Convex Hull: (Intuitive definition)

Given a point set $P$ in $\mathbb{R}^2$, imagine snipping a rubber band around the points.

$P$ \rightarrow \text{conv}(P)$

Uses:
- Shape approximation (intersection test)
- First step in other algorithms
  - Diameter
  - Width

$\text{diam}(P)$
$\text{wid}(P)$
Basic Definitions:

$\mathbb{R}^d$ - Real d-dim space $p=(p_1,\ldots,p_d)$, $p_i \in \mathbb{R}$
- Refer to as points $(p,q)$ - location
- vectors $(u,v,w)$ - displacement

$\mathbb{R}$ - scalars $\alpha, \beta, \gamma, \ldots$

usual ops from linear algebra:
$u+v, u-v$ - vector addition
$\alpha u$ - scalar multiplication
$u \cdot v$ - dot product $= \sum_{i=1}^{d} u_i v_i$

Affine + Convex Combinations:

Affine combination: $\sum_{i=1}^{k} \alpha_i p_i$: $\sum_{i=1}^{k} \alpha_i = 1$

Convex combination: $\ldots$ and $0 \leq \alpha_i \leq 1$
Lines, Hyperplanes, Halfspaces:

Given non-zero vector $u$ + scalar $\alpha$,

$$h(u, \alpha) = \{ p \in \mathbb{R}^d \mid p \cdot u = \alpha \}$$ is hyperplane.

If $\|u\| = 1$

$$h^+(u, \alpha) = \{ p \in \mathbb{R}^d \mid p \cdot u \geq \alpha \}$$

Euclidean Ball:

$$\text{dist}(p, q) = \| p - q \| = \left( \sum_{i=1}^{d} (p_i - q_i)^2 \right)^{\frac{1}{2}}$$

$$B(q, r) = \{ p \in \mathbb{R}^d \mid \| p - q \| \leq r \}$$

(Euclidean) ball of radius $r$ centered at $q$. 
Convexity:
A set \( K \subseteq \mathbb{R}^d \) is convex if \( \forall p, q \in K \) the line segment \( \overline{pq} \) (equiv. any convex combination of \( p + q \)) lies within \( K \).

Support Hyperplane:
Given convex \( K \) and any point \( p \in \partial K \), \( \exists \) hyperplane passing through \( p \) with \( K \) lying all on one side.

Convex Hull:
Given a set \( P \) of points in \( \mathbb{R}^d \), the convex hull, \( \text{conv}(P) \), is the smallest convex set containing \( P \).

- The set of all convex cumbw in \( P \)
- The intersection of all halfspaces containing \( P \)
General Position:

Geometric algorithms are complicated by rare (?) degenerate cases:
- Points having same coordinate
- ≥ 3 collinear points
- ≥ 4 cocircular points

To simplify algorithm presentation we often assume these do not arise in the input. Called general-position assumption

(Planar) Convex Hull Problem: Given a set of n pts $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$ ($p_i = (x_i, y_i)$) compute $\text{conv}(P)$.

Output: Cyclic ordering of vertices on the hull
weak hull: possible output: (indices) $\langle 4, 3, 7, 9, 8, 1 \rangle$

Note: $p_5$ not output
(can assume this away by "general position")

Alternative output: (left to right)
Upper-hull + Lower-hull
$\langle 7, 3, 4, 1 \rangle + \langle 7, 9, 8, 1 \rangle$
Graham’s Scan: $O(n \log n)$ solution
- Compute upper + lower hulls separately
  - Upper-hull:
    - Sort pts by x-coords
    - Add each to upper hull
    - Remove pts no longer on hull
  - Lower-hull: (symmetrical)

Observations:
- The rightmost pt always on hull
- Reading right to left, consecutive triples on the hull form left-hand turns

Incremental Approach:
- Store vertices (indices) of upper hull on stack
- For each new point $p_i$ (left to right)
  - While $\langle p_i, S[top], S[top-1] \rangle$ do not form LHT - pop $S$
  - Push $p_i$
Example:

How to test for LHT?

**Orientation test**

Given a sequence \( \langle p, q, r \rangle \) of 3 pts in \( \mathbb{R}^2 \)

\[
\text{orient}(p, q, r) = \text{sign} \left( \det \begin{pmatrix}
1 & p_x & p_y \\
1 & q_x & q_y \\
1 & r_x & r_y
\end{pmatrix} \right)
\]

is:
- +1 if they are oriented CCW (LHT)
- -1 if they are oriented CW (RHT)
- 0 if they are collinear (or duplicates)
Graham's Scan: (Upper Hull only)
- Sort pts by increasing x-coords $\langle p_1, \ldots, p_n \rangle$
- Push $p_1, p_2$ onto $S$
- for $i \leftarrow 3$ to $n$
  - while ($|S| \geq 2$ and $\text{orient}(p_i, S[t], S[t-1]) \leq 0$) pop $S$
  - push $p_i$

Correctness: (Sketch)

**Lemma:** After processing $p_i$, $S$ contains upper hull of $\langle p_1, \ldots, p_i \rangle$

**Proof:** By induction on $i$.
- $p_i$ must be last vertex of hull
- all the popped pts are not on upper hull
- all remaining pts are on upper hull

(omit the details)
Running time:
- \( O(n \log n) \) to sort
- for \( 3 \leq i \leq n \), let \( d_i = \text{num. of pops} \) when inserting \( p_i \)

- Time for scan is:

\[
\sum_{i=3}^{n} (d_i + 1) \leq n + \sum_{i=3}^{n} d_i
\]
for pops / for push of \( p_i \)

- Note that \( \sum d_i \leq n \rightarrow \text{Why?} \)

- Total time: \( O(n \log n + 2n) = O(n \log n) \)

Also see lecture notes for a hull algorithm based on divide + conquer