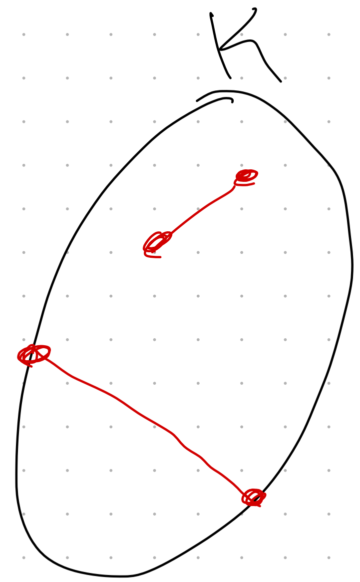
Convex Hulls

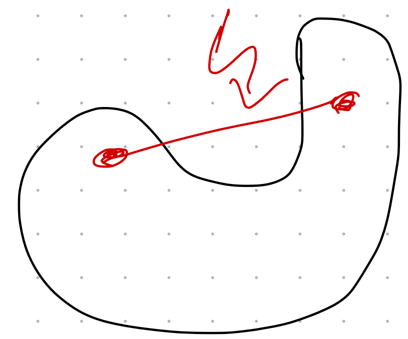
Convex Hulls

Definition I: convexity

Given $K \subseteq \mathbb{R}^d$;

K is <u>convex</u> $:\Leftrightarrow \forall p, q \in K : \overline{pq} \subseteq K$





Lemma:

If K_1, K_2 are both convex,
then $K_1 \cap K_2$ is convex, too.

Proof: exercise

Exercise: show that a straight line can intersect K in at most 2 pts.

Definition II: convexity

Let $S \subseteq \mathbb{R}^d$.

Define <u>convex hull</u> $CH(S) := \bigcap_{\substack{K \supseteq S \\ K \text{ convex}}} K$

i.e., CH is the smallest convex set containing S.

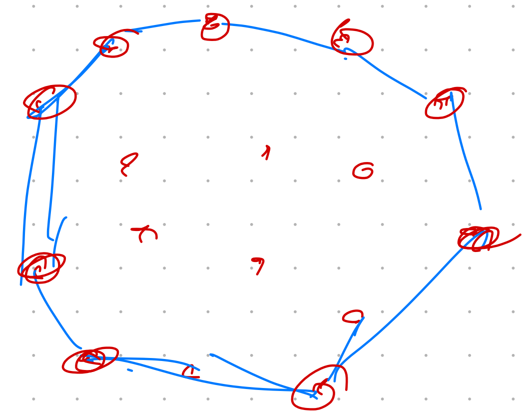
In the following, S is always finite (and discrete) !

Theorem:

Let $S \subseteq \mathbb{R}^d$ finite set of pts.

The $CH(S)$ is a polyhedron (polytope)

that has vertices in S.

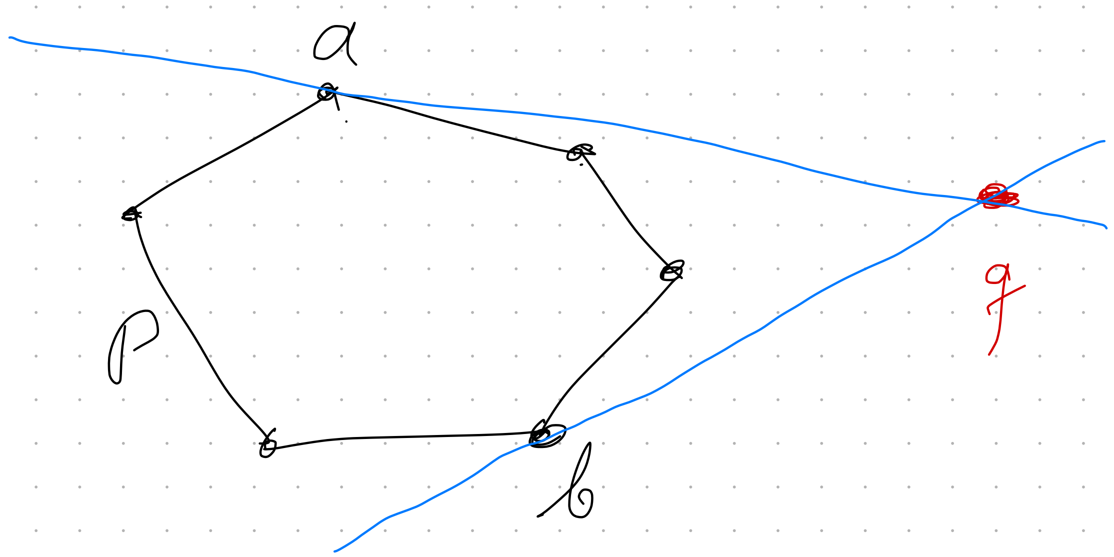


Definition: "tangent" (in 2D)

Given polygon P in 2D, pt q outside P.

A "tangent to P through q" is a line through q touching P in exactly one pt.

(a.k.a. "supporting line")



Observe: there are 2 tangents.

Proof: by induction over $n = |S|$

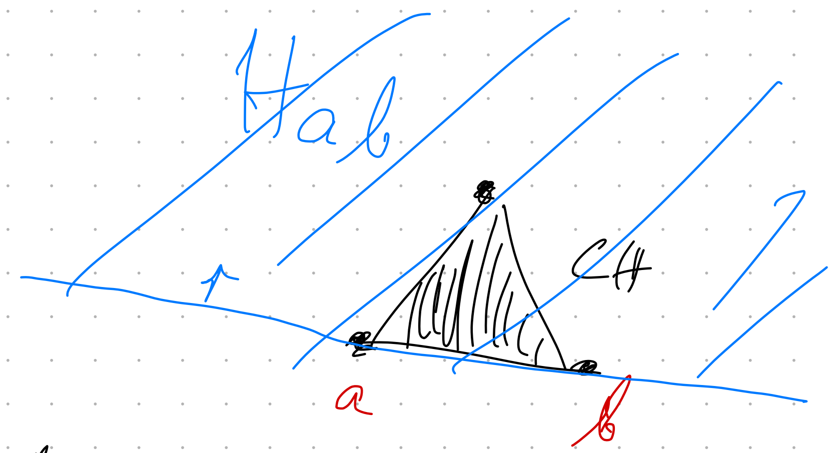
start: $S = \{a, b, c\} \Rightarrow CH(S) = \triangle abc$

Proof: $\triangle abc = H_{ab} \cap H_{bc} \cap H_{ac}$

$\triangle abc$ is convex, since intersection of convex sets.

$\triangle abc$ contain S

$\triangle abc$ is smallest convex set containing S



Step: $|S| > 3$

choose any $p \in S$, let $S' := S \setminus \{p\}$

let $C' = CH(S')$ $\rightarrow$ vertices of $C' \subseteq S' \subseteq S$.

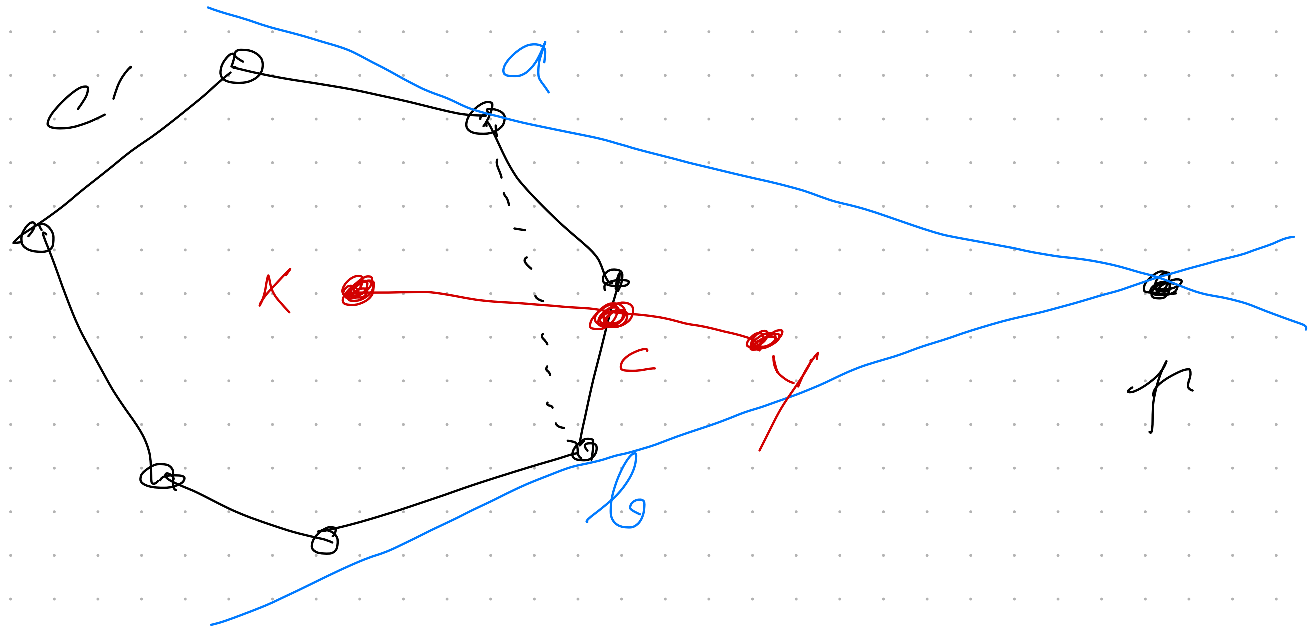
Idea: construct $CH(S)$ out of C' and p

Case 1: $p \in C'$

$\rightarrow$ nothing to be done, $C = C'$

Case 2: $p \notin C'$

find tangents through p to C' $\rightarrow$ $\underbrace{a, b}_{\in S'} \in C'$



Claim: $CH(S) = C' \cup \Delta abp$

Obviously,

$\forall$ convex $K \supseteq S$: $C' \subseteq K \wedge \Delta abp \subseteq K$

$\Rightarrow C' \cup \Delta abp \subseteq CH(S)$

Still to prove: $\forall x \in C', y \in \Delta abp : \overline{xy} \subseteq C' \cup \Delta abp$

Consider $y \in \Delta abp \setminus C'$

$c :=$ intersection pt $\overline{xy}$ with border of C'

$\rightarrow$ c must "between" tangents

$$\Rightarrow \quad \overline{cy} \subseteq \Delta abp \,,$$

$$\overline{xc} \subseteq C' \qquad \Rightarrow \quad \overline{xy} \subseteq C' \cup \Delta abp = C$$

Simple Applications

1. Mixing gases:

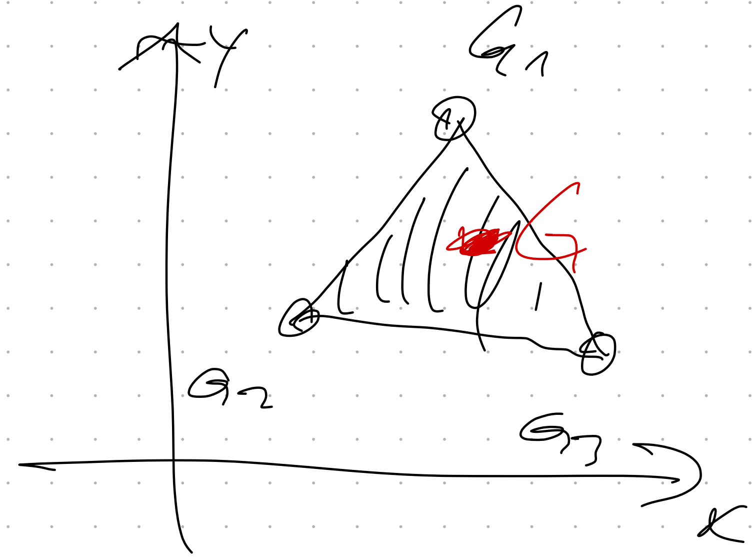
Given gases $G_i = (x_i, y_i)$, x_i = G_i contains x_i% of Oxygen

y_i = — " — y_i% of nitrogen

Question: can we mix G_1, G_2, G_3 so that $G = (x, y)$ results?

Solution: "mixing" = finding weighted sum

$$G = \sum \lambda_i G_i \,, \quad \lambda_i \geq 0 \,, \quad \lambda_i = 1$$

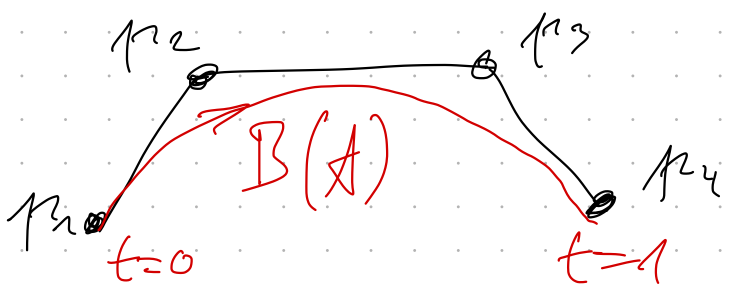


Answer: "yes" iff

$$G \in CH(G_1, G_2, G_3)$$

2. Bezier curves: "convex hull property"

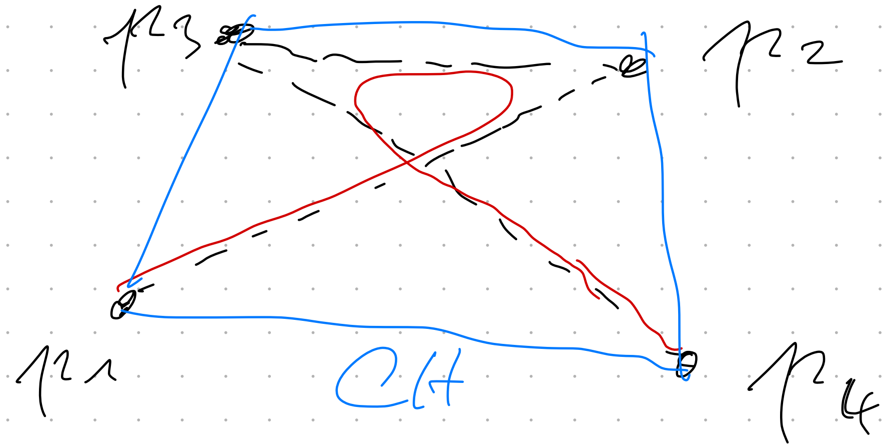
consider degree 3: constructed over "control polygons" P with 4 pts

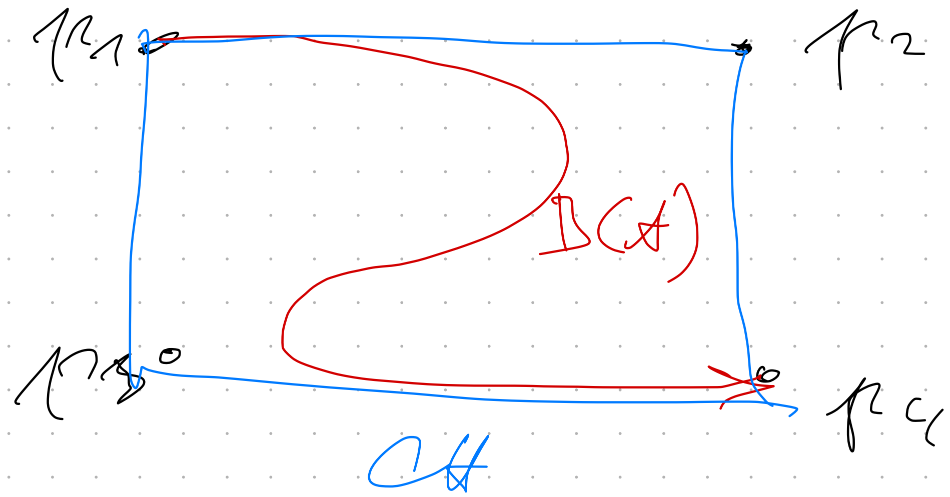


$P = \{p_1, \dots, p_k\}$

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} \cdot t^i \cdot p_i$$

Whole Bezier curve $\subseteq CH(P)$





Properties of CH

Konvex Combination:

Given k pts $\{p_1, \dots, p_k\} = S \subseteq \mathbb{R}^d$,

Def. "convex combination"

$$p = \sum_{i=1}^{k} \lambda_i p_i \quad , \quad \underline{\underline{\sum \lambda_i = 1}} \quad , \quad \forall \lambda_i \geq 0$$

"barycentric coords"

Theorem (w/o proof):

$$CH(S) = \{ \sum \lambda_i p_i \mid \sum \lambda_i = 1, \ \lambda_i \geq 0 \}$$

Hint for proof: via induction

start with p_1, p_2 :

$\lambda_1 p_1 + \lambda_2 p_2 \in \overline{p_1 p_2} \Rightarrow \lambda p_1 + (1-\lambda) p_2 \in \overline{p_1 p_2}$

provided $\lambda_1 + \lambda_2 = 1$, $\lambda_1, \lambda_2 \geq 0$

$\cap$

$CH\begin{pmatrix} p_1, \\ p_2 \end{pmatrix}$

2. Lower bound on construction algos:

Lemma:

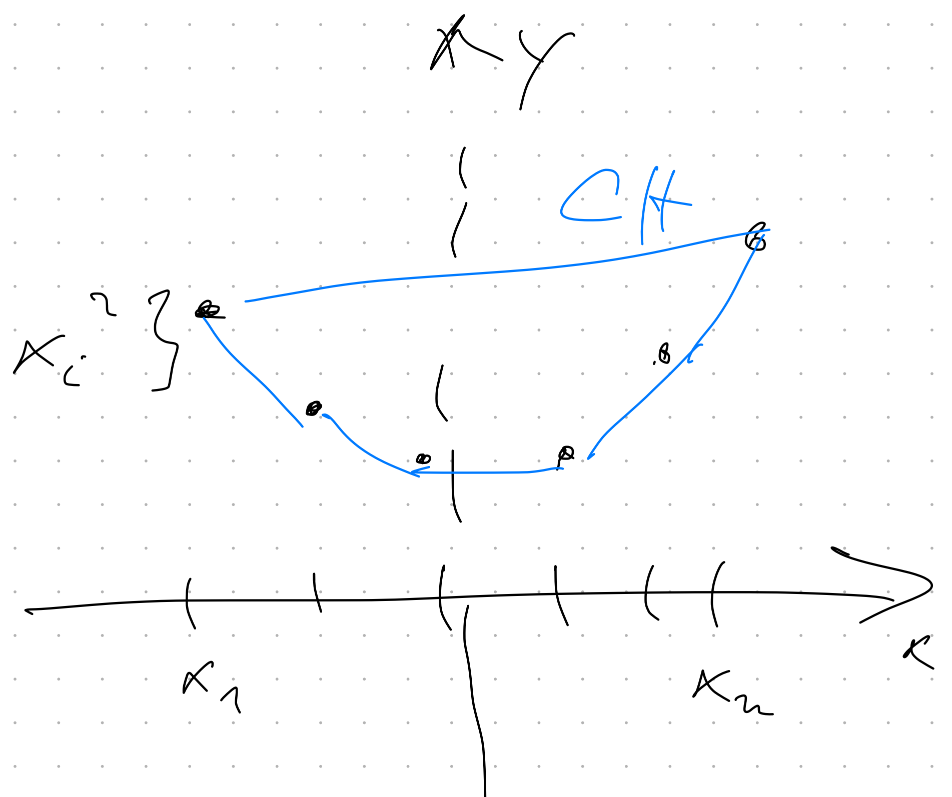
Any construction of $CH(S)$ takes time $\Omega(n \log n)$.

Proof: reduce to sorting

given $x_1, \dots, x_n \in \mathbb{R}$

construct $S = \{ (x_i, y_i) \mid y_i = x_i^2 \}$

$\Rightarrow$ all $p_i \in CH(S)$



3. "Rubberband property"

Lemma:

The border of CH(S) in the plane is the shortest curve around S, which is closed and simple.

Proof:

Let W = curve around S (should be simple)

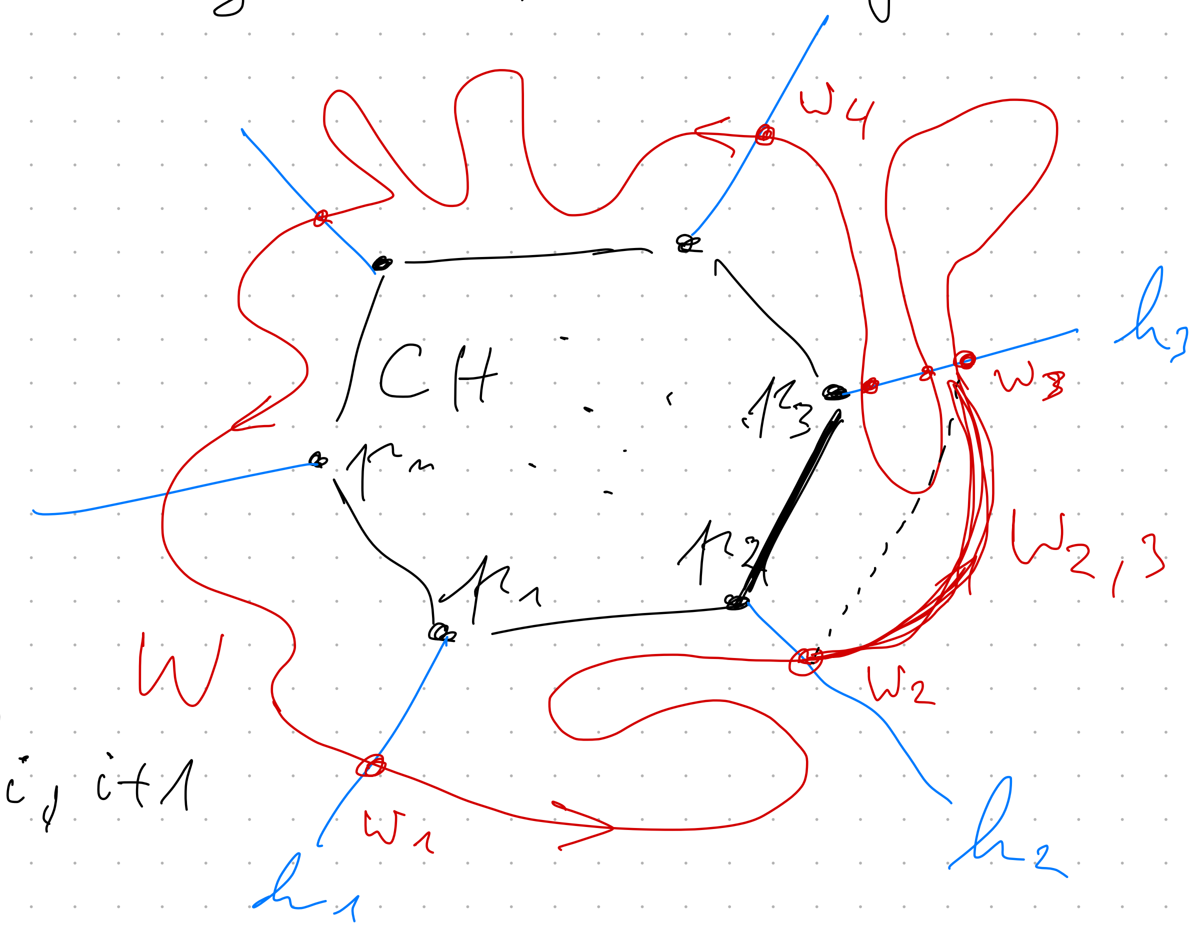
To prove: W is at least as long as border of CH

Construct "outward rays"

Assume direction of W.

Construct first intersection pts of W with h_i

$\rightarrow$ w_i



$\rightarrow$ $\|p_{i+1} - p_i\| \leq \|w_{i+1} - w_i\| < W_{i,i+1}$

Construction in 2D

Problem:

Given a set S of n pts in plane;
construct CH(S) by finding those pts in S
that appear as vertices of CH in
counter-clockwise order.

Assumption: "general position"

Here: no two pts in S have the same x-coord,
nor the same y-coord!

Naive Algo:

$C := \emptyset$ // set of candidate pts from S

for all $(p, q) \times S \times S$, $p \neq q$:

  for all $r \in S \setminus \{p, q\}$:

    if r is left of $\overline{pq}$:

      ignore p, q;

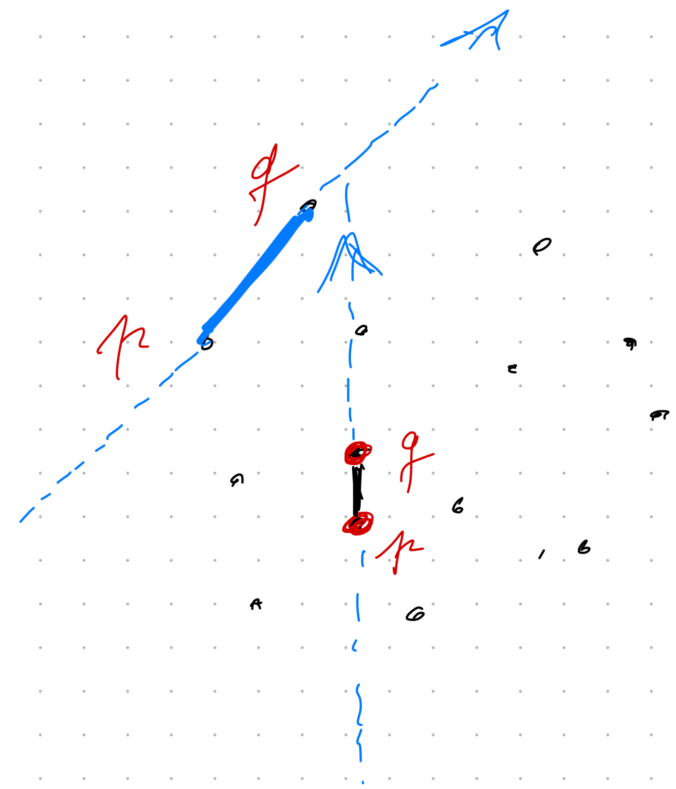
      consider next (p, q)

  end for r

  add (p, q) to C

end for

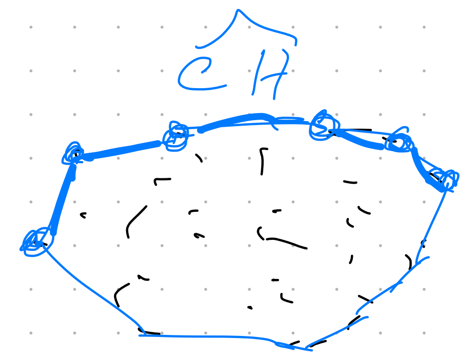
sort C



Running time: $O(n^3)$

Better approach: "incremental computation"

→ add pts one by one

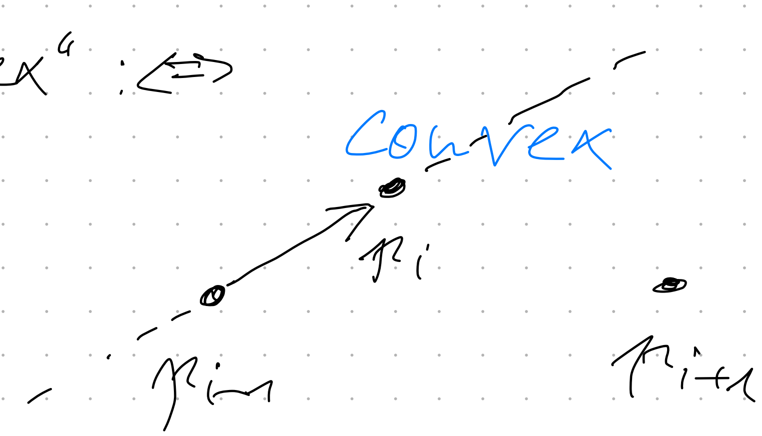


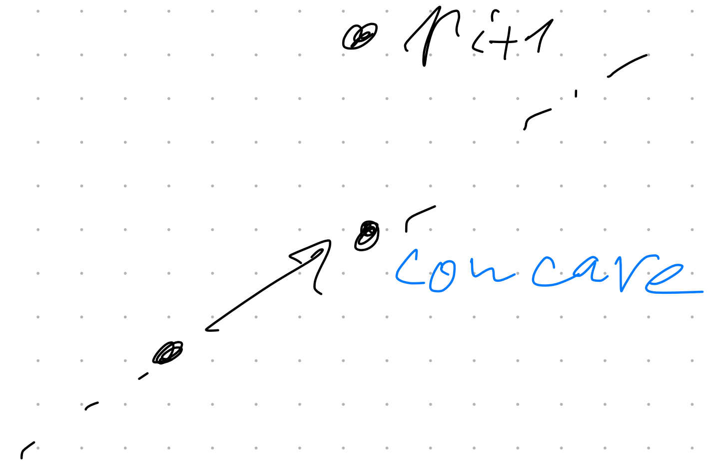
Def.: "upper hull" $\widehat{CH}(S)$ = sub segment of $CH(S)$ with monotonically increasing x-coord

Def.: a vertex p_i in a polygon is called "convex" $:\Longleftrightarrow$

p_{i+1} is right of $\overline{p_{i-1}\,p_i}$ $\Longleftrightarrow$

$\text{area}(\Delta\, p_{i-1}\, p_i\, p_{i+1}) < 0$





Algo "Graham's Scan":

sort S by x-coord $\rightarrow p_1, \dots, p_n$

init $\hat{C} := \{p_1, p_2\}$ // working set for $\widehat{CH}(S)$

for $i = 3 \dots n$:

  $\hat{C} \mathrel{+}= \{p_i\}$

  while $|\hat{C}| > 2$ and

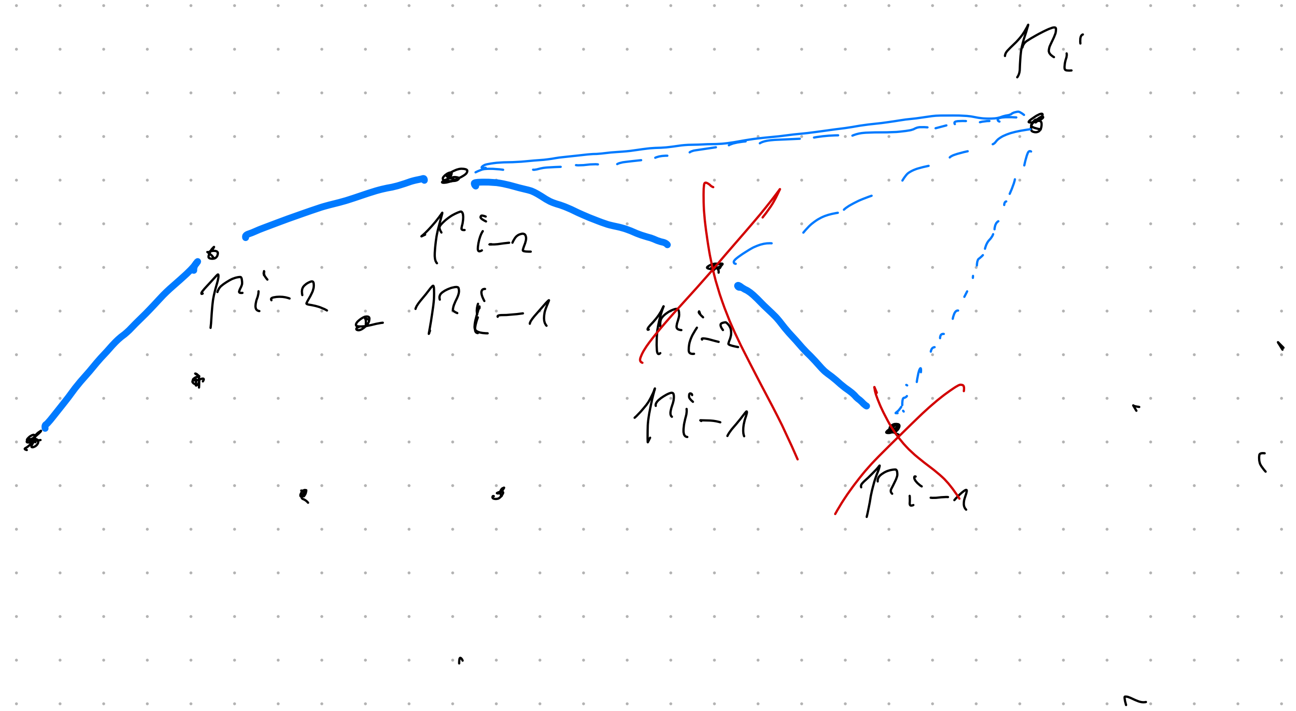
    $(p_{i-2}, p_{i-1}, p_i) \in \hat{C}$ form a concave vertex:

   delete p_{i-1} from $\hat{C}$

end for

analogous for lower hull $\check{C}$

Example:



Running time: $O(n \log n)$

Proof:

each pt in S gets deleted exactly once from $\hat{C}$;
— " — added — " —

$\Rightarrow$ body of while loop is executed at most n times in total over all i's

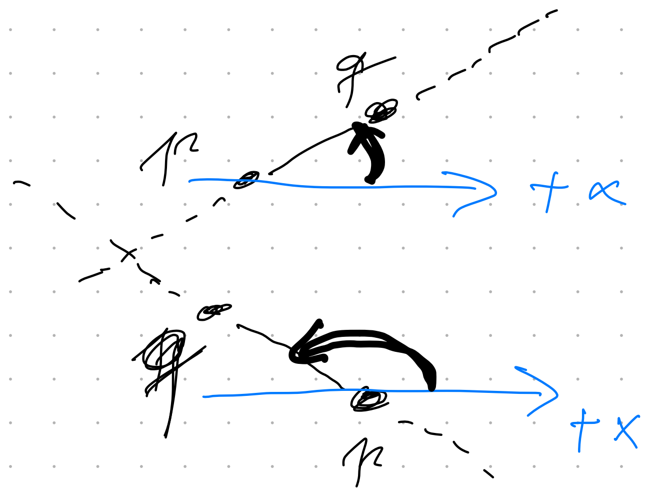
$\Rightarrow$ $O(n) + \underbrace{O(n \log n)}_{\hat{=}\text{ sorting}}$ in worst-case

In case of radix sort : $O(n)$

Algo "Gift wrapping" (aka. Jarvis' March) :

observe : if $\overline{pq} \in CH(S) \Rightarrow \exists r \in S : \overline{qr} \in CH(S)$

Notation : $\measuredangle(\overline{pq}, x) =$ angle between $\overline{pq}$ and pos. x-axis



construct right-hand hull :

let p_0 = pt in S with smallest y-coord
p^* = —"— top-most y-coord
$\}$ (**)

$p_i = p_0$
while $p_i \neq p^*$
  delete p_i from S ,
  output p_i
  $p_{i+1} := \text{find} \min_{p \in S} \measuredangle(\overline{p_i p}, x)$ (*)
end while
output p^*

Running time:

Let $h = |CH(S)|$ = # pts on CH

$\Rightarrow$ while-loop is iterated h times

step (*) $\in \Theta(n)$, step (**) $\in O(n)$

$\Rightarrow O(n \cdot h)$

$\Rightarrow$ Jarvis March is "output sensitive"

Def.:

An algo is called output sensitive, if its complexity is in $O(f(n,h))$, where h = size of the output (e.g. # pts)

$\rightarrow$ if $h \ll \log n$, then gift wrapping is faster than Graham

if $h \in O(n)$, then Jarvis' $\in O(n^2)$

Algo "Chan's algorithm":

Approach:

Do h phases

1) "Divide": "mini convex hulls"

2) "Merge": gift wrapping

Trick: determine good size for "mini hulls"

HullG(S, m, h'):

input: $S = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$

$3 \le m \le n$

$h' \geq 1$ "guessed value for h"

let $k = \lceil \frac{n}{m} \rceil$

partition $S = S_1 \dot\cup \dots \dot\cup S_k$

for $i = 1 \dots k$: "phase 1"

compute $C_i := CH(S_i)$

$p_1 :=$ lowest pt in S ; output p_1

for $l = 1 \dots h'$: "phase 2"

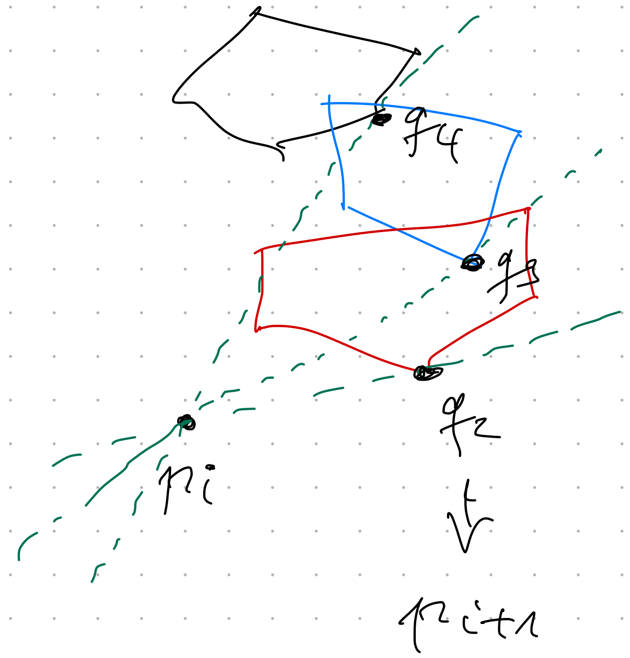
for $i = 1 .. k$: (**)

calculate the lower tangent on C_i through p_i

$\Rightarrow q_i$ = pts on C_i and those tangents

end for

let $p_{i+1} := \text{find } \min_{q \in \{q_1, .., q_k\}} \{ \measuredangle(\overline{p_i q}, x) \}$



output p_{i+1}

if p_{i+1} = top-most pt in S:

algo has output $l+1$ pts } (*)

return "CH is complete"

end for l

return "CH is incomplete"!

Observe: if $h' \geq h$, then (*) must happen

Running time of HullG:

$\forall i: |S_i| \leq m \Rightarrow C_i \in O(m \log m)$ time

$\Rightarrow$ phase 1 $\in O(k \cdot m \log m) \underset{k = \frac{n}{m}}{=} O(n \log m)$

Each tangent $\in O(\log m)$ time

Loop (**) $\in O(h \cdot \log m) \Rightarrow$ total $\in O\left(h' \frac{n}{m} \log m + n \log m\right)$

Choose $m = h'$

$T_{HullG} \in O(n \log h')$

Idea: double exponential search

Algo Hull(S):

for $t = 1, 2, \dots$:

$h' := 2^{2^t}$

$C :=$ HullG(S, h', h')

if C is "complete":

return C // reached $h' \geq h$

Running time:

iterations $= \lceil \log \log h \rceil$

t-th iteration takes time $O(n \log h') = O(n \cdot 2^t)$

Total $\sum_{t=1}^{\log\log h} O(n \cdot 2^t) = O\left(n \cdot \sum_{t=1}^{\log\log h} 2^t\right) = O\left(n \cdot 2^{\log\log h + 1}\right)$

$= O(n \log h)$

Notion: Complexity of a geometric data structure
is a $f(n)$ = size of the output/data structure
e.g., # pts + "combinatorial structure"
where n = size of input (e.g. # pts)

Intermezzo: Euler's Equation

Theorem (Euler's Equation):
Let P be a convex polyhedron, $v = |V_P|$, $e = |E_P|$, $f = |F_P|$.
Then,

$$\boxed{v - e + f = 2.}$$

Proof: see course ACG

Corollary (polyhedron complexity):

For every convex polyhedron,

$$v, e, f \in \Theta(v, e, f).$$

E.g., $v \in \Theta(e)$, $v \in \Theta(f)$, ...

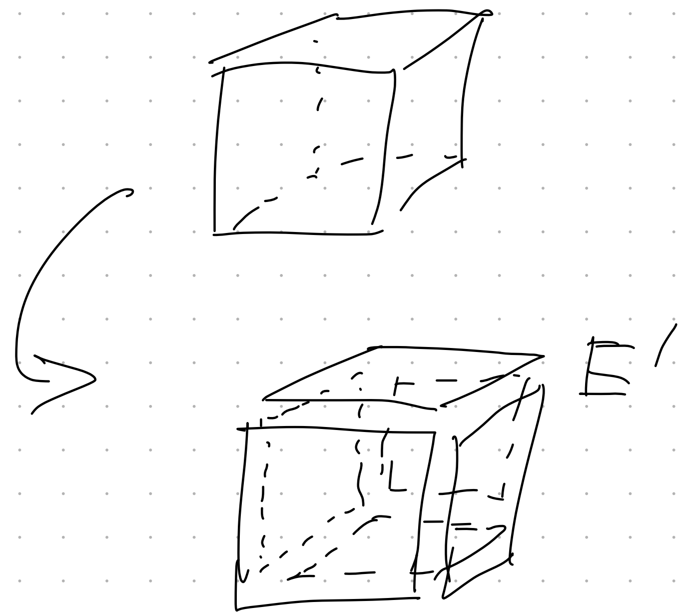
Proof:

Replace each edge by 2 edges (virtual) $\Rightarrow E'$

$\Rightarrow$ each $e \in E'$ belongs to exactly 1 face,

each face has ≥ 3 edges,

$\Rightarrow \quad 3 \cdot f \leq e' = 2 \cdot e \quad \Rightarrow \quad f \leq \frac{2}{3} e$



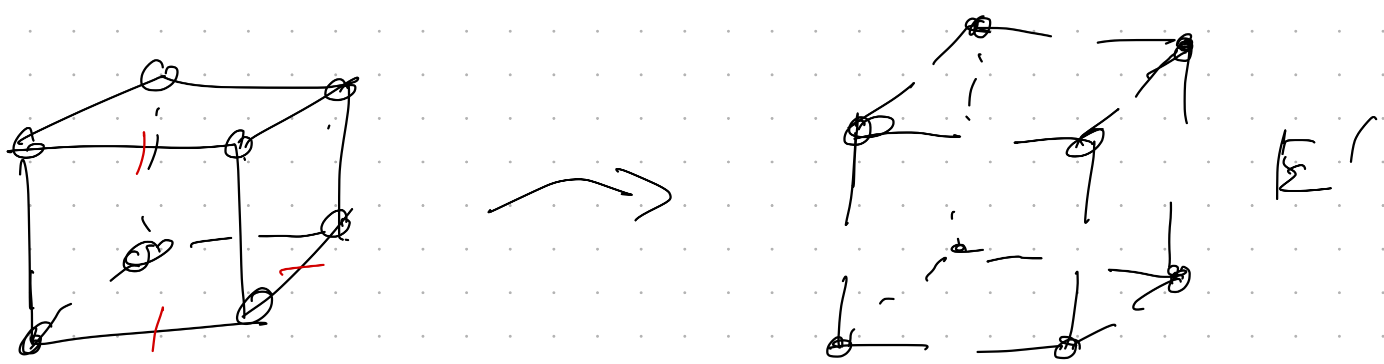
Plug into Euler:

$$2 = v - e + f \leq v - e + \frac{2}{3} e \Rightarrow e \leq 3v - 6$$

$$\Rightarrow \frac{3}{2} f \leq e \leq 3v - 6$$

Again for vertices,

Every vertex incident to ≥ 3 edges



$\Rightarrow \quad v \leq \frac{2}{3} e \quad , \quad e \leq 3f - 6$

Total:

$$\frac{2}{3} v \leq e \leq 3v - 6$$

$$\frac{2}{3} f \leq e \leq 3f - 6$$

$$\frac{1}{2} f + 3 \leq v \leq 2f - 4$$

...

Corollary:

The complexity of the CH in 3D is in $O(n)$.

Notes:

1.) In d dimensions, the complexity of CH is in $O(n^{\lfloor d/2 \rfloor})$:
"curse of dimensionality"

2. The complexity in the corollary of Euler is true for any polyhedron that is closed and 2-manifold.

Convex Hull in 3D (Clarkson–Shor)

Randomized, incremental

Input: $S = \{p_1, \dots, p_n\}$ (randomized order)

Start:

pick the first two pts p_1, p_2

find p_3 <u>not</u> on line through $\overline{p_1, p_2}$

find p_4 <u>not</u> on plane through p_1, p_2, p_3

$\Rightarrow$ tetrahedron $p_1, p_2, p_3, p_4 =: \mathcal{C}_4$

permute $p_5, \dots, p_n$ randomly

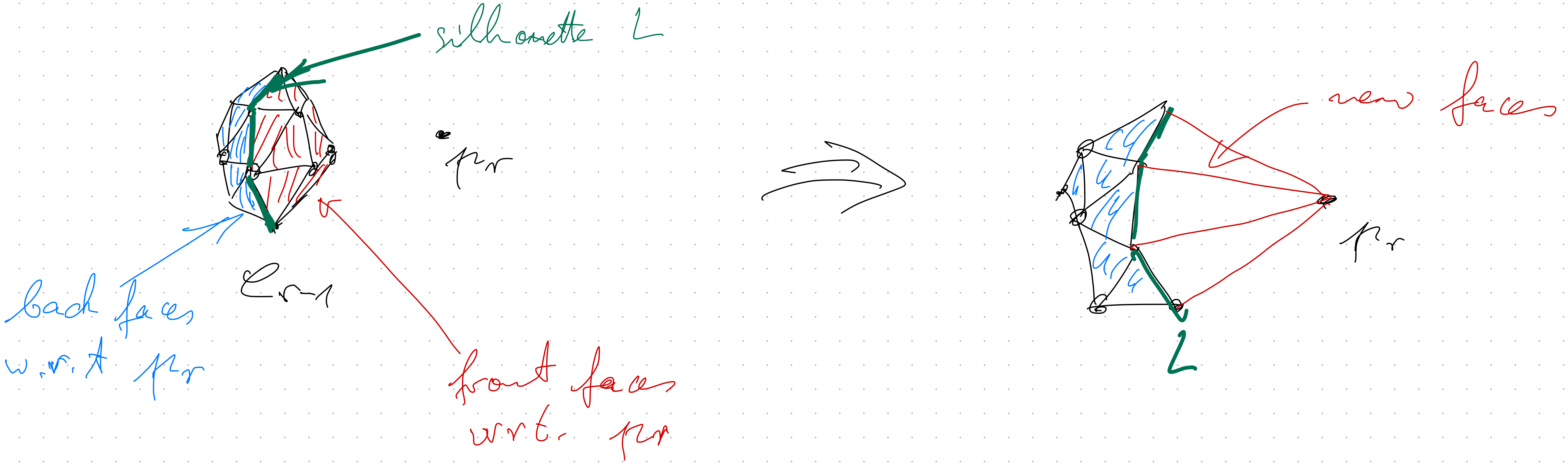
for $p_r = p_5, \dots, p_n$:

combine $\mathcal{C}_{r-1}$ and $p_r \rightarrow \mathcal{C}_r$, convex hull over $\{p_1, \dots, p_r\}$

Case 1:

$p_r \in \mathcal{C}_{r-1} \Rightarrow$ discard p_r

Case 2: $p_r \notin \mathcal{C}_{r-1}$:



How to find all front faces quickly?

Brute force $\Rightarrow O(n^2)$ algo

Idea: maintain "conflict graph"

Maintain 2 conflicting sets

for all faces $f \in \mathcal{C}_r$: $P_{conf}(f)$ = all pts that can "see" f

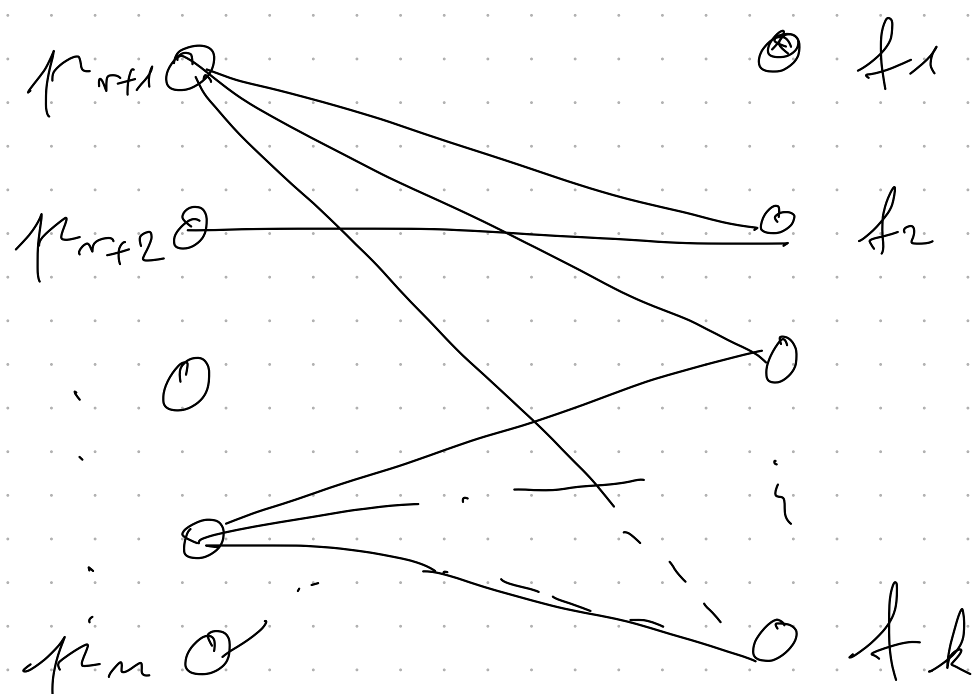
$\subseteq \{p_{r+1}, \dots, p_m\}$

for all pts p_s, $s > r$: $F_{conf}(p_s)$ = all faces that are front faces w.r.t. p_s

$\subseteq F(\mathcal{C}_r)$

We say $p \in P_{conf}(f)$ "is in conflict with f".

Auxiliary data structure: bipartite graph $\mathcal{G}$



Usage of $\mathcal{G}$ in the overall algo:

for all $f \in F_{\text{confl}}(p_r)$: // front faces wrt. p_r

delete f from $\mathcal{C}_{r-1}$

if one of f's edges is on L:

construct triangle f' with p_r and edge

add f' to $\mathcal{C}_r$

↑ pts not yet consumed, shrinks over iterations

↑ faces in current $\mathcal{C}_r$, grows over iter's

Complexity $\in O(|F_{\text{confl}}(p_r)|)$

Init. of $\mathcal{G}$:

start with $\mathcal{C}_4$

test all $p_5, \dots, p_n$ against the 4 faces of $\mathcal{C}_4$

Updating: $\mathcal{C}_{r-1} \to \mathcal{C}_r$

1. delete $\underbrace{\text{all neighbors of } p_r}_{= F_{\text{confl}}(p_r)}$ in $\mathcal{G}$
2. delete p_r from $\mathcal{G}$
3. create new nodes for the new faces

4. create new edges for conflicts

Observe:

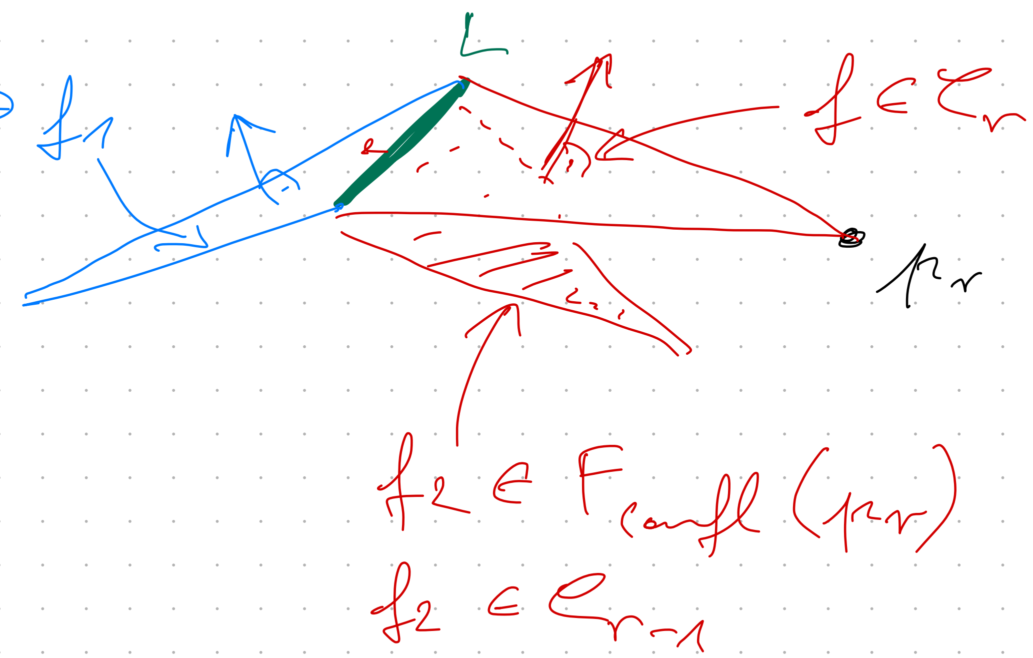
If $f \in \mathcal{C}_{r-1}$ and $f \in \mathcal{C}_r$, then $f \notin F_{confl}(p_r)$,

then $P_{confl}(f)$ stays same

$\Rightarrow$ only compute $P_{confl}(f)$ for new f's (which contain p_r)

$\subseteq \{p_{r+1}, \dots, p_n\}$

Let f "new" face,



$P_{confl}(f) \subseteq P_{confl}(f_1) \cup P_{confl}(f_2)$

because

$$H^+(f) \subseteq H^+(f_1) \cup H^+(f_2)$$

$\Rightarrow$ in (4): scan $P(f_1) \cup P(f_2)$

Algo as a whole:

find $p_1, \dots, p_4$ to form a tetrahedron

$\mathcal{C}_4 := CH(p_1, \dots, p_4)$

init $\mathcal{G}$: for all p_r, $r \geq 5$: for all $f \in \mathcal{C}_4$: if "conflict": establish edge in $\mathcal{G}$

for $r = 5, \dots, n$:

  if $F_{confl}(p_r) = \emptyset$:

    continue with next r

  for all $f \in F_{confl}(p_r)$:

    delete f from $\mathcal{C}_{r-1}$

    if one of edges e of f is on L:

      construct tri f' with p_r and e

      add f' to $\mathcal{C}_r$

      create node in $\mathcal{G}$ for f'

      for all $p \in P_{confl}(f_1) \cup P_{confl}(f_2)$,

        if f' is front facing w.r.t. p:

          add edge (p, f') to $\mathcal{G}$

Lemma:

The convex hull in 3D can be computed in expected time $O(n \log n)$.

Note: Clarkson-Shor has $O(n^3)$ worst-case time.

Sketch for proof:

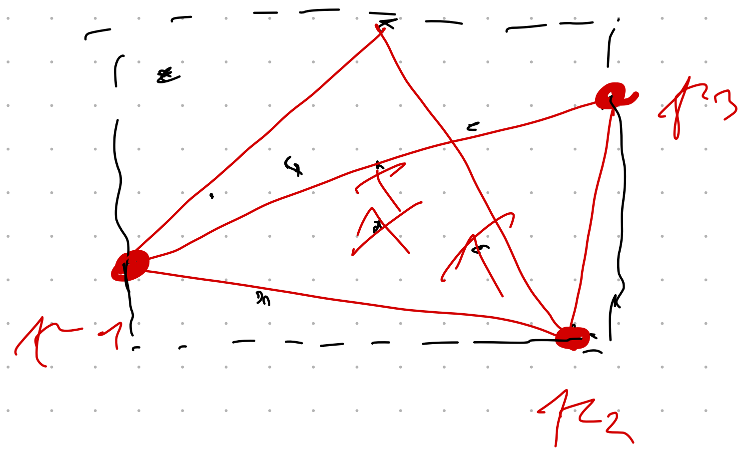
1. Show that # temporary faces $\leq 6n$

2. Show

$$\sum_{e \in L} |P_{confl}(f_1)| + |P_{confl}(f_2)| \in O(n \log n)$$

↑ all silhouette edges in total

Akl – Toussaint Heuristic (ATH)



Find p_1 with min x-coord,
" p_2 " " y-coord,
p_3 max x-coord

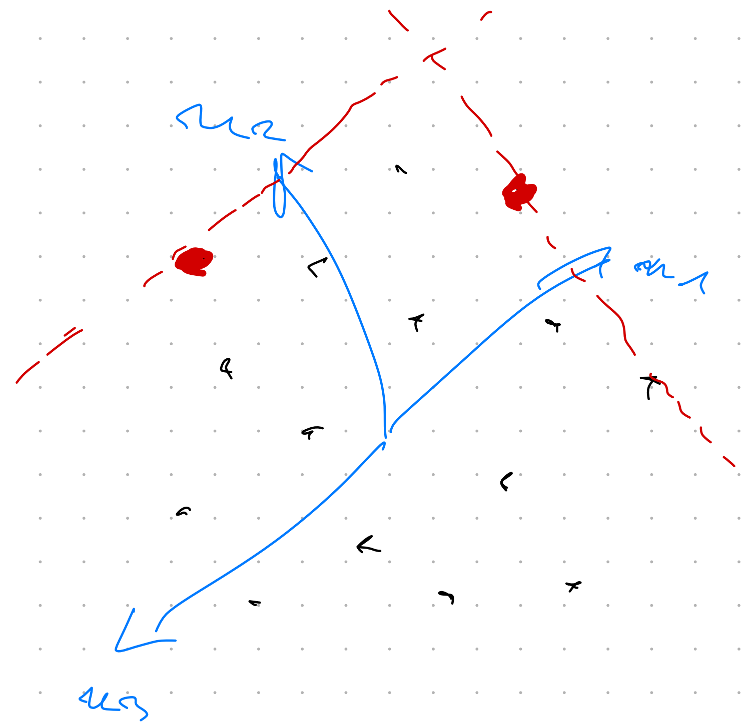
Clearly $p_1, p_2, p_3 \in CH$

$\Rightarrow$ delete all $p \in \Delta p_1 p_2 p_3$

3D: tetrahedron

Repeat with other 3 sides of bbox

Extend:



choose 3 random vectors $\vec{n}_i$ (in 3D: 4 vectors)

find $p_i^* = \max\limits_{j=1,\ldots,n} \{ \vec{n} \cdot p_j \}$

throw away all $p \in \Delta p_1 p_2 p_3$

repeat k times (or until no more pts get deleted)

Q: can turn this into a probabilistic algo for CH?