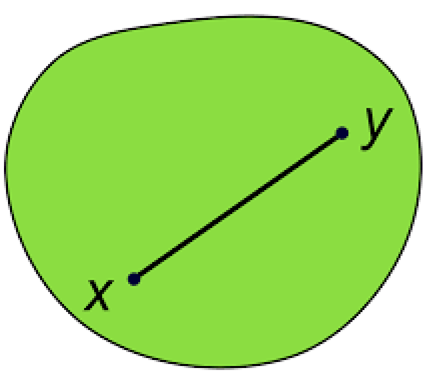
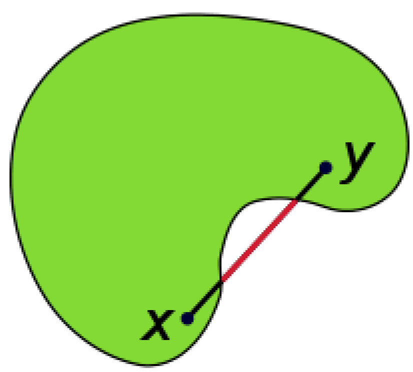
Convex Hulls

Convexity definition:

given a set $k \subseteq \mathbb{R}^d$, k is convex $\iff \forall p, q, \in k : \bar{pq} \in k$, i.e. when for each couple of points in k is possible to find a straight line connecting them fully contained in k



a convex set



a non convex set

Lemma:

if k_1, k_2 are both convex sets, then $k_1 \cap k_2$ is convex too

Convex Hull

the smallest convex set containing a subspace $S \subset \mathbb{R}^d$, $CH(S) = \bigcap_{k\ convex \supseteq S} k$. we will assume that S is finite and discrete.

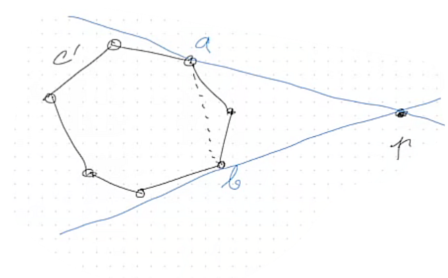
Tangent: given a polygon p in 2d and a point q outside of P, a line touching P in exactly one point and passing through q

Theorem: $CH(S)$ is a politope that has vertices in S

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

for $|S| = 3$, the CH(S) is the triangle made of the points, being the interception of the 3 halfspaces separated by the lines passing through 2 points

for $|S| > 3$, we choose a point $p \in S$, create $S' = S \setminus \{p\}$ and $C' = CH(S'), s.t. C' \subseteq S' \subseteq S$ and we construct $CH(S)$ with C' and p



- if $p \in C'$ we do nothing
- if $p \notin C'$ we find the tangents through p to C' a and b, then $CH(S) = C' \cup \triangle abp$

it follows that $\forall\ convex\ k \supseteq S : C' \subseteq k, \triangle abp \subseteq k \Rightarrow C' \cup \triangle abp \subseteq CH(S)$

we now need to prove that $\forall x \in C', y \in \triangle abp : \bar{xy} \subseteq C' \cup \triangle abp$

consider $y \in \triangle abp \setminus C'$

c is the interception point of $\bar{xy}$ with the board of C', c must be between the tangents, **it follows that** $\bar{cy} \subseteq \triangle abp, \bar{xc} \subseteq C' \Rightarrow \bar{xy} \subseteq C' \cup \triangle abp = C$

Properties

convex combination

given k points $\{p_1 \dots p_k\} = S \subseteq \mathbb{R}^d$, a **convex combination** $p = \sum_{i=1}^{k} \lambda_i p_i$, $\sum \lambda_i = 1, \lambda_i \geq 0$

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

lower bound of the construction algorithm

any construction of a $CH(S)$ has a complexity $\Omega(n \log n)$

"Rubberband" property

the border of $CH(S)$ in the plane is the shortest curve around S closed and simple

Construction in 2d

given a set S of n points in a plane we want to construct a $CH(S)$ by finding those points in S that appear as vertices of the CH in counter clockwise order

Assumption: no two points in S has the same y coordinate

Naive algorithm

```
C=empty set #candidates for S
for all pairs p, q : p!=q:
  for all r in S\{p, q}
    if r is left of the line pq:
      ignore p, q
      consider the next p, q
  add p, q to c
  sort c
```

but this algorithm is $O(n^3)$, that is not so fast

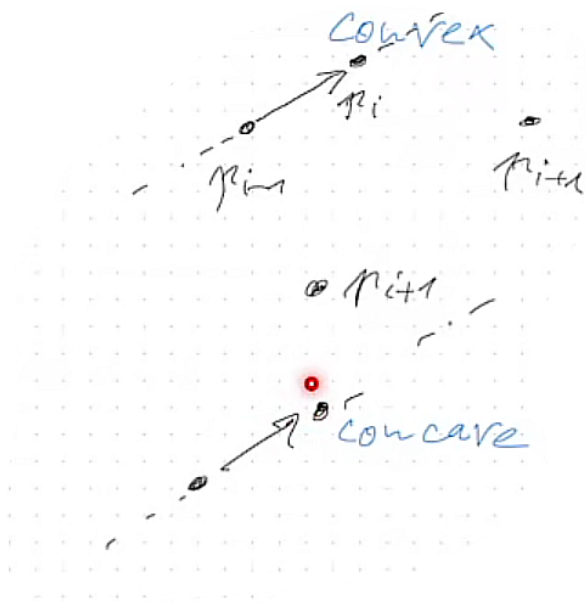
Graham's scan

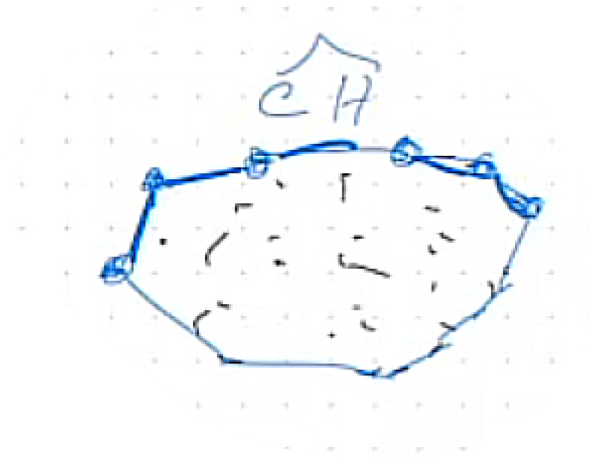
we can use a better approach: "incremental computation"

we add the points one by one

Upper hull: $\hat{C}H(s)$ **:** the sub-segment of $CH(S)$ with monotonically increasing x-coordinate

convex vertex: a vertex p_i to the right of the line passing through the previous 2 vertices p_{i-1}, p_{i-2}






```
sort s by x coordinate->p1...pn
c_hat = {p1, p2}
for i = 3 to n:
  c_hat+={pi}
  while |c_hat|>2 and pi-2, pi-1, pi in c_hat form a concave vertex:
    delete pi-1 from c_hat
analogous for the lower hull
```

with this algorithm we obtain a complexity of $O(n \log n)$, since the while is executet maximum n times, and since the complexity of the sorting is $O(n \log n)$ the complexity is the one of the sorting, if we use the **radix sort** we can also reach a $O(n)$

Gift wrapping(jarvis' march)

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

notation: $\angle|(\bar{pq}, x) =$ angle between $\bar{pq}$ and the x axis

construct the right-hand hull

p_0 si the point in S with the smallest y

p^* is the one with the top most y

$p_i = p_0$

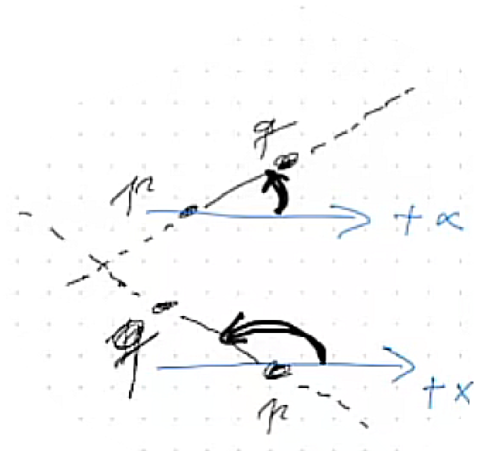
while $p_i \neq p^*$:

  delete p_i from S

  output p_i

  $p_{i+1} = \min_{p \in S} \angle|(\bar{p_i p}, x)$

output p^*



running time

let $h = |CH(S)| = \#points\ on\ CH$, the while has a complexity of $O(n)$, the initialization of $O(n \cdot h)$, and this becomes the overall complexity

the only problem is that the algorithm is pretty much output sensitive, reaching an $O(n^2)$

Chen’s algorithm

this algorithm uses a divide et impera structure, where create mini hulls and perform the gift wrapping.

```
HallG(S, m, h'):
input S={p1...pn}in R2
2>=m<=n
h'>=1
k=round(n/m)
partition S=S1 U...U Sk
for i 0 1...k:
  compute Ci 0 CH(Si)
p1 = latest point in S, output p1
for l = 1...h':
  for i 0 1...k:
    calculate  the lower tangent on ci through pi
    qi 0 points on ci an d tangents
  pi+1 0 find min q in {q1...qk}angle between piq, x
  output pi+1
  if pi+1 = top most point is S:
    output l+1 points
```

```
    return "coonvex hull complete"
return "convex hull incomplete"
```

NOTE: if $h' \geq h$ the if in the nested loop must happen

this time the complexity is $O(n \log m)$ for the first phase, while each tangent calculation takes $O(\log n)$ time, the loop is repeated $O(k \log n)$, bringing to a total running time of $O(h' \frac{n}{m} \log m + n \log m)$, if we choose $m = h'$ we obtain $O(n \log h')$

Complexity of a data structure

a function $f(n)$ representing the size of the output i.e. th ponts+the data structure , given n, the size of input i.e. the number of points

Euler's equation

lent Pbe a convec polyhedron, $v = |V_p|, e = |E_p|, f = |F_p|$(number of vertices, edges and faces), then $v - e + f = 2$

corollary: polyhedron convexity

for every convex polyhedron $v, e, f, \in \Theta(v, e, f)$

Convex Hull in 3D

a randomized and incremental algorithm

input: $S = \{p_1 \dots p_n\}$

start:

pick the first 2 points p_1, p_2

find a p_3 not in the ine of $p_1 \bar{p}_2$

find a p_4 not in the plane of p_1, p_2, p_3

$\Rightarrow$if you cannot find it go back to one dimension

take the tethraedron passing through p_1, p_2, p_3, p_4: C_4

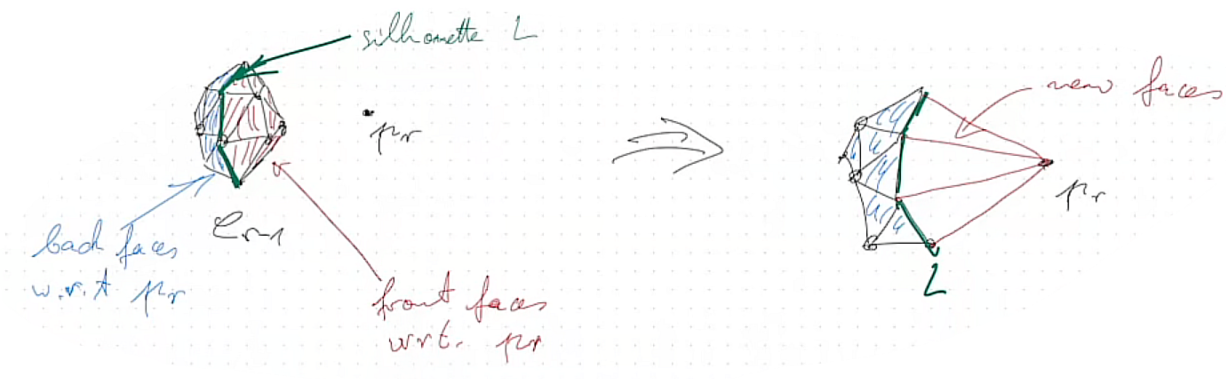
permute $p_5 \dots p_n$ randomly

for $p_r = p_5 \dots p_n$:

combine C_{r-1} and p_r to get C_r, the convex hull over $\{p_1, p_2, p_3, p_4\}$

case 1: $p_r \in C_{r-1}$ $\rightarrow$dischard p_r

case 2: $p_r \notin C_{r-1}$ $\rightarrow$



we need to find the front faces, the brute force method will lead to a $O(n^2)$, so we need to improve

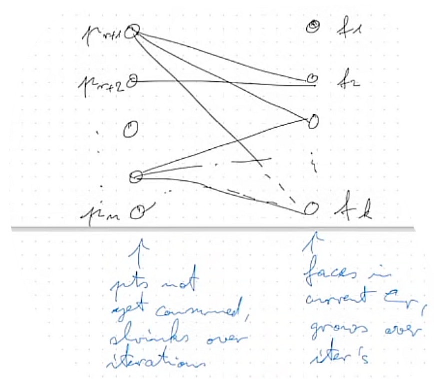
Conflict graph

we maintain 2 conflict sets, for all faces $f \in C_r$:

$P_{conf}(f)$:all the points that can see f(so is a front face)

$F_{conf}(p_s)$:all the faces that can see p_f(so is a front face)

we say that $p \in P_{conf}(f)$ is in conflict with f. We now need an auxiliary data structure, a **Bipartite Graph** G**,** in which we will connect points and faces



$\forall f \in F_{conf}(p_r)$:

delete f from C_{r-1}

if one of f's edges is on L:

construct a triangle f' with p_r and edge add f' to C_r

the complexity is $O(|F_{conf}(p_r)|)$

Initialization of G

start with C_4

test all $p_5 \dots p_n$ against the 4 faces of C_4

Updating: $C_{r-1} \rightarrow C_r$

delete all neighbors of p_r in G

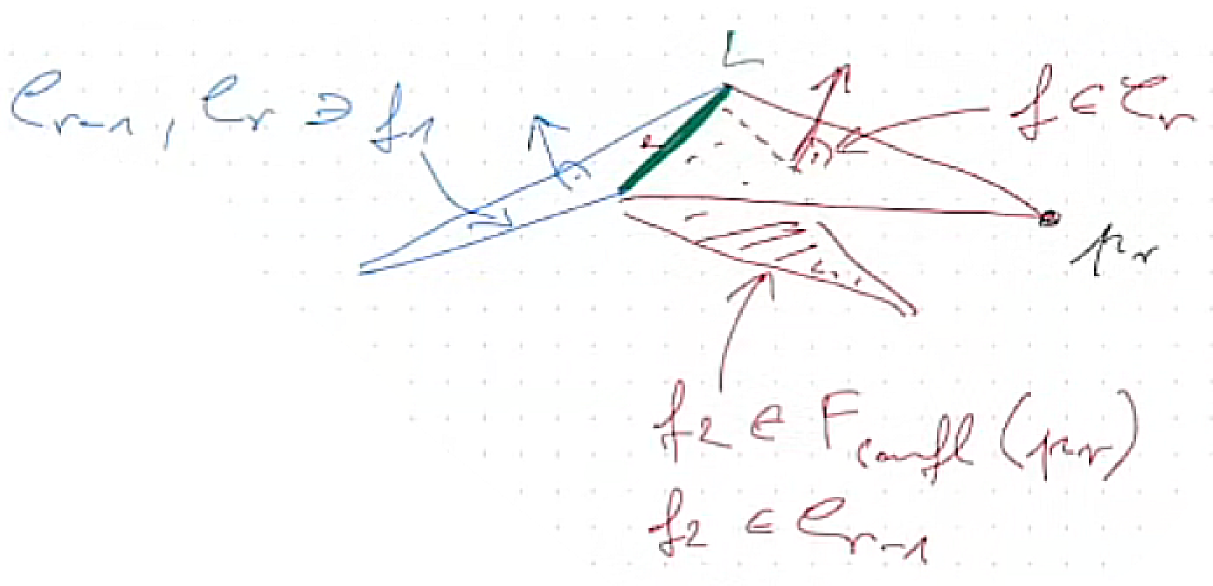
delete p_r from G

create new nodes for the new faces

create new edges for conflicts

NOTE: if $f \in C_{r-1} and f \in C_r then f \notin P_{confl}(f)$ then $P_{conf}(f)$ stays the same, so we only compute $P_{confl}(f)$ for new f's

let f be a "new" face, $P_{confl}(f) \subseteq P_{confl}(f_1) \cup P_{confl}(f_2)$, since the **positive halfspace** $H^+(f) \subseteq H^+(f_1) \cup H^+(f_2)$ **and we scan** $P(f_1), P(f_2)$



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 could be computed in expected time $O(n \log n)$ with a worst case of $O(n^3)$

AKL-Toussaint Heuristic

Find a point $p_1$with minimal x-coordinate

Find a point $p_2$with minimal y-coordinate

Find a point $p_3$with maximal x-coordinate

all 3 $\in CH$

delete all $p \in \triangle p_1, p_2, p_3$(in 3d we need another couple of mi x for z and we have a tethraedron),

repeat for all the sides of the bounding box

Extension

choose 3 random vectors(4 in 3D)

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

delete all $p \in \triangle p_1, p_2, p_3$

repeat k times

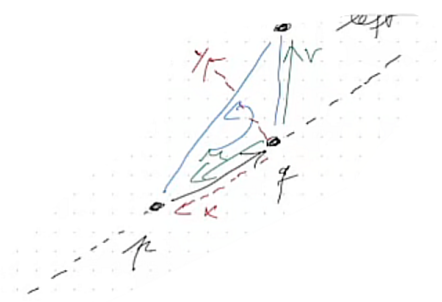
Geometric predicates

left/right

we have 3 points $p, q, r \in \mathbb{R}^2$, is r left of $\bar{pq}$?

immagining to connect the three points in a triangle and spin counterclockwise

$“left” \iff Area(\triangle pqr) > 0$, where $Area(\triangle pqr) = \frac{1}{2} \det \begin{bmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ r_x & r_y & 1 \end{bmatrix}$



orientation in 3d

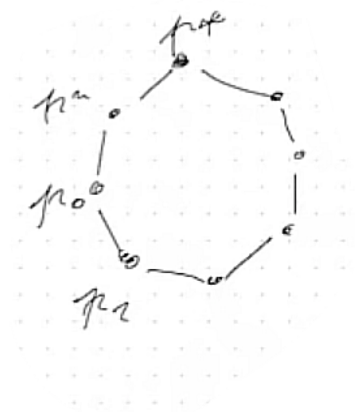
the same as above could be done in 3 dimensions, adding a point and working with a tetraedron

we have 4 points $p, q, r, s \in \mathbb{R}^3$, is s in front ofof $p\bar{q}r$? if $volume(p, q, r, s) > 0$, where

$$volume(p, q, r, s) = \frac{1}{6} \det \begin{bmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \\ r_x & r_y & r_z & 1 \\ s_x & s_y & s_z & 1 \end{bmatrix}$$

finding extreme points of polygon

Given a convex polygon $p^0, \ldots, p^n$ we want to determine the point p^* with maximum p^*_y



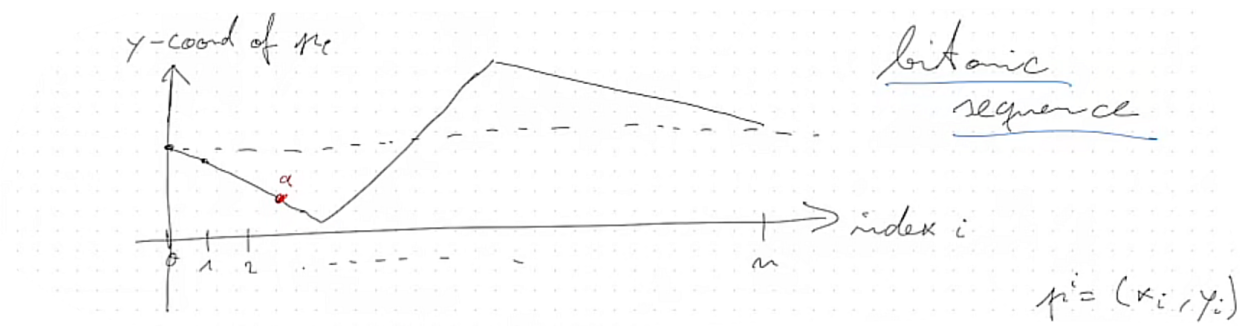
start with a=0m b=n

assume $p^*$is between $p_a, p_b \rightarrow c = \frac{1}{2}(a+b)$

case 1: $y_c < y_a < y_b$ we have a new bracket $a' = c, b' = b$ **case 2:** $y_a < y_c < y_b$ we have a new bracket $a' = c, b' = b$

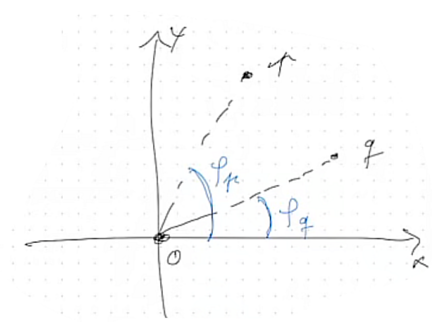
$\vdots$

in time $O(\log n)$



sort p, q by polar angle

given two points p, q in polar coordinates is $\phi_p > \phi_q$?



solution: $\iff area(\triangle oqp) > 0$