographic information systems, molecular biology, medical imaging, robotics and motion planning, mesh generation, numerical methods, and others.

The Computational Geometry Algorithms Library offers data structures and algorithms like triangulations, Voronoi diagrams, Boolean operations on polygons and polyhedra, arrangements of curves, mesh algorithms, alpha shapes, convex hull algorithms, operations on polygons, search structures, interpolation, optimization algorithms, and kinetic data structures.

All these data structures and algorithms operate on geometric objects like points and segments, and perform geometric tests on them. These objects and predicates are regrouped in CGAL Kernels.

Finally, the Support Library offers interfaces to other packages, e.g., for visualization, I/O, and number types.

More on the library can be found on its home page.

We see that CGAL is used in really various areas that need geometric computation. This proves not only its general usability, but this also indicates (together with the big list of projects that use CGAL) that the library is already very well proven.

CGAL includes implementations of many geometric algorithms and structures, but we need just a small portion of them. We need some basic point query algorithms for deciding convexity etc. CGAL offers them, of course. We also need an representation of our polyhedra, that is, a structure with a set of particular

algorithms operating on it. We have decided to use the `Nef_2` data structure. This structure is the CGAL implementation of the planar Nef polyhedron. The important thing is that it can work with singular cases (e.g., a single point being a component of the polyhedron). It allows us to perform the set operations we need (e.g., intersection, interior, union) easily, too. All details of the implementation of planar Nef polyhedra in CGAL can be found in Seel [See01].

We are aware of the fact that this implementation is far from being the most effective one. Nef polyhedron is quite general concept, whereas our polyhedra are very simple. This is also the reason why in the beginning, we did not want to use it and rather, we tried to develop some simple implementation (as we already noted when describing the early stages). Nevertheless, the effectiveness of the implementation was not an important issue, because we wanted to use it for experimental and illustrational purposes only, and using Nef polyhedra, the implementation was quite straightforward and easy. Another advantage of Nef polyhedra is the fact that there is also an implementation of three-dimensional Nef polyhedra in CGAL available. This helped us to perform some experiments with our algorithm related to three-dimensional space with incomparably less effort than would be needed when sticking to our simple but non-standard implementation.

### 10.1.3 Nef Polyhedra

To be entitled to use the Nef polyhedra for our algorithm, we need to show the relationship between Nef polyhedra and $(D, A)$-polyhedra. Fortunately, this is rather straightforward.

**Definition 10.1 (Nef [Nef78])** *A set $P \subseteq \mathbf{R}^d$ is a* Nef polyhedron *if $P$ is the result of a recursive application of set intersection and set complement starting from open half-spaces.*

H. Bieri gave alternative definitions for Nef polyhedra and proved their equivalence. We use the following one:

**Lemma 10.2 (Bieri [Bie95])** *The original Definition 10.1 of Nef polyhedron is equivalent to the following:*
*A set $P \subseteq \mathbf{R}^d$ is a* Nef polyhedron *if there exists a set of hyperplanes $H$ such that $P$ is the union of some cells of the arrangement $\mathcal{A}(H)$.*

Because the closed cell we used for the definition of $(D, A)$-polyhedron is the union of the corresponding open cell of the arrangement $\mathcal{A}(H)$ with all its bounding cells, we see that the $(D, A)$-polyhedron family is part of the Nef polyhedron family:

**Corollary 10.3** *Any $(D, A)$-polyhedron is a Nef polyhedron.*

### 10.1.4 The CGAL Implementation

We started with CGAL version 3.0, but currently we are using the version 3.3. The real arithmetic was implemented using the GMP library (GNU Multiple Precision Arithmetic Library) supplied with CGAL, as well as the standalone one (since newer versions of CGAL do not include GMP any more).

Since version 3.2 (May 2006), the CGAL team has started creating installers for standard platforms (Apple, Windows, rpms), and thus its installation becomes pretty straightforward.

For the visualisation of the algorithm, we adapted the `Nef_2 Demo with CGAL Qt_widget` demo program, which is one of the example programs supplied togeher with the CGAL library. It uses the Qt [Qt] library version 3.3 (currently, the CGAL demos work with Qt 3 and not with Qt 4).

The program was developed and tested on the Linux platform, namely Fedora Core 4 [Fed]. The C++ compiler we used was GNU version 4.0.0. The program was also succesfully compiled and tested on more recent versions, namely Fedora 5 through 8 with compiler GNU 4.1, CGAL 3.3 (installed via rpm), and GMP 4.1.4. But it should be platform independent and thus could be compiled and run on any system with compatible versions of CGAL, Qt and a C++ compiler (and GMP, if needed) installed, although we have not tried any other operating systems than Fedora.

## 10.2 The Output

The following few pictures are some screenshots from the program. The set $A$ is given by the seven points (marked by black squares)

$$\{(10, 12), (12, 14), (14, 12), (12, 10), (6, 6), (8, 7), (9, 5)\}.$$

The set of directions is

$$D = \{(0,1),(1,0),(1,-1)\}.$$

The pictures show some interesting or important iterations of the algorithm. Figure 10.1 is the initial state: we have some functionally $D$-convex $(D,A)$-polygon containing the points from $A$ (cf. Figure 8.1). Figure 10.2 shows the first iteration and Figure 10.3 shows the situation after several iterations of the algorithm, just before the polygon splits in two parts on Figure 10.4. Figure 10.5 shows the final state, i.e., the functionally $D$-convex hull of the set $A$.

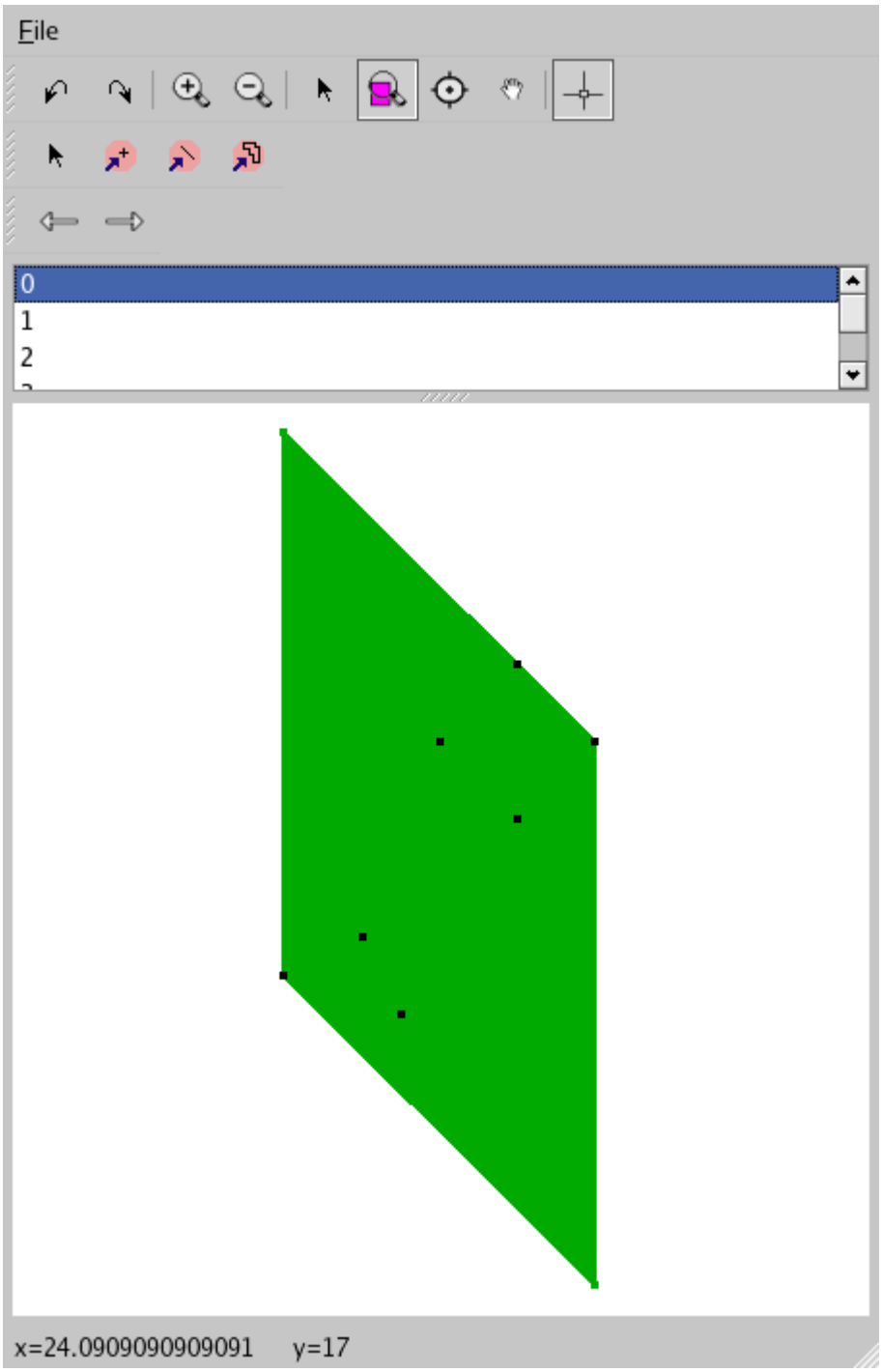



Figure 10.1: Iteration 0. The initial state.

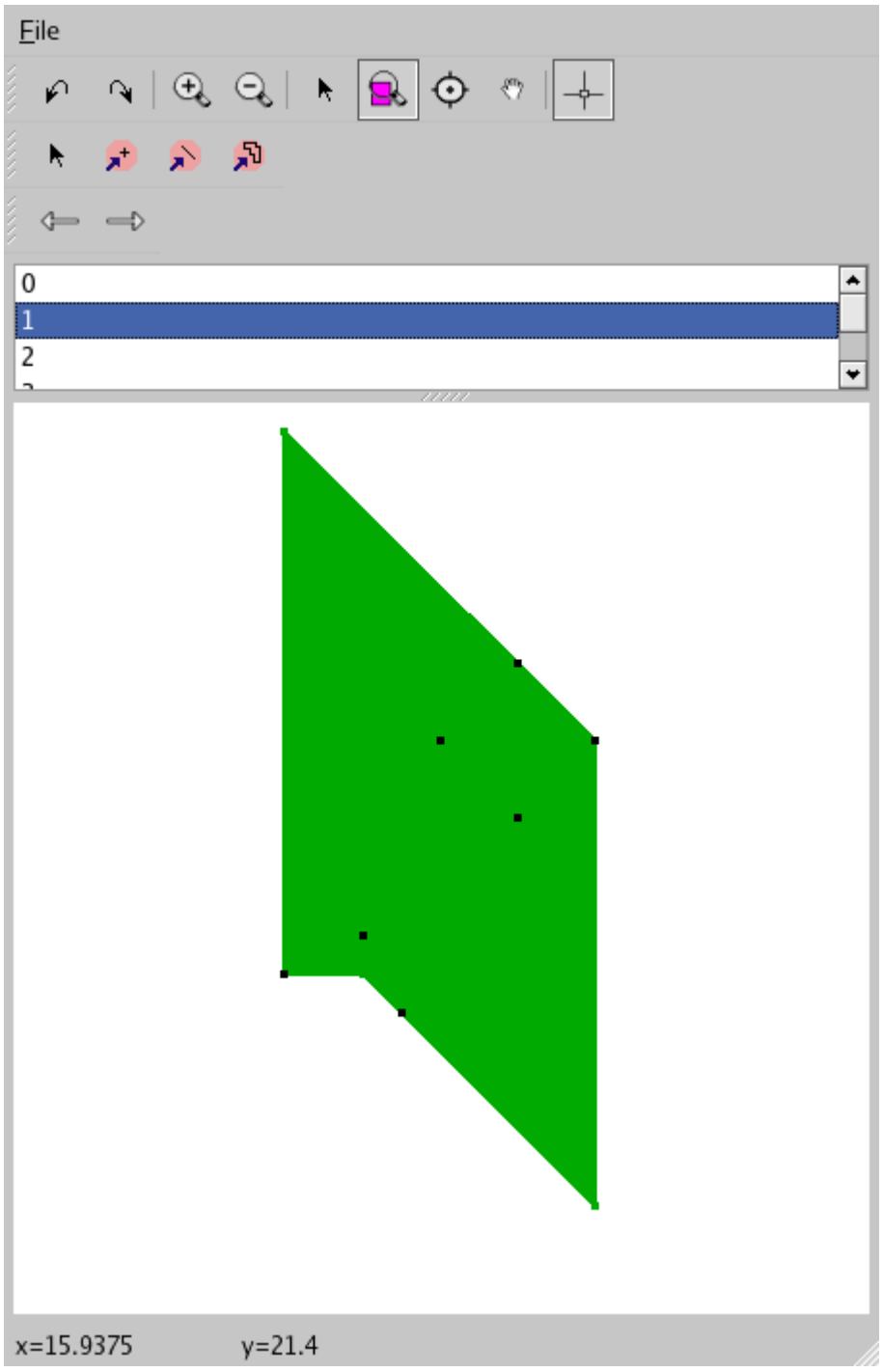



Figure 10.2: Iteration 1. The first iteration.

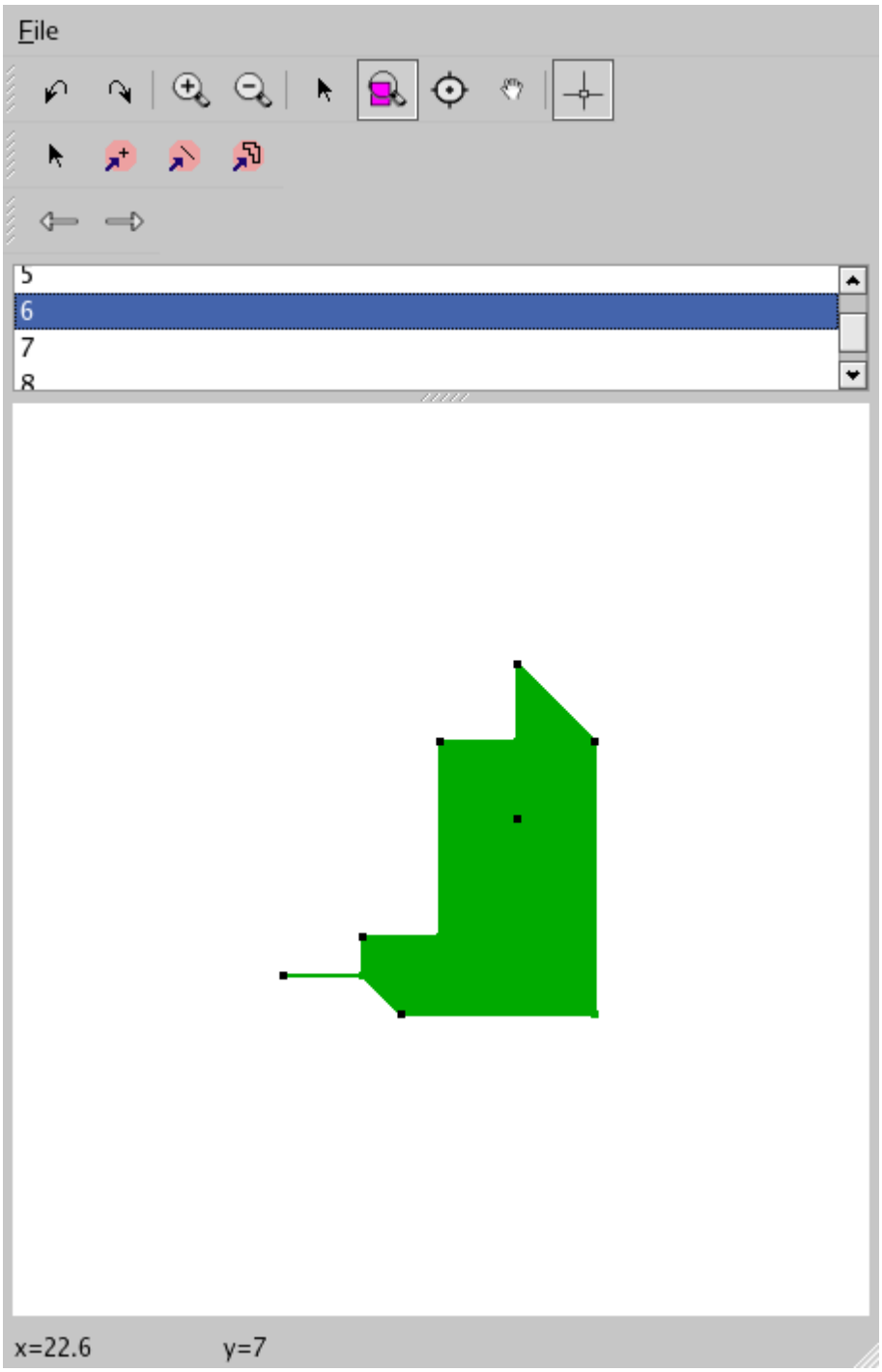



Figure 10.3: Iteration 6. Still in one piece.

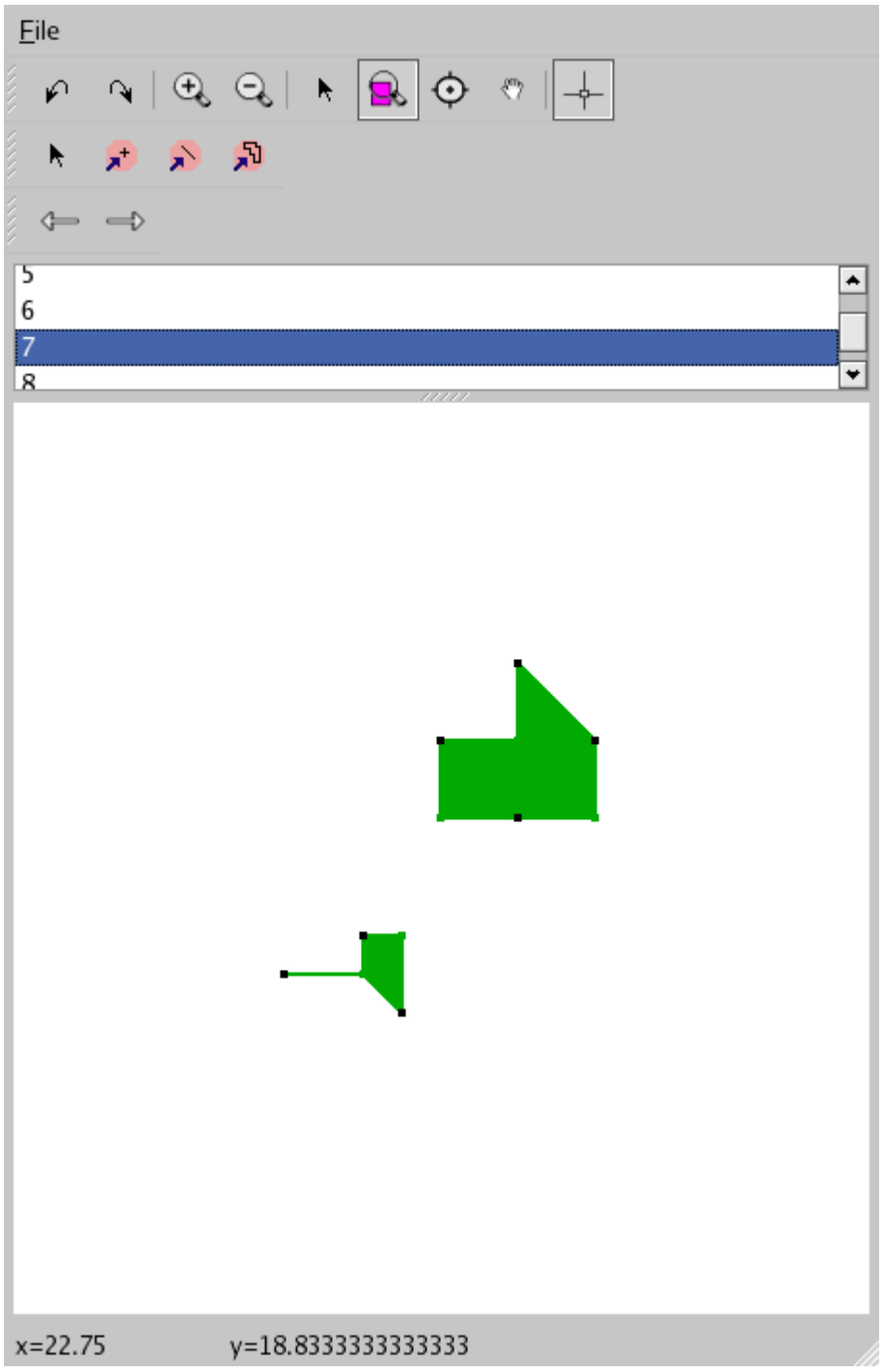



Figure 10.4: Iteration 7. Split in two parts.

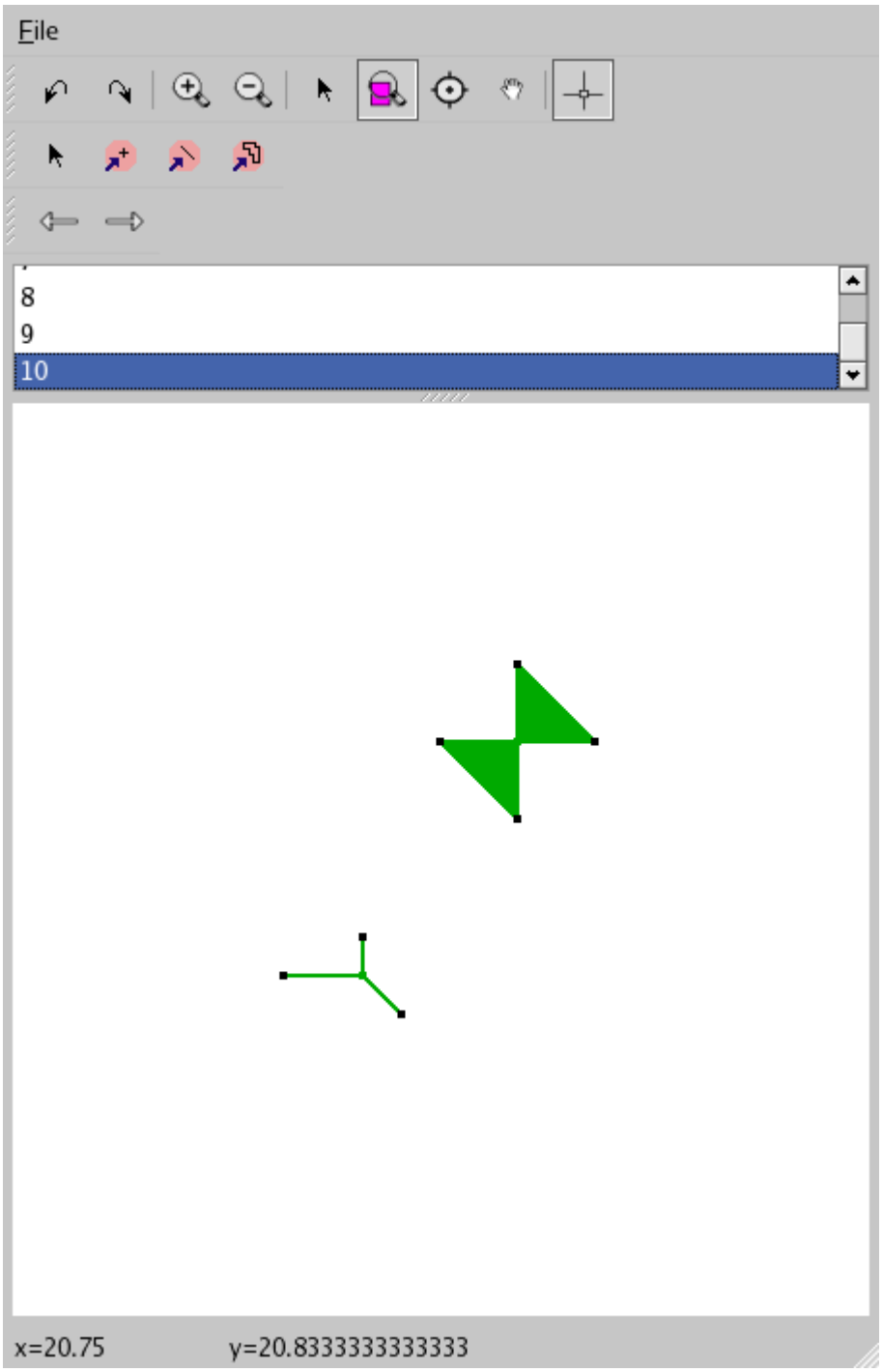



Figure 10.5: Iteration 10. The final state: the functional $D$-convex hull.

# Chapter 11

# Higher Dimension

In this chapter, we discuss the following natural question: Is it possible to extend the planar algorithm easily to a higher dimension? Unfortunately, the answer is negative. In the following paragraphs, we show some reasons for this, and we illustrate them on examples in three-dimensional space.

Many of the lemmas and facts stated in the previous chapters do not hold for arbitrary dimension. Usually, they are true only in the planar case and fail even in three-dimensional space.

As an example, consider the translation of the cutter. Unlike in the plane, the cutting cone can be in such a position from the very beginning that it cannot be moved at all. One such situation for three-dimensional space shows Figure 11.1. Here we have a cone given by four rays emenating from $v$ (the vertex to cut) which cannot be moved because it is stopped by the black points $a_1, a_2$ belonging to $A$. This has kind of the flavour of the tripod — again, we have an extremal point and we are incapable to cut it off. The difference is that in the tripod, the extremal point is not convex, while here we have convex extremal point, but the cutter for it is stuck.

While translating the cone to an extremal position, one also cannot expect that the cone in the final position will still be a $(D, A)$-polyhedron. This would lead to a generalized definition of a "$(D, A)$-cutter". We see one possible solution in establishing a definition which, for given finite sets $A$ and $D$, provides finitely many of these cutters, and at the same time, the cutters will help the algorithm to satisfy the conditions of the Cutter Lemma somehow. Such a definition would solve this problem.

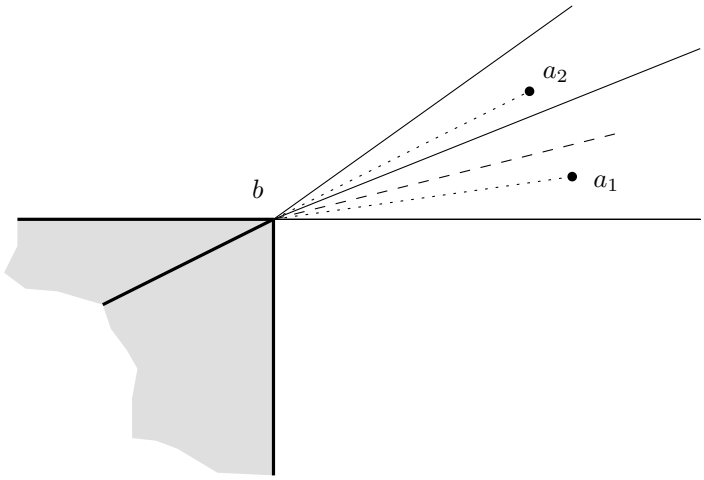



Figure 11.1: An example of a stuck cutter.

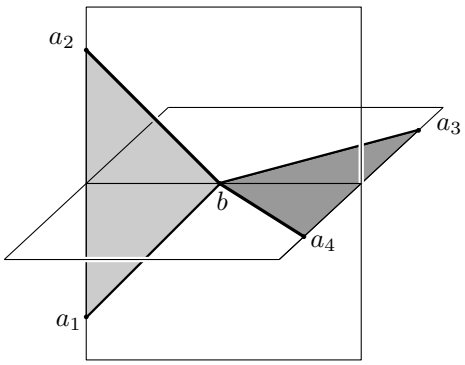



Figure 11.2: Obstructed $D$-extremal point.

Next, we don't have a direct analogy for the (generalized) tripod, or — to be precise — for situations where we have a non-convex $D$-extremal vertex which cannot be cut off. It is clear that if there are three vectors in $D$ that lie in a common plane (and no other vectors from $D$ lie in that plane), there are situations where some three edges parallel to these three vectors form a planar tripod. A more general case is the situation when four (or more) $(D, A)$-lines meet in a common point $b \notin A$ lying in the interior of the convex hull of the points from $A$ defining these lines. Let us have a look at Figure 11.2 ($d = 3$). The set $D$ is given by the four directions which are drawn by the thick segments $ba_1, ba_2, ba_3, ba_4$ and $A = \{a_1, a_2, a_3, a_4\}$. The $(D, A)$-polytope $B$ is the union of the four thick segments. It has $b$ as a $D$-extremal point, but with our algorithm directly extended to three dimensions, we cannot cut it off. At first sight, this might be perhaps considered as a direct three dimensional analogy for the planar

tripod. But there is no proof showing whether the point $b$ should be cut off or not.

It is not sure whether these are the only types of "obstructed $D$-extremal points". There may be other configurations, where the $D$-extremal point is protected from being cut off. These configurations must be found (or, it must be proved, that they do not exist) and it must be made clear, whether the hidden points should be cut off in some way or not. Of course, in higher dimensions yet other configurations may arise, that we do not see in three-dimensional space.

# Acknowledgements

The author would like to thank Prof. RNDr. Jiří Matoušek, DrSc. for his immense patience and support during creation of this work.

# Bibliography


[AH86] Robert J. Aumann and Sergiu Hart. Bi-convexity and Bi-martingales. *Israel Journal of Mathematics*, 54(2):159–180, 6 1986.

[Bar02] Alexander Barvinok. *A Course in Convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2002.

[Bie95] Hanspeter Bieri. Nef Polyhedra: A Brief Introduction. *Computing. Supplementum*, 10:43–60, 1995.

[Car07] Constantin Carathéodory. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Math. Ann.*, 64:95–115, 1907.

[CGA] Computational Geometry Algorithms Library. World Wide Web site `http://www.cgal.org/`.

[CT02] Pierre Cardaliaguet and Rabah Tahraoui. Equivalence between Rank-one Convexity and Polyconvexity for Isotropic Sets of $R^{2\times 2}$ (Parts I and II). *Nonlinear Analysis*, 50(8):1179–1239, 2002.

[dBvKOS00] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry*. Springer, Berlin, 2 2000.

[Ety] Online Etymology Dictionary. Douglas Harper, Historian. Dictionary.com website:
`http://dictionary.reference.com/browse/convex`.

[Fed] Fedora Project. World Wide Web site `http://www.cgal.org/`.

[FM] Vojtěch Franěk and Jiří Matoušek. Computing $D$-convex Hulls in the Plane. *Computational Geometry: Theory and Applications*. to appear.

[FW95] Eugene Fink and Derick Wood. Three-dimensional Strong Convexity and Visibility. In *Proceedings of the Vision Geometry IV Conference*, pages 61–72, 1995.

[FW96a] Eugene Fink and Derick Wood. Fundamentals of Restricted-Orientation Convexity. *Information Sciences*, 92(1–4):175–196, 1996. Also appeared as Technical Report HKUST-CS95-46, Hong Kong University of Science & Technology.

[FW96b] Eugene Fink and Derick Wood. Generalizing Halfspaces. In *Proceedings of the Eighth Canadian Conference on Computational Geometry*, pages 211–216, 1996.

[FW96c] Eugene Fink and Derick Wood. Restricted-Orientation Halfspaces. Technical Report HKUST-CS96-26, Hong Kong University of Science & Technology, 1996. Also available in Proceedings of the Vision Geometry V Conference, pages 24-33, 1996.

[FW98a] Eugene Fink and Derick Wood. Generalized Halfspaces in Restricted-Orientation Convexity. *Journal of Geometry*, 62:99–120, 1998. An extended version is available as Technical Report HKUST-CS95-45, Department of Computer Science, Hong Kong University of Science & Technology, 1995.

[FW98b] Eugene Fink and Derick Wood. Strong Restricted-Orientation Convexity. *Geometriae Dedicata*, 69:25–51, 1998. Also appeared as Technical Report HKUST-CS95-41, Department of Computer Science, Hong Kong University of Science & Technology, 1995. An extended version is available as Technical Report CMU-CS-95-154, School of Computer Science Carnegie Mellon University, 1995.

[FW03] Eugene Fink and Derick Wood. Planar Strong Visibility. *International Journal of Computational Geometry & Applications*, 13(2):173–187, 2003.

[FW04] Eugene Fink and Derick Wood. *Restricted-Orientation Convexity*. Springer-Verlag, Berlin, Germany, 2004.

[Kir99] Bernd Kirchheim. On the Geometry of Rank-one Convex Hulls. Manuscript, Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany, 1999.

[KM40] Mark Krein and David Milman. On Extreme Points of Regular Convex Sets. *Studia Mathematica*, 9:133–138, 1940.

[KMŠ02] Bernd Kirchheim, Stefan Müller, and Vladimír Šverák. Studying Nonlinear PDE by Geometry in Matrix Space. Preprint 49, Max-Planck-Institut für Mathematik in den Naturwissenchaften, Leipzig, Germany, 2002.

[Let99] Boris Letocha. Směrová konvexita. Master's thesis, Department of Applied Mathematics, Charles University, Prague, Czech Republic, 1999. Written in Czech language.

[Mat01] Jiří Matoušek. On Directional Convexity. *Discrete & Computational Geometry*, 25(3):389–403, 2001.

[Mat02] Jiří Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002.

[Mor52] Charles Bradfield Morrey, Jr. Quasi-convexity and the Lower Semicontinuity of Multiple Integrals. *Pacific Journal of Mathematics*, 2(1):25–53, 1952.

[Mor66] Charles Bradfield Morrey, Jr. *Multiple Integrals in the Calculus of Variations*. Springer-Verlag, Berlin, 1966.

[MP98] Jiří Matoušek and Petr Plecháč. On Functional Separately Convex Hulls. *Discrete & Computational Geometry*, 19(1):105–130, 1998.

[MR03] Eva Matoušková and Simeon Reich. The Hundal Example Revisited. *Journal of Nonlinear and Convex Analysis*, 4(3):411–428, 2003.

[MŠ99] Stefan Müller and Vladimír Šverák. Convex Integration with Constraints and Applications to Phase Transitions and Partial Differential Equations. Preprint 28, Max-Planck-Institut für Mathematik in den Naturwissenchaften, Leipzig, Germany, 1999.

[Mül98] Stefan Müller. Variational Models for Microstructure and Phase Transitions. Lecture Notes 2, Max-Planck-Institut für Mathematik in den Naturwissenchaften, Leipzig, 1998.

[Nef78] Walter Nef. *Beiträge zur Theorie der Polyeder mit Anwendungen in der Computergraphik*. Herbert Lang & Cie AG, Bern, 1978.

[NLLW83] Tina Man Fong Nicholl, D. T. Lee, Y. Z. Liao, and C. K. Wong. On the X-Y Convex Hull of a Set of X-Y Polygons. *BIT Numerical Mathematics*, 23(4):456–471, December 1983.

[NM91] Vincenzo Nesi and Graeme Walter Milton. Polycrystalline Configurations that Maximize Electrical Resistivity. *Journal of the Mechanics and Physics of Solids*, 39(4):525–542, 1991.

[PS85] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Springer, 1985.

[PŠ98] Pablo Pedregal and Vladimír Šverák. A Note on Quasiconvexity and Rank-one Convexity for 2×2 Matrices. *Journal of Convex Analysis*, 5(1):107–117, 1998.

[Qt] Qt: Cross-Platform Rich Client Development Framework. World Wide Web site `http://trolltech.org/products/qt/homepage`.

[Roy88] Halsey L. Royden. *Real Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, third edition, 1988.

[Rud76] Walter Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, New York, third edition, 1976.

[Sch74] Vladimir Scheffer. *Regularity and Irregularity of Solutions to Nonlinear Second-Order Elliptic Systems of Partial Differential Equations and Inequalities*. Ph.D. dissertation, Princeton University, 1974.

[See01] Michael Seel. Implementation of Planar Nef Polyhedra. Research Report MPI-I-2001-1-003, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany, August 2001.

[SRW91] Sven Schuierer, Gregory J. E. Rawlins, and Derick Wood. A Generalization of Staircase Visibility. In Hanspeter Bieri, editor, *CG '91: Proceedings of the International Workshop on Computational Geometry — Methods, Algorithms and Applications*, pages 277–287, London, UK, 3 1991. Springer-Verlag.

[Šve92] Vladimír Šverák. Rank-one Convexity Does not Imply Quasiconvexity. In *Proceedings of the Royal Society of Edinburgh, Section A. Mathematics*, volume 120, pages 185–189, 1992.

[SW91a] Sven Schuierer and Derick Wood. Restricted-orientation Visibility. Technical Report 40, Institut für Informatik, Universität Freiburg, 1 1991.

[SW91b] Sven Schuierer and Derick Wood. Visibility in Semi-convex Spaces. Technical Report 37, Institut für Informatik, Universität Freiburg, 1 1991.

[SW97] Sven Schuierer and Derick Wood. Visibility in Semi-Convex Spaces. *Journal of Geometry*, 60:160–187, 1997. Also appeared as Technical Report HKUST-CS95-39, Hong Kong University of Science & Technology.

[Tar93a] Enrico Casadio Tarabusi. An Algebraic Characterization of Quasiconvex Functions. *Ricerche di Matematica*, 42:11–24, 1993.

[Tar93b] Luc Tartar. Some Remarks on Separately Convex Functions. In D. Kinderlehrer, R. James, M. Luskin, and J. L. Ericksen, editors, *Microstructure and Phase Transition*, volume 54 of *The IMA Volumes in Mathematics and Its Applications*, pages 191–204. Springer-Verlag, New York, 1993.

[WWW85] Peter Widmayer, Ying-Fung Wu, and C. K. Wong. Distance Problems in Computational Geometry with Fixed Orientations. In *SCG '85: Proceedings of the First Annual Symposium on Computational Geometry*, pages 186–195, New York, NY, USA, 1985. ACM.