Computational Geometry

Convexity and convex hull algorithms in general dimensions

Vissarion Fisikopoulos

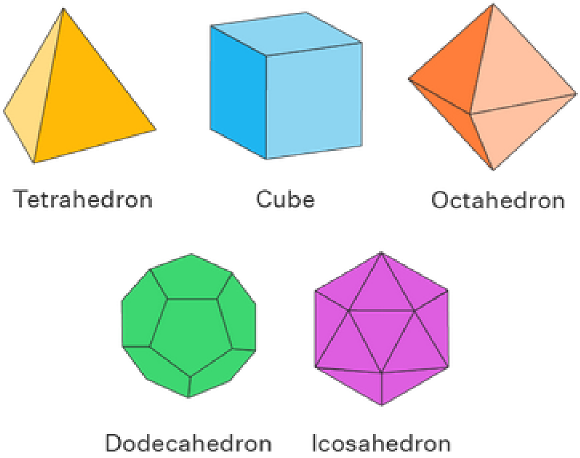
Department of Informatics & Telecommunications
National & Kapodistrian University of Athens

Spring 2025

Contents

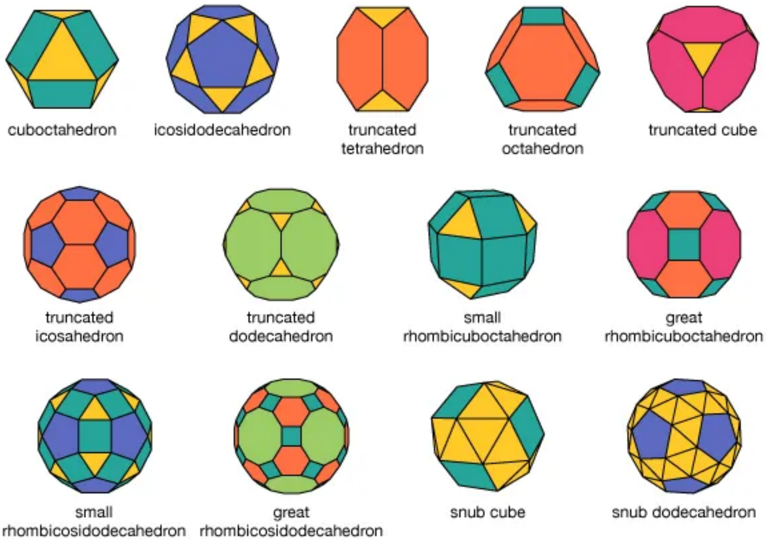
Platonic solids

one type of regular polygon

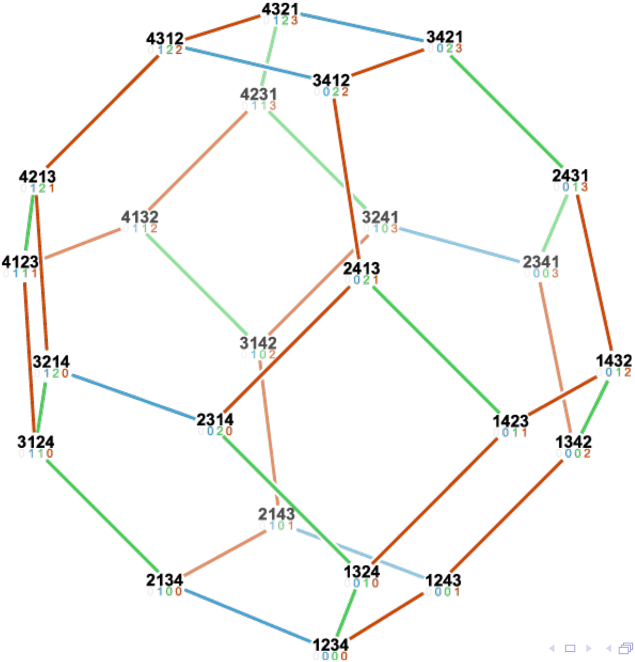


Archimedian solids

different types of regular polygons

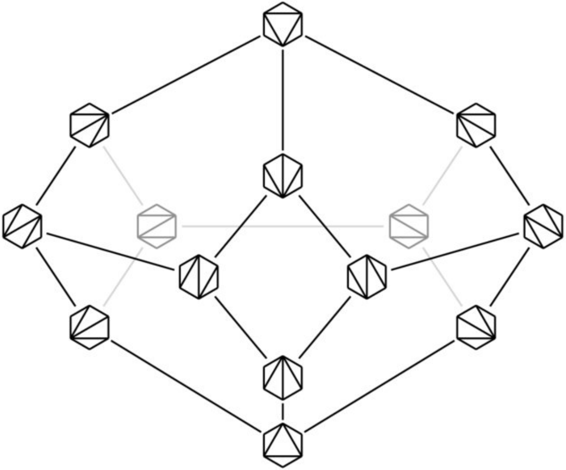


Permutahedron

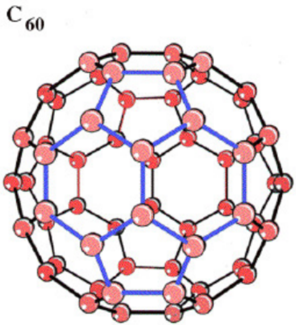


Associahedron

triangulations of convex polygon

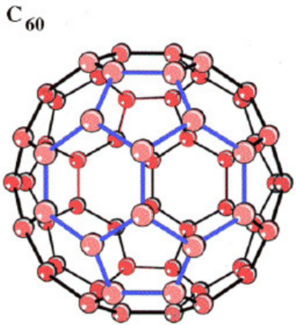


Fullerene



- A convex polyhedron with 60 vertices and 32 faces (12 pentagons and 20 hexagons).
- Nobel Prize in Chemistry, 1996.
- How many edges does it have?

Fullerene



- A convex polyhedron with 60 vertices and 32 faces (12 pentagons and 20 hexagons).
- Nobel Prize in Chemistry, 1996.
- How many edges does it have? (Hint: euler: $v - e + f = 2$)

Contents

Hyperplanes as Point Sets

Algebraic Definition

A hyperplane is defined as the set of points satisfying a linear equation:

$$f(x_1, \ldots, x_d) = k_1 x_1 + \cdots + k_d x_d + k_0 = 0, \quad k_i \in \mathbb{R}. \tag{1}$$

Examples:

- Line: $\{(x_1, x_2) \mid 2x_1 - x_2 = 3\} \subset \mathbb{R}^2$
- Plane: $\{(x_1, x_2, x_3) \mid x_1 + 2x_2 - x_3 = -1\} \subset \mathbb{R}^3$
- Hyperplane: $\{(x_1, x_2, x_3, x_4) \mid x_1 + 2x_2 - x_3 + 5x_4 = 1\} \subset \mathbb{R}^4$

Hyperplanes as Point Sets

Gradient Representation

Equivalently, a hyperplane can be described using the normal vector v and distance $k/\|v\|$ from the origin:

$$f = v \cdot (x_1, \ldots, x_d) + k_0. \tag{2}$$

Properties:

- If $k_0 = 0$, the hyperplane passes through the origin.
- Example: $v = (2, -1), k_0 = -3 \Rightarrow f = 2x_1 - x_2 - 3$
- For $f(x_1, \ldots, x_d) = k_1 x_1 + \cdots + k_d x_d + k_0$:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_d} \right) = (k_1, \ldots, k_d) = v$$

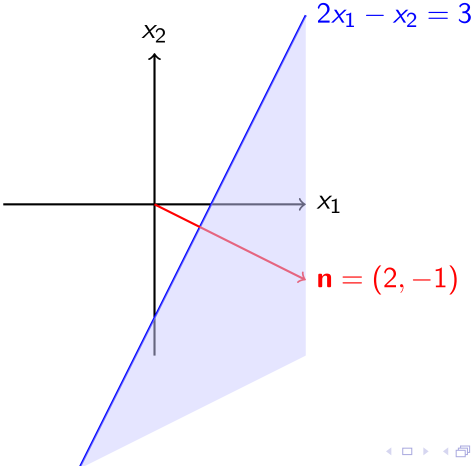
Halfspaces

A half-space is one side of a given hyperplane with dimension d.

$$\{(x_1, \ldots, x_d) \mid f(x_1, \ldots, x_d) \gtrless 0\}. \tag{3}$$

Examples:

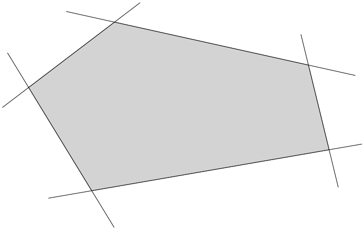
- Half-plane: $\{(x_1, x_2) \mid 2x_1 - x_2 > 3\} \subset \mathbb{R}^2$



Convex Polytope (Definition 1)

as Intersection of Half-Spaces

- A convex polytope (polyhedron) is the intersection of a finite number of half-spaces.
- Expanding the set of half-spaces reduces or maintains the intersection size.
- Example: Any $d = 2$ convex polytope can be expressed as an intersection of half-planes.



Convex Polytope (Definition 2)

Representation via Convex Combinations

Given points $A_1, \ldots, A_n \in \mathbb{R}^d$, the convex hull $CH(A_1, \ldots, A_n)$ is the smallest convex polytope containing these points.

- The vertices of CH belong to $\{A_1, \ldots, A_n\}$.
- Points in CH are convex combinations:

$$\lambda_1 P_1 + \cdots + \lambda_k P_k, \quad \sum \lambda_i = 1, \lambda_i \geq 0. \tag{4}$$

- Carathéodory's Theorem: Any point in CH is a convex combination of at most $d + 1$ vertices.

Supporting Hyperplanes

Faces, Edges, and Facets

A supporting hyperplane of a polyhedron P intersects P but does not divide it into separate parts.

Face of polytope P is the intersection of a sup. hyperplane with P.

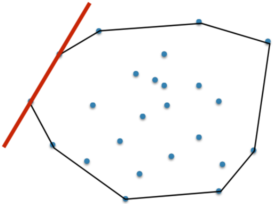
Common Faces:

- **Vertex**: 0-dimensional face.
- **Edge**: 1-dimensional face.
- **Facet**: $(d-1)$-dimensional face.
- **Ridge**: $(d-2)$-dimensional face.
- A d-dimensional polyhedron is considered a face of dimension d.
- The empty set $\emptyset$ is a face of dimension -1.

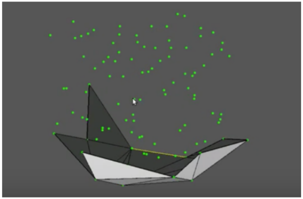
Supporting Hyperplanes (Revisited)

Supporting Hyperplanes of Faces

- Each facet has a unique supporting hyperplane, while other faces have infinitely many.
- For $k = 0$ (vertex), the set of supporting hyperplanes has dimension:
  - 1 in the plane (eliminates one angle),
  - 2 in three-dimensional space (eliminates two angles),
  - $d - 1$ in general.
- The set of supporting hyperplanes of a k-face has dimension $d - k - 1$.



2D

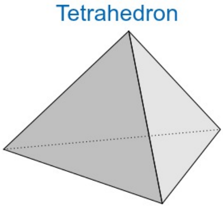


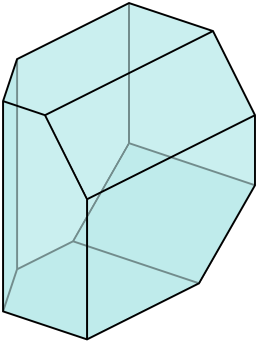
3D

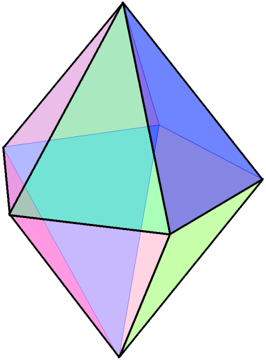
Contents

Types of Polytopes

- Simplex: A polytope with $d+1$ affinely independent vertices.
- Simple Polytopes: Every vertex belongs to exactly d facets.
- Simplicial Polytopes: Each facet is a simplex.







Simplices

Definition

d-**Simplex**: A convex polyhedron $\mathrm{CH}(A_0, \ldots, A_d)$ such that the points $A_i \in \mathbb{R}^d$ are affinely independent, meaning $A_i - A_0$ are linearly independent.

Lemma

Each simplex has:

- $d + 1 = \binom{d+1}{d}$ *facets, each defined by* d *vertices.*
- $\binom{d+1}{2}$ *edges: each pair of vertices defines an edge.*
- *Each* $k + 1$ *vertices define a* k*-simplex with* $\binom{d+1}{k+1}$ *faces of dimension* k*.*

Types of Convex Polyhedra

Definition

Simple Polyhedron: A convex polyhedron where exactly d facets meet at each vertex.

Definition

Simplicial Polyhedron: A convex polyhedron where every facet is a simplex of dimension $d - 1$.

Lemma (Exercise)

- *In the plane, every polygon is both simple and simplicial.*
- *The only simple and simplicial polyhedron in dimensions* ≥ 3 *is the simplex.*
- *Any polyhedron can be made simplicial by triangulating its facets.*
- *Find a i) simple (not simplicial) polytope, ii) simplicial (not simple) polytope, iii) not simple neither simplicial polytope*

Contents

Upper Bound Theorem

Theorem (McMullen)

Any d-dimensional polyhedron with n vertices (or n facets) contains:

$$O\left(n^{\lfloor d/2 \rfloor}\right) \tag{5}$$

k-dimensional faces, for dimensions $k = 0, \ldots, d-1$.

Corollary

- $d = 2$: $O(n)$ *edges and vertices.*
- $d = 3$: $O(n)$ *facets, edges, and vertices.*
- $d = 4$: $O(n^2)$ *facets and edges for n vertices.*

Computational Complexity Results

Corollary

The worst-case computational complexity for computing the convex hull of n points in $\mathbb{R}^d$ *is:*

$$\Omega(n \log n + n^{\lfloor d/2 \rfloor}) \tag{6}$$

Corollary

The storage complexity of the adjacency graph of a polyhedron is:

$$\Omega(n^{\lfloor d/2 \rfloor}) \tag{7}$$

Cyclic Polytopes

- A **cyclic polytope** $C_d(n)$ is the convex hull of n points on the **moment curve** in $\mathbb{R}^d$:

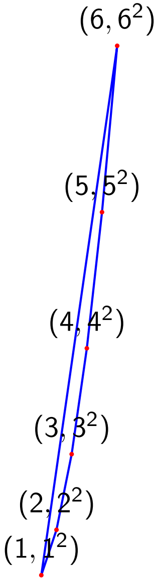
$$(x_1, x_1^2, x_1^3, \ldots, x_1^d), \ldots, (x_n, x_n^2, x_n^3, \ldots, x_n^d)$$

  where $x_1 < x_2 < \cdots < x_n$ are distinct real numbers.
- **Key properties:**
  - **Maximal Simpliciality**: Every facet is a simplex.
  - **Upper Bound Theorem**: Achieves the maximal number of faces for given n and d.
  - **Neighborly Property**: Any $\lfloor d/2 \rfloor$ vertices form a face.

Examples of Cyclic Polytopes

- **Low-dimensional cases:**
  - $C_2(n)$: Convex n-gon in $\mathbb{R}^2$.
  - $C_3(n)$: A 3D convex polyhedron with triangular faces.
  - $C_4(n)$: A 4D polytope with tetrahedral facets.
- **Numerical examples:**
  - $C_2(6)$: Convex hexagon in $\mathbb{R}^2$ with points $(1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36)$.
  - $C_3(6)$: A 3D convex polyhedron with points $(1, 1, 1), (2, 4, 8), (3, 9, 27), (4, 16, 64), (5, 25, 125), (6, 36, 216)$.

$C_2(6)$: Convex hexagon



Contents

Beneath-Beyond Algorithm

General Framework

1. Given the convex hull of k points, insert a new point p.
2. Determine the position of p relative to the convex hull; ignore it if it is inside.
3. Otherwise, compute a *certificate* proving that p is an exterior point.
4. Use this certificate to update the convex hull (preserve part of the hull with p, remove another part).

Incremental Algorithm for Convex Hull in 3D

Input: n points in $\mathbb{R}^3$, in general position.
Output: The convex hull (e.g., as an adjacency graph).

1. Sort points lexicographically by decreasing x_1: $p_1, \ldots, p_n$.
2. Initialization: Start with a tetrahedron from the four rightmost points.
3. For p_k, $k = 5, \ldots, n$:
   - Check facets incident to p_{k-1}: Identify any red facet.
   - Identify all red facets and purple edges.
   - Remove red facets, edges, vertices from the hull.
   - Insert new facets (edges) defined by p_k and purple edges (vertices).
4. Return the updated convex hull.

Correctness of the Beneath-Beyond Algorithm

Lemma (Predicate)

For each facet of the current polyhedron, the following conditions are equivalent:

- *The facet is either blue or red.*
- *The facet is not visible / visible from the new point.*
- *The new point lies in the same / different half-space relative to the supporting plane of the facet.*
- *The sign of the orientation predicate for the facet vertices with the new point is the same / different compared to that with any point inside the current polyhedron.*

Orientation Predicate

Lemma

The orientation of four points $p_i = (x_i, y_i, z_i)$, $i = 0, \ldots, 3$ *reduces to the sign of the determinant:*

$$\det \begin{bmatrix} 1 & x_0 & y_0 & z_0 \\ 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \end{bmatrix}.$$

It is zero if and only if the four points are coplanar.

Complexity of the Beneath-Beyond Algorithm

Lemma

At each incremental step:

- *The set of purple edges/vertices is topologically equivalent to a convex polygon.*
- *This polygon is a convex hull of at most n points, thus has size $O(n)$.*
- *The set of new facets/edges corresponds one-to-one with the purple edges/vertices.*

Overall Complexity for 3D Convex Hull Algorithm

- Initial sorting: $O(n \log n)$.
- Complexity depends on:
  - Total number of red facets/edges, bounded by the total number of constructed facets/edges $O(n^2)$.
  - Total number of red vertices $\leq n$.
  - Number of purple edges/vertices $= O(n)$ per step.
  - Number of constructed facets/edges $= O(n)$ per step.
- Overall complexity: $O(n^2)$.

Alternative: unsorted insertion, point location, randomized $O(n \log n)$

Generalization to Higher Dimensions

- New convex hull $C' = \text{CH}(C \cup \{p\})$.
- Facets of C split into two categories:
  - $F =$ blue / red if p is in same / different half-space.
- General position ensures p is not on a supporting hyperplane.
- Lower-dimensional faces split into:
  - Red: Intersection of only red facets.
  - Blue: Intersection of only blue facets.
  - Purple: Intersection of both red and blue facets.

Overall Complexity of Beneath-Beyond in dD

Theorem

Given n points in $\mathbb{R}^d$, the worst-case time complexity for constructing the convex hull is:

$$O(n \log n + n^{\lfloor (d+1)/2 \rfloor})$$

which is optimal only for even dimensions.

Theorem (Seidel)

Using randomized techniques, an expected time complexity of $O(n \log n + n^{\lfloor d/2 \rfloor})$ can be achieved.

Theorem (Chazell)

A more complicated deterministic version has time complexity of $O(n \log n + n^{\lfloor d/2 \rfloor})$ (worst-case optimal).

Gift Wrapping Approach

- n points in general position in $\mathbb{R}^d$: every d points define a hyperplane, and no $d+1$ points lie on the same hyperplane.
- Data structure RACH stores known ridges for examination (one adjacent facet is known).
- Ridges are stored as $(F - \{x\}, x)$ where:
  - F is the set of points defining a facet containing the ridge.
  - x is the vertex of the facet not in the ridge.

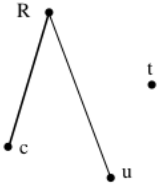
Function FIND-OTHER-FACET

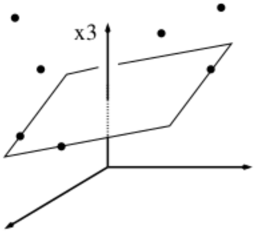
Input: Ridge R and vertex $c \notin R$, where $R \cup \{c\}$ forms a facet of the convex hull.
Output: A facet of the convex hull containing R and different from $R \cup \{c\}$.

1. Select candidate point u not in $R \cup \{c\}$.
2. For each $t \notin R \cup \{c, u\}$:
   - If c and t are in different half-spaces relative to the hyperplane of $R \cup \{u\}$, then update $u \leftarrow t$.
3. Return facet $R \cup \{u\}$.

Complexity $= O(n)$ calls to CCW, thus $O(nd^3)$.

Examples





- Execution of FIND-OTHER-FACET(R, c) in $\mathbb{R}^2$, where ridge R is a vertex.
- Finding the first facet of the convex hull in $\mathbb{R}^3$ (lower hull).

Initialization: Searching for a Supporting Hyperplane

- Equation of a hyperplane:

$$k_1 x_1 + \cdots + k_{d-1} x_{d-1} + k_d x_d + \lambda, \quad k_1, \ldots, k_d, \lambda \in \mathbb{Q}. \quad (8)$$

- Searching for a facet non-parallel to the x_d-axis, meaning the hyperplane intersects the axis: $k_d \neq 0$.
- Can be written as:

$$x_d = k_1 x_1 + \cdots + k_{d-1} x_{d-1} + \lambda, \quad k_1, \ldots, k_{d-1}, \lambda \in \mathbb{Q}. \quad (9)$$

- Such a facet exists if the volume of the convex hull is > 0 in $\mathbb{R}^d$.

Constraints of the Supporting Hyperplane

- Each input point $p_i = (p_{i1}, p_{i2}, ..., p_{id})$ must satisfy:

$$k_1 p_{i1} + k_2 p_{i2} + \cdots + k_{d-1} p_{i(d-1)} + \lambda \leq p_{id}. \tag{10}$$

- Points satisfying equality lie on the facet; the rest are above it.
- If the convex hull is bounded, such a supporting facet defines the lower boundary.
- The hyperplane intersects the x_d-axis as high as possible by maximizing λ.

Exercise

Apply this initialization in two dimensions: which edge is computed?

Linear Program for Initializing Gift Wrapping

- The first facet of the convex hull is found by solving the following linear program:

Linear Program

$$\begin{aligned} \text{Maximize} \quad & \lambda \\ \text{subject to} \quad & k_1 p_{i1} + k_2 p_{i2} + \cdots + k_{d-1} p_{i,d-1} + \lambda \leq p_{i,d}, \quad \forall i = 1, \ldots, n. \\ & k_d = 1. \end{aligned}$$

Wrapping Algorithm for Convex Hull

general dimension

Input: n points in $\mathbb{R}^d$ in general position.
Output: Convex hull representation.

1. Compute and print an initial facet F.
2. Initialize the RACH structure with ridges $(F - \{x\}, x)$ for all $x \in F$.
3. While RACH has elements:
   - Let $(R, c) \in$ RACH.
   - Compute and print $F \leftarrow$ FIND-OTHER-FACET(R, c).
   - For each vertex $x \in F$:
     - If a ridge $(F - \{x\}, y)$ exists in RACH, delete it.
     - Otherwise, insert $(F - \{x\}, x)$ into RACH.

Complexity of Gift Wrapping Algorithm

- Initial facet: Solving a linear program with n constraints in d dimensions, cost $O(n)$ *[Megiddo]*.
- Initializing RACH $= O(d)$.
- Wrapping:
  - Searching and adding ridges in RACH $O(\log n^{\lfloor d/2 \rfloor}) = O(d \log n)$.
  - Each FIND-OTHER-FACET call takes $O(nd^3)$.
  - In a simplicial polyhedron, there are $O(d)$ points per facet $\Rightarrow$ cost $O(d^2 \log n)$.
  - Total time complexity $= O(nHd^3)$, where H is the number of facets.
  - Is it output-sensitive?