Blockcourse on High-Dimensional Computational Geometry Spring 2013

Lecture 2: Convex hulls in hi-dimensions — July 23, 2013

Wolfgang Mulzer *Scribe: Vissarion Fisikopoulos and Elena Khramtcova*

1 The asymptotic upper bound theorem

In the first part of this lecture we would like to answer the following question. Can the number of faces of a polytope be much larger than the number of faces of the cyclic polytope? The answer is negative and is given by the theorem below. First, given a simplicial polytope P we define f_j to be the number of j-faces of P, for $j = -1, \dots, d$.

Theorem 1 (Asymptotic upper bound theorem [Sei81]). *Let P be a simplicial polytope with n vertices then*

(i) $\sum_{j=-1}^{d-1} f_j \leq 2^d f_{d-1}$,

(ii) $f_{d-1} \leq 2 f_{\lfloor d/2 \rfloor - 1}$.

That is, $f_{d-1} \leq 2\binom{n}{\lfloor d/2 \rfloor}$ *and (by (i)) the number of faces of P is at most* $2^{d+1}\binom{n}{\lfloor d/2 \rfloor}$.

Proof. (i) Every face is a face of a facet and each facet has 2^d subfaces since it is a $(d-1)$-simplex.

(ii) Consider the dual polytope P^* where f_j^* denotes the number of its j-faces. There is a bijection between the j-faces of P and the $(d-j-1)$ faces of P^*. We will show that $f_0^* \leq 2 f^*_{\lceil d/2 \rceil}$. Suppose w.l.o.g. that all vertices of P^* have distinct x_d-coordinates. Then, orient the edges of P^* according to the x_d-coordinates (e.g. upwards). Each vertex has indegree $\geq \lceil d/2 \rceil$ or outdegree $\geq \lceil d/2 \rceil$. Suppose the latter and let v be a vertex with $\lceil d/2 \rceil$ outgoing edges. Then these edges, namely $e_1, e_2, \dots, e_{\lceil d/2 \rceil}$, define a $\lceil d/2 \rceil$-face of P^*. To see this consider the following argument; v is a vertex of P^* that corresponds to a dual face f of P, which is a $(d-1)$-simplex with vertex set $p_f = \{p_1, p_2, \dots, p_d\}$. The outgoing edges of v, $e_1, e_2, \dots, e_{\lceil d/2 \rceil}$, correspond to the dual (incident to f) ridges $e_1^*, e_2^*, \dots, e^*_{\lceil d/2 \rceil}$. We can write $e_i^* = conv(p_f \{p_i\})$ for each $i \in \{1, 2, \dots, \lceil d/2 \rceil\}$. Then, the face $g := conv(p_{\lceil d/2 \rceil + 1}, p_{\lceil d/2 \rceil + 2}, \dots, p_d)$ is contained in each e_i^* and thus its dual $\lceil d/2 \rceil$-face g^* is spanned by $e_1, e_2, \dots, e_{\lceil d/2 \rceil}$. Moreover, v is the vertex of g^* with the smallest x_d-coordinate. Thus, the number of vertices with outdegree $\geq \lceil d/2 \rceil$ is at most $f^*_{\lceil d/2 \rceil}$. Similarly, the number of vertices with indegree $\geq \lceil d/2 \rceil$ is at most $f^*_{\lceil d/2 \rceil}$.

□

Remark 2.

- *By a "perturbation argument" the theorem can be extended to non-simplicial polytopes with n vertices.*
- *The full upper bound theorem is due to McMullen [McM70] and states that cyclic polytopes maximizes f_j for each j among all polytopes with n vertices.*

2 Algorithms for convex hulls

Given is a set of points $P = \{p_1, p_2, \dots, p_n\} \subseteq \mathbb{R}^d$ in general position. The problem is to find the face lattice of $conv(P)$.

For the $d = 2$ case a lot of algorithms exist such as the Graham-scan, the randomized incremental algorithm, the Jarvis march, the divide and conquer and the quickhull. The first three are generalizable in higher dimensions. The rest of this lecture is devoted to the first two. We will not cover the third known as the gift-wrapping algorithm.

2.1 Seidel's algorithm

We start by the generalization of Graham-scan algorithm which is due to Seidel [Sei95]. Recall that in two dimensions Graham-scan sort the points by their x-coordinate, start with the first three points that form a triangle and incrementally construct the convex hull by inserting the rest of the points from left to the right.

This is can be generalized in d-dimensions as follows.

1. Sort P according to x_1-coordinate and let $p_1, p_2, \dots, p_n$ be now the sorted points. Also suppose that are all distinct.

2. Successively compute the face lattice of $conv(p_1, p_2, \dots, p_{d+1})$, $conv(p_1, p_2, \dots, p_{d+2})$, $\dots$, $conv(p_1,\ p_2, \dots, p_n)$.

   (a) Finding $conv(p_1, p_2, \dots, p_{d+1})$ is easy; it is a d-dimensional simplex.

   (b) In order to get from $conv(p_1, p_2, \dots, p_i)$ to $conv(p_1, p_2, \dots, p_{i+1})$ we first check all facets incident to p_i in order to find the one in *conflict* with p_{i+1}, i.e. check whether p_{i+1} and $conv(p_1, p_2, \dots, p_i)$ are in different sides of the hyperplane spanned by the facet under examination.

   (c) Perform a Depth-First-Search (DFS) among the facets to find all facets that need to be deleted. Observe that this is a DFS on a simply connected set.

   (d) Identify the *boundary* of the conflict region, i.e. all faces (ridges and lower dimensional) incident to at least one conflict facet and one non conflict facet.

   (e) Delete all conflict facets and faces that are contained in conflict facets that are not on the boundary

   (f) For every j-face on the boundary create a $(j+1)$-face with p_{i+1} as an additional vertex and update the current face lattice with the new faces.

Analysis. The running time of each step can be summarized as follows, where the steps 2(b)-2(f) are referring to round $i + 1$.

1. $O(n \log n)$

2. (a) $O(2^d)$

   (b) O(# of facets created in round i)

   (c) O(# of conflict facets = # of facets to be deleted) *(because the polytope is simplicial)*

   (d) O(# of facets to be created)

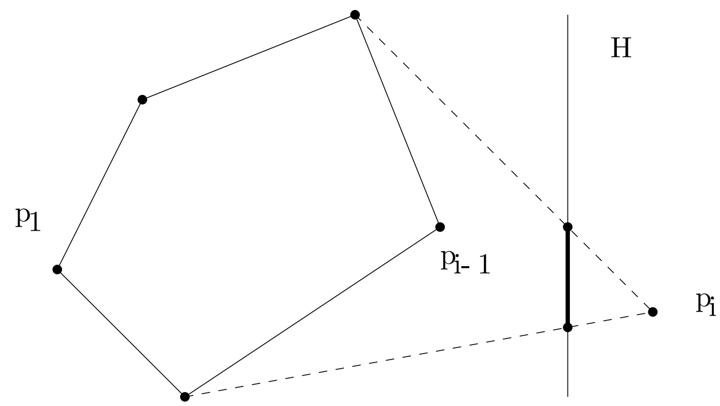


Figure 1: The vertex figure of p_i.

(e) O(# of facets deleted)

(f) O(# of facets created)

To conclude, the total running time of the algorithm is $O(n \log n + \sum_{i=d+2}^{n} \#$ faces created in round $i)$. It remains to count how many faces are created in round i. First note that all new faces are incident to p_i. Then take a hyperplane H separating p_i from $conv(p_1, p_2, \ldots, p_{i-1})$. Now, $H \cap conv(p_1, p_2, \ldots, p_{i-1})$ is a $(d-1)$-dimensional polytope whose i-faces correspond to $(i+1)$-faces incident to p_i in $conv(p_1, p_2, \ldots, p_{i-1})$. This polytope is called the *vertex figure* of p_i and has $\leq i-1$ vertices (Fig. 1). It follows that the number of its facets is $O((i-1)^{\lfloor \frac{d-1}{2} \rfloor} 2^d)$. Thus, the total running time (supposed d is constant but large) is

$$O(n \log n + \sum_{i=1}^{n} i^{\lfloor \frac{d-1}{2} \rfloor}) = O(n \log n + n^{\lfloor \frac{d-1}{2} \rfloor + 1}) = O(n \log n + n^{\lfloor \frac{d+1}{2} \rfloor}).$$

For d even this becomes $O(n \log n + n^{\lfloor \frac{d}{2} \rfloor})$ which is optimal and for d odd is $O(n \log n + n^{\lfloor \frac{d}{2} \rfloor + 1})$ which is not optimal.

3 Probabilistic Divide and Conquer algorithm

Seidel's algorithm for the construction of a convex hull is optimal (in the worst case, not to be confused with *output-sensitive* algorithms) for all even dimensions. Now we present another algorithm with expected optimal running time $O(n^{\lfloor \frac{d}{2} \rfloor})$ for any d. The algorithm is due to Clarkson and Shor [CS89], we follow the presentation by Chazelle [Cha02].

Problem.
Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of d-dimensional points,
Find $conv(P)$.

We are passing to the dual problem: for each point $p_i \in P$ we pick the dual hyperplane h_i and we wish to compute an intersection of the halfspaces bounded by these hyperplanes. Formally:

(Dual) Problem.
Given a set $H = \{h_1, h_2, \dots, h_n\}$ of dD hyperplanes,
Find $H^\cap := \bigcap_{i=1}^{n} h_i^-$, where h_i^- is a halfspace bounded by h_i and containing 0 (the origin).

We wish to output the result in the convenient form, namely, not only as a face lattice, but we wish to compute the *canonical triangulation* of $H^\cap$.The canonical triangulation subdivides all faces of $H^\cap$ into simplices, such that the resulting structure is well-behaved. Namely, it is a simplicial complex.

Suppose $H^\cap$ is bounded (we always can add a large bounding box). We recursively define the canonical triangulation of $H^\cap$:

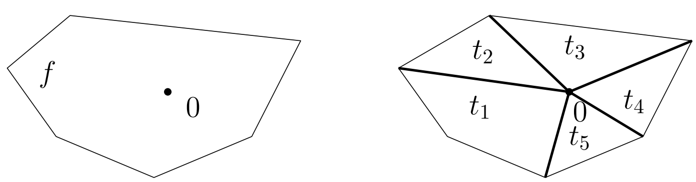
Definition 3 (Canonical/Bottom-vertex triangulation of $H^\cap$)**.**
All $0-$ *and* $1-$ *faces are already triangulated, as they are simplices.*
For $j = 2, \dots, (d-1)$*:*
Suppose all $(j-1)-$*faces have been triangulated.*
Let f *be a* $j-$*face.*

- *Take the* bottom *vertex* v *of* f *(the one with the lexicographically smallest coordinate vector).*
- *For each simplex* δ *in the triangulation of each* $(j-1)-$*face which is a subface of* f*, and such that* v *is not a vertex of* δ*, lift* δ *to a simplex triangulating* f *by adding* v *as a vertex.*

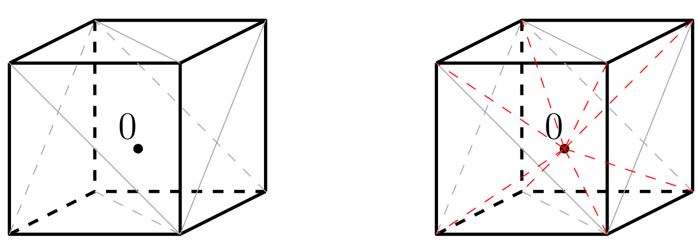
For a single $d-$*face, which is* $H^\cap$ *itself, perform the same, but pick the origin* 0 *instead of the bottom vertex* v*.*

Examples.

1. For $2-$face f (the 2D polygon itself) we added five edges (shown in bold) and five triangles $t_1, \dots, t_5$.



2. For a cube c, suppose all its $2-$faces are triangulated (see the left figure). For its only $3-$face we added twelve tetrahedra (their new boundary edges are shown in red dashed lines).



Fact 4. *Given an $H-$polytope $\mathcal{P}$, defined by n hyperplanes, the canonical triangulation of $\mathcal{P}$ has $O(n^{\lfloor \frac{d}{2} \rfloor})$ simplices and can be computed in time $O(n^{\lfloor \frac{d}{2} \rfloor})$.*

Algorithm (Probabilistic Divide and Conquer) to compute $H^{\cap}$.

1. Compute a chain of subsets of H: $H = H_0 \supseteq H_1 \supseteq H_2 \supseteq \cdots \supseteq H_K$, such that each $h \in H_i$ is contained in H_{i+1} with probability $\frac{1}{2}$.

   Observation 5.

   - *For each $h \in H$, $Pr[h \in H_i] = \frac{1}{2^i}$*
   - *With high probability $K = O(\log n)$.*

2. Successively compute canonical triangulations $\mathcal{T}(H_K^{\cap}), \mathcal{T}(H_{K-1}^{\cap}), \ldots, \mathcal{T}(H_0^{\cap}) = \mathcal{T}(H^{\cap})$

   (a) $\mathcal{T}(H_K^{\cap})$ is found easily (by brute force) since $|H_K| = O(1)$.

   (b) For each simplex $\sigma \in \mathcal{T}(H_K^{\cap})$ find the conflict list $H_\sigma = \{h \in H | \sigma \cap h \neq \emptyset\}$ by testing σ against each $h \in H$.

   (c) Going from $\mathcal{T}(H_i^{\cap})$ to $\mathcal{T}(H_{i-1}^{\cap})$:

      (i) For each simplex $\sigma \in \mathcal{T}(H_i^{\cap})$ compute $S_\sigma = \{h \in H_{i-1} | h \cap \sigma\}$ and the intersection $h \cap \sigma$ (restricted to σ) with each $h \in S_\sigma$. This is done by brute force, checking all hyperplanes from H_σ. Update the conflict lists by checking all new simplices with all hyperplanes in H_σ.

      (ii) Walk along the boundaries of old simplices and merge new simplices (and their conflict lists) touching along these boundaries to obtain $\mathcal{T}(H_{i-1}^{\cap})$.

      (iii) Compute the canonical triangulation of each new simplex.

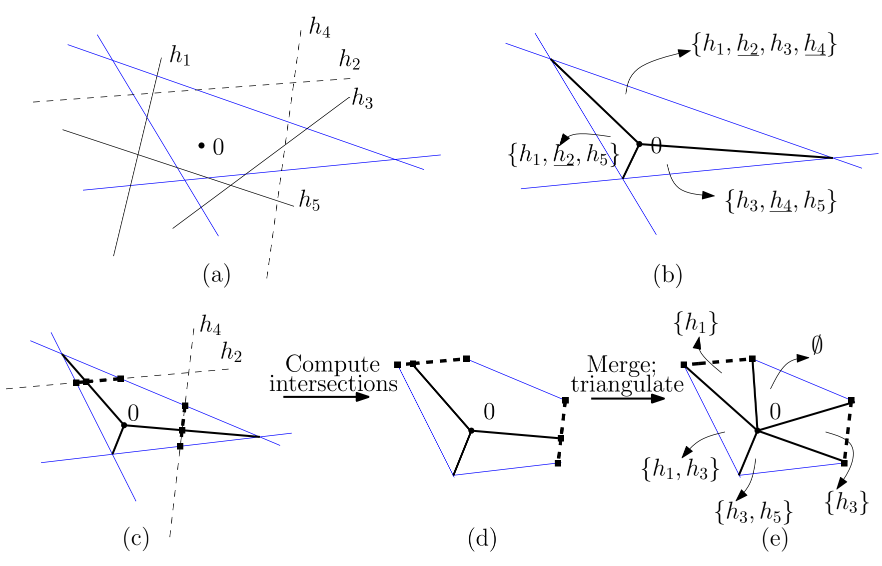


Figure 2: Execution of the algorithm: H_K is three blue lines, H_{K-1} is two dashed lines (fig. (a)). $\mathcal{T}(H_K^{\cap})$ (only the 2−face) with conflict lists (fig. (b)). Getting $\mathcal{T}(H_{K-1}^{\cap})$ from $\mathcal{T}(H_K^{\cap})$ (fig. (c)-(d)).

Lemma 6. *The expected work performed in round i of the algorithm is* $O\left(\left(\frac{n}{2^i}\right)^{\lfloor\frac{d}{2}\rfloor} + n\left(\frac{n}{2^i}\right)^{\lfloor\frac{d}{2}\rfloor-1}\right)$.

Note that the total expected running time of the algorithm is then $O\left(\left(\frac{n}{2^i}\right)^{\lfloor\frac{d}{2}\rfloor} + n\log n\right)$, since steps 1 and 2(a) take constant time in n, and step 3 takes linear time.

Proof. Define the set of all possible simplices: $\mathcal{X} = \{\sigma - simplex | \exists F \subseteq H : \sigma \in \mathcal{T}(F^{\cap})\}$.

Fact 7. *For each $\sigma \in \mathcal{X}$ exists a* trigger set $t_\sigma \subseteq H, |t_\sigma| = O(d^2)$, *such that for any $F \subseteq H$ the simplex $\sigma \in \mathcal{T}(F^{\cap})$ if and only if $t_\sigma \subseteq F$ and $H_\sigma \cap F = \emptyset$.*

Proof(Fact 7). Each simplex has $d+1$ vertices, each of which is an intersection of d hyperplanes. □

For any $\sigma \in \mathcal{X}$ let $s_\sigma = |H_{i-1} \cap \sigma|$, $n_\sigma = |H \cap \sigma|$. The expected work in round i is proportional to

$$E\left[\sum_{\sigma \in \mathcal{T}(H_i^{\cap})} \left(s_\sigma^{d+1} + n_\sigma s_\sigma^{\lfloor\frac{d}{2}\rfloor}\right)\right]. \tag{1}$$

The first term stands for computing the intersections inside the simplex (which is done by brute force) and for the merging step, the second term bounds the cost of updating the conflicts: we check each new simplex, intersecting σ against all the hyperplanes in H_σ.

Expression (1) (by linearity of expectation, and by introducing the indicator variable) equals

$$\sum_{\sigma \in \mathcal{X}} Pr\left[\sigma \in \mathcal{T}(H_i^{\cap})\right]\left(E\left[s_\sigma^{d+1} | \sigma \in \mathcal{T}\left(H_i^{\cap}\right)\right] + n_\sigma\left(E\left[s_\sigma^{\lfloor\frac{d}{2}\rfloor} | \sigma \in \mathcal{T}\left(H_i^{\cap}\right)\right]\right)\right). \tag{2}$$

Claim 8.

- $E\left[s_\sigma^{d+1} | \sigma \in \mathcal{T}\left(H_i^{\cap}\right)\right] = O\left(\left(\frac{n_\sigma}{2^i}\right)^{d+1}\right)$.
- $E\left[s_\sigma^{\lfloor\frac{d}{2}\rfloor} | \sigma \in \mathcal{T}\left(H_i^{\cap}\right)\right] = O\left(\left(\frac{n_\sigma}{2^i}\right)^{\lfloor\frac{d}{2}\rfloor}\right)$.

Proof(Claim 8). Since n_σ is the number of hyperplanes in H that intersect σ, and since H_{i-1} contains each hyperplane from H with probability $\frac{1}{2^{i-1}}$, $E\left[s_\sigma\right] = \frac{n_\sigma}{2^{i-1}}$. By (homework) Exercise 2.2, the claim follows. □

Expression (2) by the Claim 8 is proportional to

$$\sum_{\sigma \in \mathcal{X}} Pr\left[\sigma \in \mathcal{T}(H_i^{\cap})\right]\left(\left(\frac{n_\sigma}{2^i}\right)^{d+1} + n_\sigma\left(\frac{n_\sigma}{2^i}\right)^{\lfloor\frac{d}{2}\rfloor}\right) \leq$$

$$\sum_{t=1}^{\infty} \sum_{\substack{\sigma \in \mathcal{X} \\ (t-1)2^i \leq n_\sigma \leq t2^i}} Pr\left[\sigma \in \mathcal{T}(H_i^{\cap})\right]\left(t^{d+1} + 2^i t^{\lfloor\frac{d}{2}\rfloor+1}\right) \leq$$

$$\sum_{t=1}^{\infty} E\left[\#\text{ of simplices } \sigma \in \mathcal{T}(H_i^{\cap}) : n_\sigma \geq (t-1)2^i\right]\left(t^{d+1} + 2^i t^{\lfloor\frac{d}{2}\rfloor+1}\right) \leq$$

(by Chazelle-Friedman bound, proportional to)

$$\sum_{t=1}^{\infty} \left(\frac{n}{2^i}\right)^{\lfloor \frac{d}{2} \rfloor} t^{d^2} 2^{-t} \left(t^{d+1} + 2^i t^{\lfloor \frac{d}{2} \rfloor + 1}\right) =$$

$$\left(\frac{n}{2^i}\right)^{\lfloor \frac{d}{2} \rfloor} \sum_{t=1}^{\infty} t^{O(d^2)} 2^{-t+1} + \left(\frac{n}{2^i}\right)^{\lfloor \frac{d}{2} \rfloor} 2^i \sum_{t=1}^{\infty} t^{O(d^2)} 2^{-t+1} =$$

$$O\left(\left(\frac{n}{2^i}\right)^{\lfloor \frac{d}{2} \rfloor} + n\left(\frac{n}{2^i}\right)^{\lfloor \frac{d}{2} \rfloor - 1}\right).$$

□

References

[Cha02] B. Chazelle. The discrepancy method: randomness and complexity. *Cambridge University Press*, 2002.

[CS89] K. Clarkson, P. Shor. Application of random sampling in computational geometry, II *Disc. Comput. Geom, 4:387–421*, 1989.

[McM70] P. McMullen. The maximum number of faces of a convex polytope. *Mathematika, XVII:179-184*, 1970.

[Sei81] Raimund Seidel. A convex hull algorithm optimal for point sets in even dimensions. *Master's thesis, University of British Columbia, Dept. of Computer Science, Report 81-14*, 1981.

[Sei95] Raimund Seidel. The upper bound theorem for polytopes: an easy proof of its asymptotic version. *Comput. Geom.*, 5:115–116, 1995.