Outline

1. Position and orientation
2. Polygon and polytope
3. Convex hull
4. Problem solving strategies in computation geometry

Position and orientation

- Foundamental question: would two lines intersect in 2D / 3D ?

  proper intersection: two line segments have unique intersection not at the ends.

Examples of line segments do not intersect properly.

- Analytic solution and issues. We need to write in general solution

$$Ax + By + C = 0$$

to avoid slope $= \infty$ issues. Then the system of equations $\begin{cases} Ax + By + C = 0 \\ Dx + Ey + F = 0 \end{cases}$

  (1) No solution → no intersection between these two lines

  (2) ∞ solution → not proper intersection between these two lines

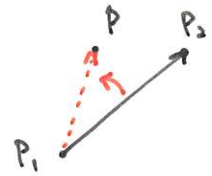
  (3) unique solution → determine if the solution occur at the ends

  → Floating point errors! That's why we need computational geometry!

- Problem Reduction
  - If the intersection does not happen on the ends, then the ends of one line must be located at different sides of the other.
  - Vice versa. So two lines intersect properly if and only if this condition holds.

- How do we determine if the ends of a line segment are located on different sides of another?

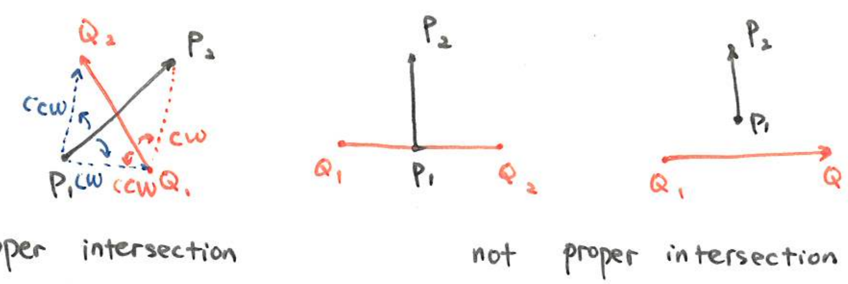
  We define the direction of a line starting from the first end and ending at the second, denoted $\overrightarrow{P_1P_2}$. Now we want to determine given a point P, it locates on the right side of $\overrightarrow{P_1P_2}$ or the left side (in relative orientation).



This reduces the problem to determing $\overrightarrow{PP_1}$ is CCW or CW from $\overrightarrow{P_1P_2}$.

- Solution: For two line segments P_1P_2 and Q_1Q_2, if Q_1 and Q_2 are located on different sides of $\overrightarrow{P_1P_2}$ and P_1 and P_2 are located on different sides of $\overrightarrow{Q_1Q_2}$, then P_1P_2 and Q_1Q_2 intersect.

  Examples:

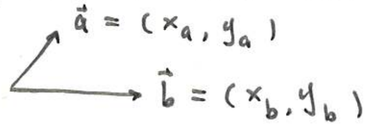


- How do we determine if $\overrightarrow{P_1P_2}$ to $\overrightarrow{P_1Q_1}$ is CCW or CW?

Cross product and dot product.

Goal: Find out $\vec{a} = \overrightarrow{P_1P_2}$ to $\vec{b} = \overrightarrow{P_1Q_2}$ is CW or CCW.

In the first quadrant, this can be done by comparing slope:



$$\frac{y_a}{x_a} - \frac{y_b}{x_b} > 0 \Rightarrow \vec{a} \text{ to } \vec{b} \text{ CW}.$$

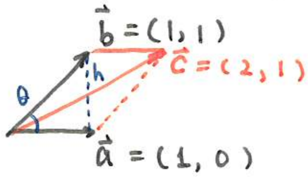
But this can't be extended to other quadrants, division is expensive and $x_a, x_b \neq 0$.

Solution: This motivates the use of cross product: we find that

$$\vec{a} \times \vec{b} := \begin{vmatrix} x_a & y_a \\ x_b & y_b \end{vmatrix} = x_a y_b - x_b y_a \quad \begin{cases} >0 & \text{if } \vec{a} \text{ to } \vec{b} \text{ CCW} \\ <0 & \text{if } \vec{a} \text{ to } \vec{b} \text{ CW} \end{cases}$$

Signed area.

Set $\vec{c} = \vec{a} + \vec{b}$, then the area of the parallelogram oacb is twice that of oab. The signed area of oab is positive if $\vec{a}$ to $\vec{b}$ is CCW, and negative if otherwise



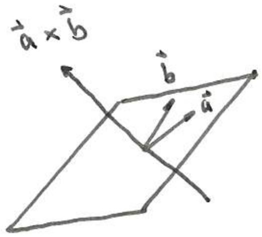
$$\vec{a} \times \vec{b} = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} = 1 = S_{oacb} \quad (\text{CCW})$$

if we set $\vec{a} = (1,1)$, $\vec{b} = (1,0)$, $\vec{a} \times \vec{b} = -1$ (CW)

Note $S_{oacb} = |\vec{a}| \cdot h = |\vec{a}| \cdot \sin\theta\, |\vec{b}| \Rightarrow \vec{a} \times \vec{b} = |\vec{a}||\vec{b}| \sin\theta$, $\theta \in [0, \pi)$

Geometric interpretation of cross product in 3D

$\vec{a} \times \vec{b}$ is perpendicular to the plane spanned by $\vec{a}$ and $\vec{b}$.



$$\vec{a} \times \vec{b} = \begin{vmatrix} i & j & k \\ x_a & y_a & z_a \\ x_b & y_b & z_b \end{vmatrix} = (y_a z_b - y_b z_a) i - (x_b z_a - x_a z_b) j + (x_a y_b - x_b y_a) k$$

→ The i direction of $\vec{a} \times \vec{b}$ signifies the cross product of $\mathrm{Proj}_{y,z} \vec{a}$ and $\mathrm{Proj}_{y,z} \vec{b}$, i.e., the projection of $\vec{a}$ and $\vec{b}$ onto the yoz plane

→ The j direction of $\vec{a} \times \vec{b}$ signifies $\mathrm{Proj}_{x,z} \vec{a} \times \mathrm{Proj}_{x,z} \vec{b}$

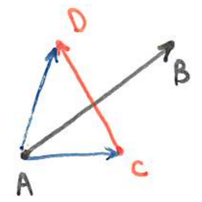
→ The k direction of $\vec{a} \times \vec{b}$ signifies $\mathrm{Proj}_{x,y} \vec{a} \times \mathrm{Proj}_{x,y} \vec{b}$

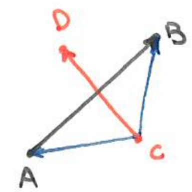
Summary: The proper intersection of line segments AB and CD

$\Leftrightarrow$ $(\vec{AB} \times \vec{AC})_z \cdot (\vec{AB} \times \vec{AD})_z < 0$ and $(\vec{CD} \times \vec{CA})_z \cdot (\vec{CD} \times \vec{CB})_z < 0$

in the z direction

$A, B, C, D \in xoy$ plane.



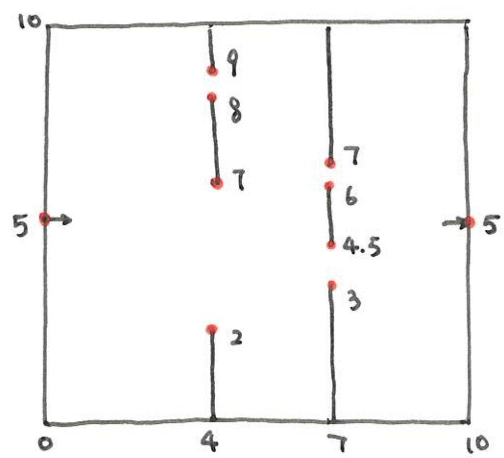


From an area view point:

$$\text{Area2}(A,B,C) := 2\,\text{Area}(A,B,C) = (\vec{AB}\times\vec{AC})_z = \begin{vmatrix} x_b - x_a & y_b - y_a \\ x_c - x_a & y_c - y_a \end{vmatrix}.$$

Floating Point Errors: treat ε-neighborhood of 0 to be 0, i.e., 10^{-6}.

Ex (UVA 393, The Doors) The room is bounded by $x=0,10$, $y=0,10$. Starting at (0.5), ending at $(10,5)$. N walls $(0 \le N \le 18)$, 2 doors on each wall with interval given for the doors. All input are integers, output shortest path.



Solution. Construct an undirected graph $V = \{\text{starting}, \text{doors\_up}, \text{doors\_down}, \text{ending}\}$

$E = \{$Euclidean distance between each pair of vertics if does not intersect walls.$\}$

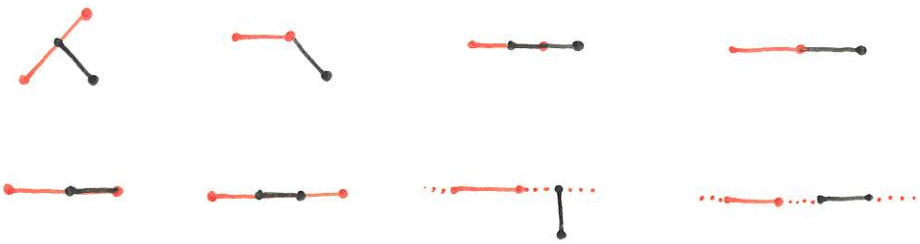
Algorithm: Dijkstra

Improper Intersection

First recall proper intersection in area form:

$$\underbrace{(\vec{AB}\times\vec{AC})_z}_{\Updownarrow\ 2\text{Area}(A,B,C)}\ \underbrace{(\vec{AB}\times\vec{AD})_z}_{\Updownarrow\ 2\text{Area}(A,B,D)} < 0 \quad \text{and} \quad \underbrace{(\vec{CD}\times\vec{CA})_z}_{\Updownarrow\ 2\text{Area}(C,D,A)}\ \underbrace{(\vec{CD}\times\vec{CB})_z}_{\Updownarrow\ 2\text{Area}(C,D,B)} < 0$$

Two line segments that do not intersect properly



(no intersection) (no intersection)

We note that for proper intersection,

$$(\vec{AB} \times \vec{AC})_z,\quad (\vec{AB} \times \vec{AD})_z,\quad (\vec{CD} \times \vec{CA})_z,\quad (\vec{CD} \times \vec{CB})_z \neq 0$$

At least one of the above cross products are zero for improper intersection as well as does not intersect.

- We can determine if it is improper intersection by determing if at least one end point of one segment lies on the other.
- Goal: find a function such that

$= -1$ if 

i.e., $C \in AB$ and $C \neq A$ and $C \neq B$

$= 0$ if 

i.e., $C = A$ or $C = B$

$= 1$ if 

i.e., $C \notin AB$

- To achieve that, we introduce Dot Product:

$$\vec{A} \cdot \vec{B} = \sum_{i=1}^{n} x_i x_i', \quad \vec{A} = (x_1, \ldots, x_n),\ \vec{B} = (x_1', \ldots, x_n').$$

Remark: the angle between two vectors $\in [0, \pi]$, where sine does not has inverse function but cosins has. Given

$$\vec{A} \times \vec{B} = |\vec{A}||\vec{B}| \sin\theta, \quad \vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}| \cos\theta$$

we have

$$\theta = \cos^{-1} \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|}.$$

- Use dot product to determine position of three points on a line

$\vec{CA} \cdot \vec{CB} = 0$ if $C = A$ or $C = B$

$\vec{CA} \cdot \vec{CB} < 0$ if $C \in AB$

$\vec{CA} \cdot \vec{CB} > 0$ if $C \notin AB$

Coordinate Comparison. Given A, B, C on the same line, we only need to consider one coordinate. To avoid floating point errors, we choose the dimension with greater difference, i.e.,

if $|a.y - b.y| > |a.x - b.x|$, then we investigate if

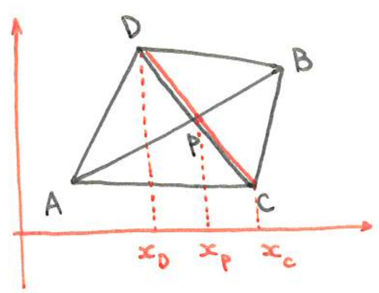
$$c.y \in [\min(a.y, b.y), \max(a.y, b.y)]$$

Assuming the absolute difference in the y-direction is the largest, we compute

$$\underbrace{\text{dblcmp}}_{\text{returns zero for floating point } \varepsilon}(c.y - \min(a.y, b.y)) \times \text{dblcmp}(c.y - \max(a.y, b.y))$$

if			prod
$c < \min$	-1	-1	outside → 1
$c = \min$	0	-1	0
$\min < c < \max$	1	-1	inside → -1
$c = \max$	1	0	0
$c > \max$	1	1	outside → 1

Intersection of line segments



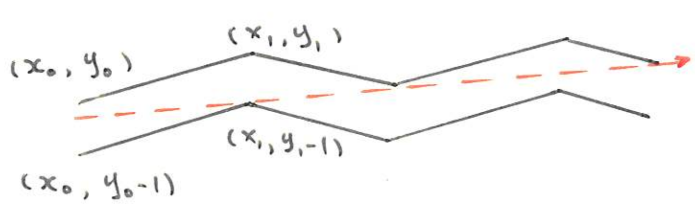
$$\frac{|DP|}{|CP|} = \frac{S_{\triangle ADB}}{S_{\triangle ACB}} = \frac{|\vec{AD} \times \vec{AB}|}{|\vec{AC} \times \vec{AB}|}$$

(note Area2(A, B, D) × Area2(A, B, C) < 0 for proper intersection.)

$$\frac{|DP|}{|CP|} = \frac{x_P - x_D}{x_C - x_P} = \frac{S_{\triangle ADB}}{S_{\triangle ACB}} \Rightarrow x_P S_{\triangle ACB} - x_D S_{\triangle ACB} = x_C S_{\triangle ADB} - x_P S_{\triangle ADB}$$

$$\Rightarrow x_P = \frac{x_C S_{\triangle ADB} + x_D S_{\triangle ACB}}{S_{\triangle ACB} + S_{\triangle ADB}}$$

Ex (UVA 303, Pipe).



A pipe with vertices $(x_0, y_0), \ldots, (x_n, y_n)$ and $(x_0, y_0-1), \ldots, (x_n, y_n-1)$. What is the longest distance a ray can travel?

- Solution.

1. If a ray never touch the vertices, it is not optimal.
   (We can always translate the ray to elongate it.)

2. If a ray only touches one vertex, it is not optimal.
   (We can rotate the ray to improve it.)

3. Optimal rays touch a vertex from above and one vertex from below.

4. Algorithm. → Iterate all pairs of (upper_vertex, lower_vertex) that forms a line ℓ.
   → Check if ℓ falls in (x_0, y_0) and $(x_0, y_0 - 1)$, if so, the ray is permissible.
   → Check if ℓ intersects with either wall, and record the nearest intersection.
   → If never intersect, then the line ℓ goes through the pipe!

※. This algorithm can be improved.

Polygon and polytope

- Polygon : in 2D, closed chain formed by consecutive line segments. More precisely, set $P_0, \dots, P_{n-1} \in \mathbb{R}^2$, $E_0 = P_0P_1, \cdots, E_{n-1} = P_{n-1}P_0$ are n line segments. These n line segments form a polygon iff

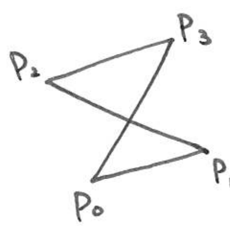
  any two consecutive line segments E_i and E_{i+1} $(0 \le i \le n-1)$ share precisely one common point, P_{i+1}.

- Complex polygon: If a polygon does not satisfy

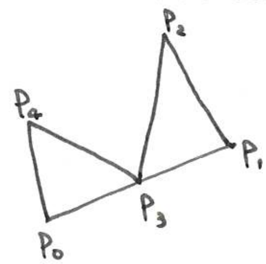
$$\forall j \neq i+1, \ E_i \cap E_j = \phi.$$

  it is called a complex polygon, otherwise simple polygon.

- No-fit polygon (NFP): NFP is a special type of complex polygon that touches another edge but does not intersect.

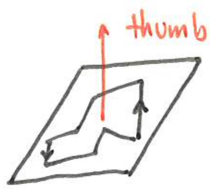


complex polygon



no-fit polygon

- Right-hand rule specifies that the positive direction of a polygon is defined as the direction that when walking alone it using four fingers of the right hand, the thumb pointing upwards. The interior of the polygon is defined as the left hand side.



- Jordan Curve Theorem (Differential Topology, Guillemin & Pollack). Every simple closed curve in $\mathbb{R}^2$ divides the plane into two pieces, the "inside" and "outside" of the curve. (simple: non-self-intersection.)

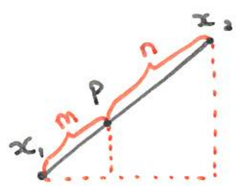
- **Convex Polygon:** For any edge e, the polygon lies on one side of the edge; it is on the left side if the edge is positively oriented.

- **Convexity of arbitrary dimension**

Definition 1. A region D is convex if and only if $\forall x_1, x_2 \in D, \frac{x_1+x_2}{2} \in D$

Definition 2. A region D is convex

$$\iff \forall x_1, x_2 \in D, \forall \lambda \in [0,1], \lambda x_1 + (1-\lambda) x_2 \in D$$



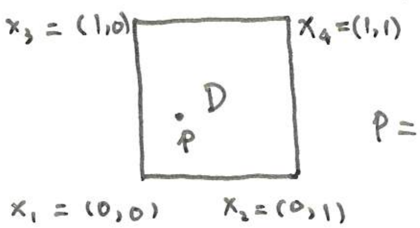
$P = (e, f)$, $x_1 = (a, b)$, $x_2 = (c, d)$

$$\lambda = \frac{m}{n} \Rightarrow \frac{m}{m+n} = \frac{e-a}{c-a} = \frac{f-b}{d-b}$$

$$\Rightarrow e = \frac{mc+an}{m+c}, \quad f = \frac{md+bn}{m+n}$$

Definition 3. A region D is convex

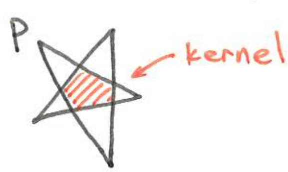
$$\iff \forall i,\ 1 \le i \le n,\ \forall \lambda_i \ge 0,\ \sum_i \lambda_i = 1,\ \text{have } \sum_i \lambda_i x_i \in D.$$



$\lambda_1 = \frac{1}{2}$, $\lambda_2 = \frac{1}{4}$, $\lambda_3 = \frac{1}{8}$, $\lambda_4 = \frac{1}{8}$

$$P = (0,0) + (0, \tfrac{1}{4}) + (\tfrac{1}{8}, 0) + (\tfrac{1}{8}, \tfrac{1}{8}) = (\tfrac{1}{4}, \tfrac{3}{8})$$

- **Star Polygon.** Exists **one point** inside the polygon that connects to any other point inside the polygon without intersecting the boundary. The collection of such points is called the **kernel.**



A star polygon and its kernel.

$$\mathcal{K}(P) := \{ x \in P \mid \forall y \in P : (xy \subseteq P) \land ((xy \cap \partial P) \subseteq \{x, y\}) \}$$

$P = \overline{P} = \mathring{P} \cup \partial P$, P containes the boundary, i.e., is the union of its interior $\mathring{P}$ and the boundary ∂P.

Museum Problem. In an art gallery, what is the minimum number of guards who together can observe the gallery (see all walls)?

- Is one guard enough? If so, where should the guard be located?

- **Reduced problem:** walls are horizontal or vertical (y-axis direction doesn't matter)

Algorithm: we squeeze it to find the rectangle kernel.

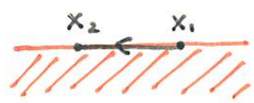
Initialization: Set $x_{min} = y_{min} = -\infty$, $y_{max} = x_{max} = +\infty$.

Assuming path is positively oriented, i.e., the interior is on the left, then

$(x_1, y) \to (x_2, y)$ $\quad x_1 < x_2$

$y_{max} = \min(y_{max}, y)$

$(x_1, y) \to (x_2, y)$ $\quad x_1 > x_2$



$y_{min} = \max(y_{min}, y)$

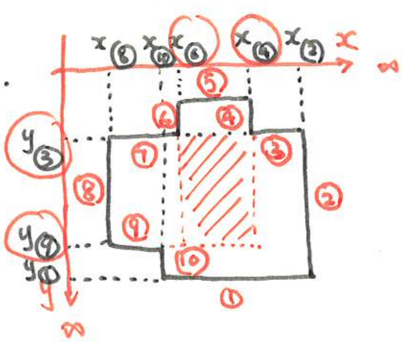
$(x, y_1) \to (x, y_2)$ $\quad y_1 < y_2$

$x_{min} = \max(x_{min}, x)$

$(x, y_1) \to (x, y_2)$ $\quad y_1 > y_2$

$x_{max} = \min(x_{max}, x)$

Eg.

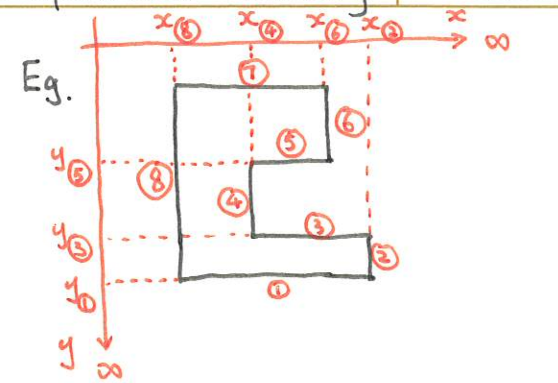


$x \in [x_6, x_4]$

$y \in [y_3, y_9]$

Initialization	y_{max}^{∞}	$y_{min}^{-\infty}$	x_{max}^{∞}	$x_{min}^{-\infty}$
Step 1	y_1			
Step 2			x_2	
Step 3		y_3		
Step 4			x_4	
Step 5		y_3		
Step 6				x_6
Step 7		y_3		
Step 8				x_8
Step 9	y_9			
Step 10				x_6

Eg.



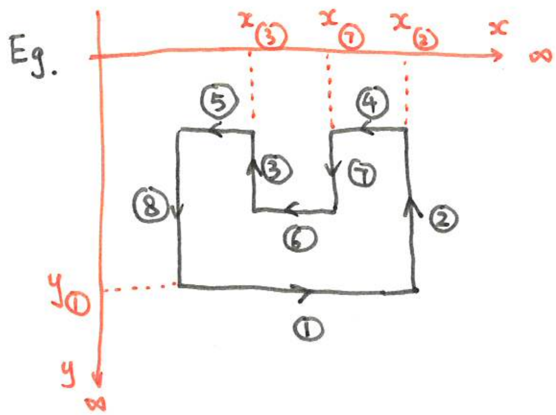
Initialization	y_{max}^{∞}	$y_{min}^{-\infty}$	x_{max}^{∞}	$x_{min}^{-\infty}$
step 1	y_1			
step 2			x_2	
step 3		y_3		
step 4			x_4	
step 5	y_5			
step 6			x_4	
step 7		y_3		
step 8				x_8

$x \in [x_8, x_4]$

$y \in [y_5, y_3] \in \phi$

as expected.

Eg.



$\rightarrow$: $y_{max} = y_1$

$\uparrow$: $x_{max} = x_2, x_3$

$\leftarrow$: $y_{min} = y_4, y_6$

$\downarrow$: $x_{min} = x_7$

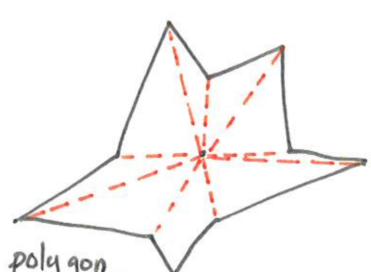
$x \in [x_7, x_3] \in \phi$ as expected.

- Problem Equivalence.

A. Existence of one interior point seeing all boundary points

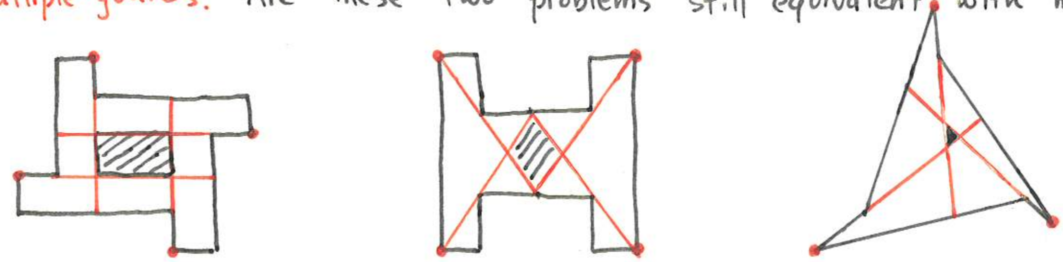
B. Existence of one interior point seeing all interior points (i.e., a star polygon)

A $\Rightarrow$ B via triangulation: if a guard can see all points on each line segment on the boundary, triangulate the polygon by connecting this interior point with all vertices, then the guard can see all points in each triangle and thus can see the entire interior of the polygon.



B $\Rightarrow$ A This direction can be assumed.

- Multiple guards. Are these two problems still equivalent with multiple guards?



Guards can see all the boundard but not the shaded area.

Polyhedron

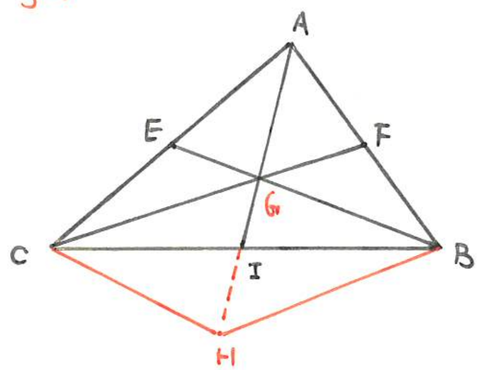
- A mathematical definition. A polyhedron is a finite, connected set of plane polygons, such that every side of each polygon belong to precisely another polygon, with the proviso that the polygons surrounding each vertex form a single circuit. Regular Polytopes, HSM Coxeter
- Convex polyhedron. The polyhedron lies on one side of each face, the "inside" - the inner normal direction.
- Star polyhedron. Exist a point inside the polyhedron that can see all points inside the polyhedron and on the boundary.
- Kernel of polygon and polydedron. Intersection of half-spaces
- Representation of polyhedron

  Point - Surface: Represent polyhedral as a collection of surfaces, and each surface is represented by a collection of ordered points that are positively oriented (right-hand rule has thumb pointing outside of the polyhedron)
  - points can be represented as indices
  - edges are stored by two faces
  - no adjacency information for faces and edges.

  Point - Edge - Surface:
  - Each edge is directed, i.e., faces defined by directed edges
  - Each edge specifies the "left face" and the "right face"
  - Right-hand Rule: polyhedron normal → positive edge → left face inner normal

Volumn and Centroid for Polygon, Polyhedron and Generalizations

- **Cross product:** direction and magnitude (signed area)
- **Area of triangle:** $\text{Area}(A, B, C) = \frac{1}{2} \vec{AB} \times \vec{AC} = \frac{1}{2} \begin{vmatrix} x_B - x_A & y_B - y_A \\ x_C - x_A & y_C - y_A \end{vmatrix}$.
- **Centroid of triangle:** the centroid of a triangle distributes all medians in $2:1$.



proof. Let E, F being the center of AC and AB. Connect CF and BE, denote their intersection G. Extend AG to H such that $AG = GH$, and connect CH and BH.

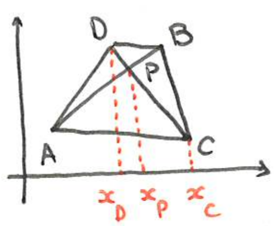
G, F are midpoints of AH and $AB \Rightarrow GF \parallel BH$, i.e., $CG \parallel BH$

E, G are midpoints of AC and $AH \Rightarrow EG \parallel CH$, i.e., $BG \parallel CH$

So $CHBG$ is a parallelogram thus CB and GH bisect each other.

Thus AI is a median, and $GI = \frac{1}{2} GH = \frac{1}{2} AG$. Same for GE, GF. □

Therefore, given $F = \left(\frac{x_A + x_B}{2}, \frac{y_A + y_B}{2}\right)$, then recall



$$\frac{|DP|}{|PC|} = \frac{x_P - x_D}{x_C - x_P} \Rightarrow x_P = \frac{x_C |DP| + x_D |CP|}{|DP| + |CP|}$$

$$|DP| x_C - |DP| x_P = |PC| x_P - |PC| x_D$$

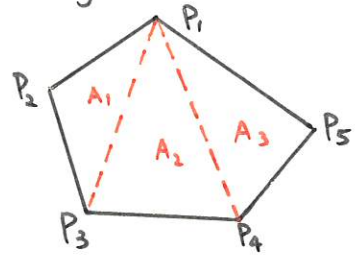
So in the case of centroid, $G = \left(\frac{x_A + x_B + x_C}{3}, \frac{y_A + y_B + y_C}{3}\right)$.

$$\frac{|CG|}{|GF|} = \frac{x_G - x_C}{x_F - x_G} \Rightarrow \frac{2}{1} = \frac{x_G - x_C}{\frac{x_A + x_B}{2} - x_G}$$

$$\Rightarrow x_A + x_B - 2x_G = x_G - x_C \text{, same for the } y\text{-axis.}$$

Triangulation of Convex Polygon.

- An n-polygon can be triangulated into $n-2$ triangles all contained inside of the polygon if it is convex.



$$2A(P_1 P_{i+1} P_{i+2}) = \begin{vmatrix} x_{i+1}-x_1 & y_{i+1}-y_1 \\ x_{i+2}-x_{i+1} & y_{i+2}-y_{i+1} \end{vmatrix}$$

$$= x_{i+1}y_{i+2} - \cancel{x_{i+1}y_{i+1}} - \cancel{x_1 y_{i+2}} + x_1 y_{i+1} - x_{i+2}y_{i+1} + \cancel{x_{i+2}y_1} + \cancel{x_{i+1}y_{i+1}} - x_{i+1}y_1$$

$$2A(P_1 P_{i+2} P_{i+3}) = \begin{vmatrix} x_{i+2}-x_1 & y_{i+2}-y_1 \\ x_{i+3}-x_{i+2} & y_{i+3}-y_{i+2} \end{vmatrix}$$

$$= x_{i+2}y_{i+3} - \cancel{x_{i+2}y_{i+2}} - x_1 y_{i+3} + \cancel{x_1 y_{i+2}} - x_{i+3}y_{i+2} + x_{i+3}y_1 + \cancel{x_{i+2}y_{i+2}} - \cancel{x_{i+2}y_1}$$

$$2A(P_1 P_{i+1} P_{i+2}) + 2A(P_1 P_{i+2} P_{i+3}) = \underbrace{\begin{vmatrix} x_1 & y_1 \\ x_{i+1} & y_{i+1} \end{vmatrix}} + \begin{vmatrix} x_{i+1} & y_{i+1} \\ x_{i+2} & y_{i+2} \end{vmatrix} + \underbrace{\begin{vmatrix} x_1 & y_1 \\ x_{i+3} & y_{i+3} \end{vmatrix}} + \begin{vmatrix} x_{i+2} & y_{i+2} \\ x_{i+3} & y_{i+3} \end{vmatrix}$$

For the starting term, $i=1$, this is $\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix}$

will be canceled out by $A(P_1 P_{i+3} P_{i+4})$

$$2A(P_1 P_{i+3} P_{i+4}) = \begin{vmatrix} x_{i+3}-x_1 & y_{i+3}-y_1 \\ x_{i+4}-x_{i+3} & y_{i+4}-y_{i+3} \end{vmatrix} = x_{i+3}y_{i+4} - x_1 y_{i+4} + \underbrace{x_1 y_{i+3}} - x_{i+4}y_{i+3} + x_{i+4}y_1 \underbrace{- x_{i+3}y_1}$$

will be canceled out by $A(P_1 P_{i+4} P_{i+5})$

Terminating terms:

cancel out $A(P_1 P_{n-3} P_{n-2})$

$$2A(P_1 P_{n-2} P_{n-1}) = \begin{vmatrix} x_{n-2}-x_1 & y_{n-2}-y_1 \\ x_{n-1}-x_{n-2} & y_{n-1}-y_{n-2} \end{vmatrix} = x_{n-2}y_{n-1} - x_1 y_{n-1} + \underbrace{x_1 y_{n-2}} - x_{n-1}y_{n-2} + x_{n-1}y_1 \underbrace{- x_{n-2}y_1}$$

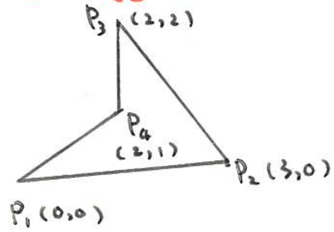
$$2A(P_1 P_{n-1} P_n) = \begin{vmatrix} x_{n-1}-x_1 & y_{n-1}-y_1 \\ x_n - x_{n-1} & y_n - y_{n-1} \end{vmatrix} = x_{n-1}y_n - x_1 y_n + x_1 y_{n-1} - x_n y_{n-1} + x_n y_1 - x_{n-1}y_1$$

$$A = \sum_{i=1}^{n-2} A(P_1 P_{i+1} P_{i+2}) = \frac{1}{2}\left(\begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} + \sum_{i=1}^{n-2}\left(\begin{vmatrix} x_{i+1} & y_{i+1} \\ x_{i+2} & y_{i+2} \end{vmatrix} \right) + \begin{vmatrix} x_n & y_n \\ x_{n+1} & y_{n+1} \end{vmatrix} \right)$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n-1}\left(\begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix} \right) + \begin{vmatrix} x_n & y_n \\ x_{n+1} & y_{n+1} \end{vmatrix} \right) = \frac{1}{2}\sum_{i=1}^{n} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}. \quad \square$$

Thus, we've proved the Shoelace formula, or also called Gauss' area formula: Given a planar simple polygon with positively oriented (CCW) sequence of points $P_i = (x_i, y_i)$, $i = 1, \dots, n$ in Cartisian coordinat system, with $P_0 = P_n$, $P_1 = P_{n+1}$, the triangle formula gives $A = \frac{1}{2}\sum_{i=1}^{n} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}$.

Concave Polygon.

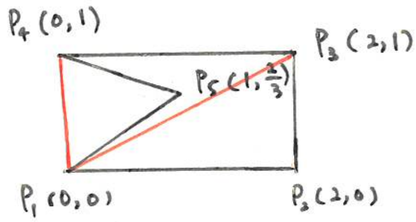


$$A_{P_1P_2P_3P_4} = A_{P_1P_2P_3} - A_{P_1P_3P_4} = S_{P_1P_2P_3} + S_{P_1P_3P_4}$$

A : Area. S : Signed area. $= \sum_{i=1}^{N} \frac{1}{2} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}$

$$A_{P_1P_2P_3P_4} = \frac{1}{2}\begin{vmatrix} 0 & 0 \\ 3 & 0 \end{vmatrix} + \frac{1}{2}\begin{vmatrix} 3 & 0 \\ 2 & 2 \end{vmatrix} + \frac{1}{2}\begin{vmatrix} 2 & 2 \\ 2 & 1 \end{vmatrix} + \frac{1}{2}\begin{vmatrix} 2 & 1 \\ 0 & 0 \end{vmatrix}$$

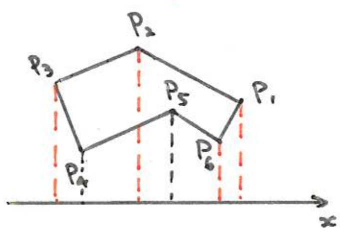
$$= 0 + 3 + (-1) + 0 = 2 = A_{P_1P_2P_4} + A_{P_2P_3P_4}$$



$$A_{P_1P_2P_3P_4P_5} = \frac{1}{2}\left(\begin{vmatrix} 0 & 0 \\ 2 & 0 \end{vmatrix} + \begin{vmatrix} 2 & 0 \\ 2 & 1 \end{vmatrix} + \begin{vmatrix} 2 & 1 \\ 0 & 1 \end{vmatrix} + \begin{vmatrix} 0 & 1 \\ 1 & \frac{2}{3} \end{vmatrix} + \begin{vmatrix} 1 & \frac{2}{3} \\ 0 & 0 \end{vmatrix} \right)$$

$$= \frac{1}{2}(0 + 2 + 2 - 1 + 0) = \frac{3}{2} = 1 + \frac{1}{2}\left(\frac{1}{3} \times 1 + \frac{2}{3} \times 1\right)$$

Vertical Stripes.



$$A_{P_1P_2P_3P_4P_5P_6} = \sum_{i=1}^{N} A_i = \frac{1}{2} \sum_{i=1}^{N} (y_i + y_{i+1})(x_i - x_{i+1})$$

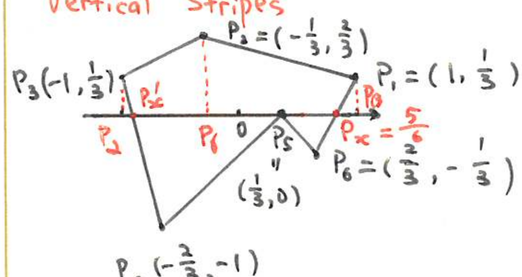
$$= \frac{1}{2}\Big(\cancel{x_1 y_1} + x_1 y_2 - x_2 y_1 - \cancel{x_2 y_2} + \cancel{x_2 y_2} + x_2 y_3 - x_3 y_2 - \cancel{x_3 y_3}$$

$$+ \cdots + \cancel{x_{n-1} y_{n-1}} + x_{n-1} y_n - x_n y_{n-1} - \cancel{x_n y_n} + \cancel{x_n y_n} + x_n y_1$$

$$- x_1 y_n - \cancel{x_1 y_1} \Big) = \frac{1}{2}\big(x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 + \cdots$$

$$+ x_{n-1} y_n - x_n y_{n-1} + x_n y_1 - x_1 y_n \big) = \frac{1}{2} \sum_{i=1}^{n} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}.$$

Vertical Stripes



$$\text{Signed Area} = \frac{1}{2}\left(\begin{vmatrix} 1 & \frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{vmatrix} + \begin{vmatrix} -\frac{1}{3} & \frac{2}{3} \\ -1 & \frac{1}{3} \end{vmatrix} + \begin{vmatrix} -1 & \frac{1}{3} \\ -\frac{2}{3} & -1 \end{vmatrix} \right.$$

$$\left. + \begin{vmatrix} -\frac{2}{3} & -1 \\ \frac{1}{3} & 0 \end{vmatrix} + \begin{vmatrix} \frac{1}{3} & 0 \\ \frac{2}{3} & -\frac{1}{3} \end{vmatrix} + \begin{vmatrix} \frac{2}{3} & -\frac{1}{3} \\ 1 & \frac{1}{3} \end{vmatrix} \right)$$

$$= \frac{1}{2}\left(\frac{2}{3} + \cancel{\frac{1}{9}} - \cancel{\frac{1}{9}} + \frac{2}{3} + 1 + \frac{2}{9} + \frac{1}{3} - \frac{1}{9} + \frac{2}{9} + \frac{1}{3} \right) = \frac{5}{3}$$

- Check with elementary geometry:

The line P_1P_6 : $\begin{cases} \frac{1}{3} = k \cdot 1 + b \\ -\frac{1}{3} = k \cdot \frac{2}{3} + b \end{cases} \Rightarrow \begin{cases} k = 2 \\ b = -\frac{5}{3} \end{cases} \Rightarrow 0 = 2 \cdot P_x - \frac{5}{3} \Rightarrow P_x = \frac{5}{6}$

The line P_3P_4 : $\begin{cases} \frac{1}{3} = -k + b \\ -1 = -\frac{2}{3}k + b \end{cases} \Rightarrow \begin{cases} \frac{1}{3}k = -\frac{4}{3}, \quad k = -4 \\ b = -\frac{11}{3} \end{cases} \Rightarrow 0 = -4P'_x - \frac{11}{3} \Rightarrow P'_x = -\frac{11}{12}$

$$\text{Positive area} = \underbrace{\frac{1}{2}\left(\frac{1}{3} + \frac{2}{3}\right) \cdot \frac{2}{3}}_{\square P_\gamma P_2 P_3 P_\alpha = \frac{1}{3}} - \underbrace{\frac{1}{2}\left(-\frac{11}{12} + 1\right) \cdot \frac{1}{3}}_{\triangle P_\alpha P'_x P_3 = \frac{1}{72}} + \underbrace{\frac{1}{2}\left(\frac{1}{3} + \frac{2}{3}\right)\left(1 + \frac{1}{3}\right)}_{\square P_\gamma P_\beta P_1 P_2 = \frac{2}{3}} - \underbrace{\frac{1}{2}\left(1 - \frac{5}{6}\right) \cdot \frac{1}{3}}_{\triangle P_x P_\beta P_1 = \frac{1}{36}} = \frac{23}{24}$$

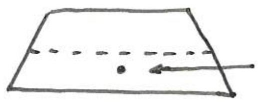
$$\text{Negative area} = \frac{1}{2}\left(\frac{1}{3} + \frac{11}{12}\right)(1) + \frac{1}{2}\left(\frac{5}{6} - \frac{1}{3}\right) \cdot \frac{1}{3} = \frac{15}{24} + \frac{1}{12} = \frac{17}{24}$$

Conclusion: Since the area of the polygon $\frac{5}{3}$ is the sum of the positive area $\frac{23}{24}$ and negative area $\frac{17}{24}$, it is independent of the choice of x-axis.

The formula $A = \frac{1}{2}\sum_{i=1}^{n} \begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}$ gives away another way to calculate the area of polygon by looking as triangulation from the origin.

Center of mass for polygons.

- Center of mass for triangles: $\left(\frac{1}{3}\sum_{i=1}^{3} x_i,\ \frac{1}{3}\sum_{i=1}^{3} y_i\right)$
- This does not generalize to n-gon for $n > 3$ unless weights are distributed on the vertices. Two counter examples:

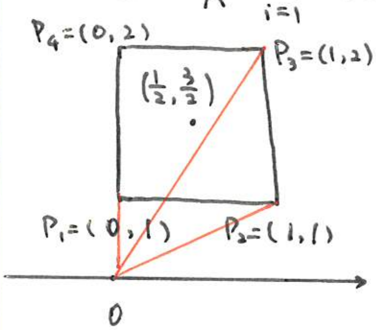


center of mass should be below the median but $\frac{1}{4}\sum_{i=1}^{4} y_i$ is on the median.

center of mass will converge to the arch if #points of the arch $\to \infty$

Weighted sum of triangulations.

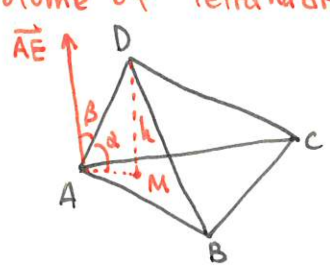
$$C = \frac{1}{A}\sum_{i=1}^{N} A_i \vec{C_i}, \quad \vec{C_i} = \text{Centroid}(\Delta ABC) = \frac{\vec{0} + \vec{OP_i} + \vec{OP_{i+1}}}{3}, \quad A_i = \frac{1}{2}\vec{OP_i} \times \vec{OP_{i+1}}$$



$$\vec{C_1} = \frac{1}{3}(1,2),\ 2A_1 = \vec{OP_1} \times \vec{OP_2} = \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} = -1$$
$$\vec{C_2} = \frac{1}{3}(2,3),\ 2A_2 = \vec{OP_2} \times \vec{OP_3} = \begin{vmatrix} 1 & 1 \\ 1 & 2 \end{vmatrix} = 2-1 = 1$$
$$\vec{C_3} = \frac{1}{3}(1,4),\ 2A_3 = \vec{OP_3} \times \vec{OP_4} = \begin{vmatrix} 1 & 2 \\ 0 & 2 \end{vmatrix} = 2$$
$$\vec{C_4} = \frac{1}{3}(0,3),\ 2A_4 = \vec{OP_4} \times \vec{OP_1} = \begin{vmatrix} 0 & 2 \\ 0 & 1 \end{vmatrix} = 0$$
$$C = \frac{1}{A_1+A_2+A_3+A_4}\left(-\frac{1}{6}(1,2) + \frac{1}{6}(2,3) + \frac{2}{6}(1,4)\right) = \left(\frac{1}{2}, \frac{3}{2}\right)$$

$$\Rightarrow C = \frac{1}{A}\sum_{i=1}^{N} \frac{(\vec{OP_i} + \vec{OP_{i+1}})(\vec{OP_i} \times \vec{OP_{i+1}})}{3} = \left(\frac{1}{6A}\sum_{i=1}^{N}(x_i + x_{i+1})\begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix},\ \frac{1}{6A}\sum_{i=1}^{N}(y_i + y_{i+1})\begin{vmatrix} x_i & y_i \\ x_{i+1} & y_{i+1} \end{vmatrix}\right)$$

Volume of Tetrahadron



$V_{ABCD} = \frac{1}{3} S_{\Delta ABC} h$

$S_{\Delta ABC} = \frac{1}{2}|\vec{AB} \times \vec{AC}|$, $h = |\vec{AD}|\sin\alpha$

α is the angle between $\vec{AM}$ and $\vec{AD}$

Set $\vec{AE} = \vec{AB} \times \vec{AC}$, β is the angle between $\vec{AD}$ and $\vec{AE}$

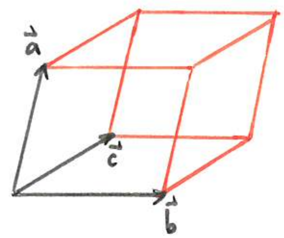
So $\cos\beta = \frac{\vec{AD} \cdot \vec{AE}}{\cancel{|\vec{AD}|}\,\cancel{|\vec{AE}|}} \Rightarrow V_{ABCD} = \frac{1}{6}\cancel{|\vec{AB} \times \vec{AC}|}\,\cancel{|\vec{AD}|}\,|\cos\beta|$

$\Rightarrow V_{ABCD} = \frac{1}{6}|\vec{AD} \cdot \vec{AE}| = \frac{1}{6}(\vec{AD} \cdot (\vec{AB} \times \vec{AC}))$.

Triple product. Geometrically, the triple product

$$\vec{a} \cdot (\vec{b} \times \vec{c})$$

is the signed volumn of the parallelepiped defined by the three vectors.



2D: $$\vec{AB} \times \vec{AC} = \begin{vmatrix} B_x - A_x & B_y - A_y \\ C_x - A_x & C_y - A_y \end{vmatrix} = B_x C_y - B_x A_y - A_x C_y + A_x A_y - B_y C_x + A_y C_x + B_y A_x - A_x A_y$$

$$= \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix} = \begin{vmatrix} B_x & B_y \\ C_x & C_y \end{vmatrix} - \begin{vmatrix} A_x & A_y \\ C_x & C_y \end{vmatrix} + \begin{vmatrix} A_x & A_y \\ B_x & B_y \end{vmatrix}$$

3D: $$\vec{AB} \times \vec{AC} = \begin{vmatrix} i & j & k \\ B_x - A_x & B_y - A_y & B_z - A_z \\ C_x - A_x & C_y - A_y & C_z - A_z \end{vmatrix}$$

$$\vec{AD} \cdot (\vec{AB} \times \vec{AC}) = (D_x - A_x) \begin{vmatrix} B_y - A_y & B_z - A_z \\ C_y - A_y & C_z - A_z \end{vmatrix} - (D_y - A_y) \begin{vmatrix} B_x - A_x & B_z - A_z \\ C_x - A_x & C_z - A_z \end{vmatrix}$$

$$+ (D_z - A_z) \begin{vmatrix} B_x - A_x & B_y - A_y \\ C_x - A_x & C_y - A_y \end{vmatrix}$$

$$= \begin{vmatrix} D_x - A_x & D_y - A_y & D_z - A_z \\ B_x - A_x & B_y - A_y & B_z - A_z \\ C_x - A_x & C_y - A_y & C_z - A_z \end{vmatrix}.$$

Simplex. A simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions. It represents the simplest possible polytope in any give dimension. The **volume** of an n-simplex in n-dimensional space with vertices $(\vec{v}_0, \ldots, \vec{v}_n)$ is:

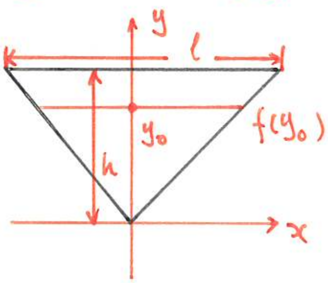
$$\text{Volume} = \frac{1}{n!} \left| \det(\vec{v}_1 - \vec{v}_0 \quad \vec{v}_2 - \vec{v}_0 \quad \cdots \quad \vec{v}_n - \vec{v}_0) \right| = \frac{1}{n!} \left| \det \begin{vmatrix} \vec{v}_0 & \cdots & \vec{v}_n \\ 1 & \cdots & 1 \end{vmatrix} \right|$$

Center of Mass for Simplices. In 1 dimension, the center of mass $\bar{x}$ satisfies: $\sum_{k=1}^{n} (x_k - \bar{x}) m_k = 0$, where m_k is the mass of the kth particle, x_k is the coordinates:

So the center of mass obeys

$$\sum_{k=1}^{n} x_k m_k = \bar{x} \sum_{k=1}^{n} m_k \Rightarrow \bar{x} = \frac{\sum_{k=1}^{n} x_k m_k}{\sum_{k=1}^{n} m_k}$$

In 2 dimension



The area of triangle is $S = \int_0^h f(y)\,dy$

So analogously, the center of mass is

$$C_y = \frac{\int_0^h y f(y)\,dy}{\int_0^h f(y)\,dy}$$

Because $\frac{l}{f(y)} = \frac{h}{y}$

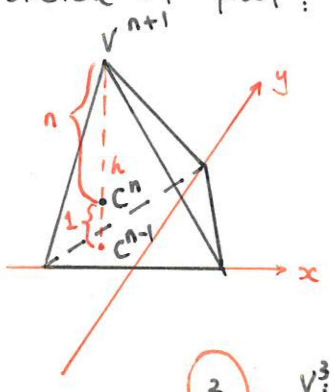
$$\Rightarrow C_y = \frac{\int_0^h \frac{y^2 \cancel{l}}{\cancel{k}}\,dy}{\int_0^h \frac{y\cancel{l}}{\cancel{k}}\,dy} = \frac{\frac{1}{3}y^3\Big|_0^h}{\frac{1}{2}y^2\Big|_0^h} = \frac{2h}{3}$$

In higher dimension

The center of mass is $\frac{\int y^{n+1}\,dy}{\int y^n\,dy} = \frac{n}{n+1} h$

i.e., center of mass locates at $\frac{n}{n+1}$ place of n-th coordinates.

Sketch of proof: $f_n(x_n) = c \cdot x_n^{n-1}$



Then in n dimension:

$$C_n = \frac{\int_0^h x_n f_n(x_n)\,dx_n}{\int_0^h f_n(x_n)\,dx_n} = \frac{\int_0^h x_n^n\,dx_n}{\int_0^h x_n^{n-1}\,dx_n} = \frac{\frac{1}{n+1}x_n^{n+1}}{\frac{1}{n}x_n^n}\Bigg|_0^h = \frac{n}{n+1}h$$

This shows $\frac{n}{n+1} = \frac{V_i^{n+1} - C_i^{n+1}}{V_i^{n+1} - C_i^n}$

In 2D, $\frac{2}{3} = \frac{V_i^3 - C_i^3}{V_i^3 - C_i^2} = \frac{V_i^3 - C_i^3}{V_i^3 - \frac{V_i^1 + V_i^2}{2}} \Rightarrow -V_i^1 - V_i^2 = V_i^3 - 3C_i^3 \Rightarrow C_i^3 = \frac{V_i^1 + V_i^2 + V_i^3}{3}$

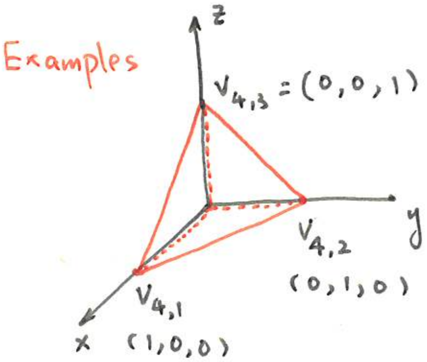
Assume true for k-dim: $C_i^k = \frac{1}{k}(V_i^1 + \cdots + V_i^k)$.

In $(k+1)$-dim, $\frac{k+1}{k+2} = \frac{V_i^{k+2} - C_i^{k+2}}{V_i^{k+2} - \frac{1}{k+1}(V_i^1 + \cdots + V_i^{k+1})} \Rightarrow C_i^{k+2} = \frac{1}{k+2}(V_i^1 + \cdots + V_i^{k+2})$

Volume of Polyhedron. We triangulate the polyhedron into simplicies. Set the number of faces of the polyhedron being m, and each face has N_i vertices, the volume of the polyhedron is

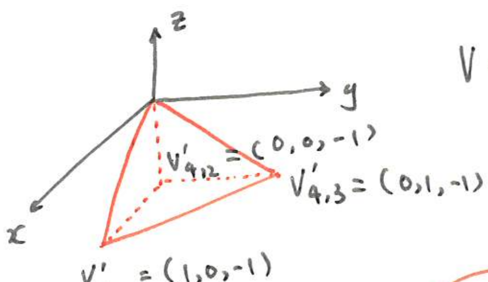
$$V = \frac{1}{6}\sum_{i=1}^{m}\sum_{j=2}^{N_i-1}\begin{vmatrix} x_{i,1} & y_{i,1} & z_{i,1} \\ x_{i,j} & y_{i,j} & z_{i,j} \\ x_{i,j+1} & y_{i,j+1} & z_{i,j+1}\end{vmatrix}, \quad \text{positive orientation}$$

Examples

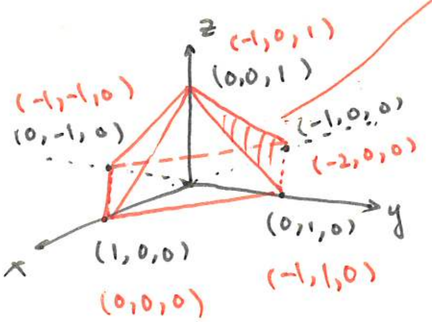


$$V = \frac{1}{6}\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1\end{vmatrix} = \frac{1}{6}$$

if we translate the z-coordinate by -1:



$$V = \frac{1}{6}\begin{vmatrix} 1 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 1 & -1\end{vmatrix} = \frac{1}{6}(+1)$$



$$V = \frac{1}{6}\left(\begin{vmatrix} -1 & 0 & 1 \\ -1 & 1 & 0 \\ -2 & 0 & 0\end{vmatrix} + \begin{vmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ -2 & 0 & 0\end{vmatrix}\right) = \frac{2}{3} = \frac{1}{3}\cdot h\cdot(\sqrt{2})^2$$

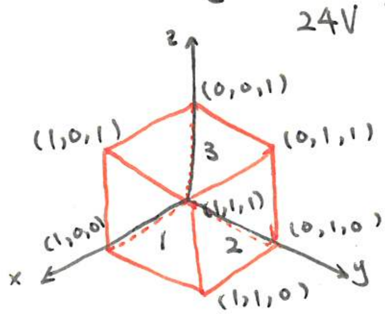
If we set the apex at the origin:

$$V = \frac{1}{6}\left(\begin{vmatrix} 1 & 0 & -1 \\ -1 & 0 & -1 \\ 0 & 1 & -1\end{vmatrix} + \begin{vmatrix} 1 & 0 & -1 \\ 0 & -1 & -1 \\ -1 & 0 & -1\end{vmatrix}\right) = \frac{1}{6}(1+1+1+1) = \frac{2}{3}$$

Center of Mass for Polyhedron. Similarly, the center of mass for polyhedron is

$$\vec{C} = \frac{1}{V}\sum_{i=1}^{m} V_i C_i, \quad \vec{C}_i = \frac{1}{V_i}\sum_{j=1}^{N_i} V_{ij}\vec{C}_{ij}, \quad \vec{C}_{ij} = \frac{\vec{O} + \vec{P}_{i,1} + \vec{P}_{i,j} + \vec{P}_{i,j+1}}{4}.$$

$$\vec{C} = \frac{1}{24V}\sum_{i=1}^{m}\sum_{j=1}^{N_i}(\vec{P}_{i,1} + \vec{P}_{i,j} + \vec{P}_{i,j+1})\begin{vmatrix} x_{i,1} & y_{i,1} & z_{i,1} \\ x_{i,j} & y_{i,j} & z_{i,j} \\ x_{i,j+1} & y_{i,j+1} & z_{i,j+1}\end{vmatrix}$$



$$C_{11}V_{11} = \frac{1}{6}\begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1\end{vmatrix}\frac{1}{4}(3,1,2) = \frac{1}{24}(3,1,2)$$

$\vec{C}_i$ center of the mass for the pyramid between the face and the apex. $\vec{C}_{ij}$ the center of mass for simplex

$$C_{12} V_{12} = \frac{1}{24} \begin{vmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix} (3,2,1) = \frac{1}{24}(3,2,1)$$

$$C_{21} V_{12} = \frac{1}{24} \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} (2,3,2) = \frac{1+1-1}{24}(2,3,2)$$

$$C_{22} V_{22} = \frac{1}{24} \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix} (1,3,1) = \frac{1}{24}(1,3,1)$$

$$C_{31} V_{31} = \frac{1}{24} \begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{vmatrix} (1,1,3) = \frac{1}{24}(1,1,3)$$

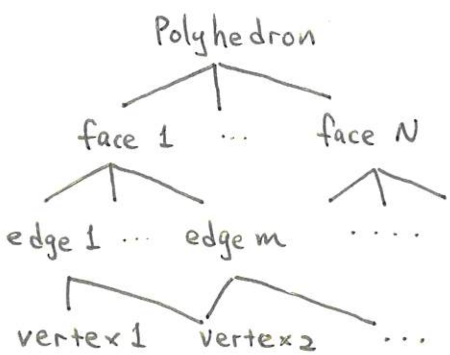
$$C_{32} V_{32} = \frac{1}{24} \begin{vmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix} (2,2,3) = \frac{1+1-1}{24}(2,2,3)$$

$$V = 1, \quad \vec{C} = \frac{1}{24}(12, 12, 12) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right).$$

Computational Complexity.

- Computing determinant $O(n!)$ directly
- Gauss elimination $\rightarrow$ $O(n^3)$ the compute determinant $O(n)$.
- Total complexity $O(vn^3)$, v the number of vertices.

Data Structure : tree



(dim n)

(dim $n-1$)

(dim $n-2$)

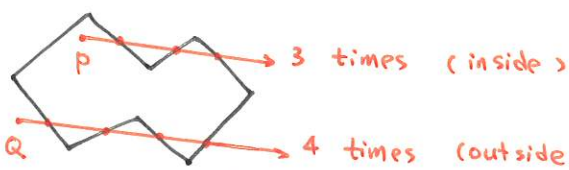
$\vdots$

(dim 0)

Point and Polygon & Polyhedron

- Point in the interior : $P \in D - \partial D$
- Point outside : $P \notin D$
- Point on the boundary : $P \in \partial D$

Rays. To determine if a point is in the interior, we draw a ray starting from this point (usually take the positive direction of the x-axis). If it intersects even number of times with D, it is outside. Otherwise inside:

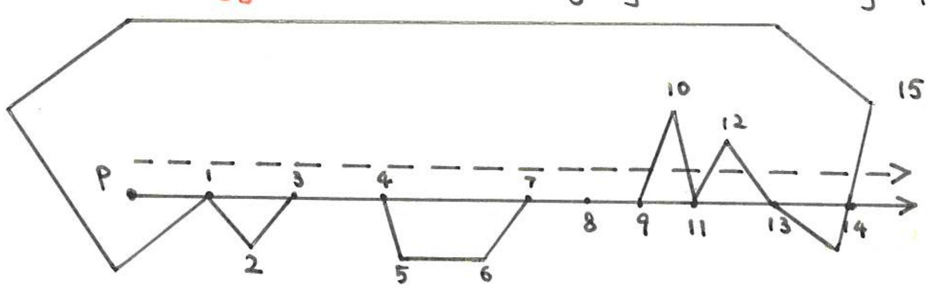


Special cases.

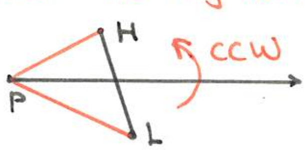
- **Improper intersection** : neighboring points not lay on different sides of the ray
- **Overlap**: a consecutive segment of the edge overlap with the ray.

For these two cases: omit points or segments that do not intersect properly or overlap.

- **Alternative strategy: translation.** Slightly move the ray upward or downward.



- **Alternative strategy: cross product.** For each line segment, we assign the higher end to be H and the lower end to be L. We require the segment intersect the ray to the right, i.e., $\vec{PL} \times \vec{PH} > 0$ or ≥ 0.



※ Okay to have L on the ray

Corner Case: What if P lies on the boundary or vertices?

- This depends on if we require $\vec{PL} \times \vec{PH} > 0$ or ≥ 0.

· If P on 14 : 0 intersection (↙ CCW < 0), outside if > 0, otherwise inside

· If P on 13 : 1 intersection (↙ @ 14), inside if > 0, otherwise outside (↙ @ 12-13)

· If P on 11 : 2 intersection (↙ @ 12-13, 14), outside. If ≥ 0, also intersect at 10-11, 11-12

· If P on 9 : 4 intersection (↙ @ 12-13, 14, 11-12, 10-11), outside if > 0. otherwise inside (↙ @ 9-10)

· If P on 8 : 4 intersection (↙ same as above), outside

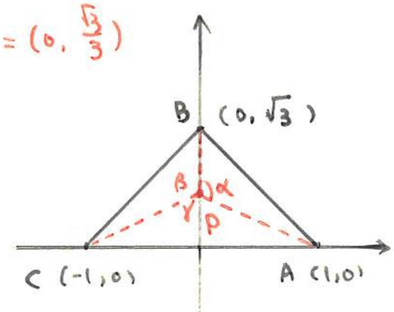
· If P on 1, 3, 4, 7 : same as 8.

Alternative Strategy: Angle Method

Consider a line segment with one end fixed at P, and another end trace around the polygon. If the angle tracing through the polygon changed by 2π, $P \in D$. If the signed angle changed by 0, the point is outside.

Implementation of Angle Method.

$P = (0, \frac{\sqrt{3}}{3})$



$$PA \times PB = \begin{vmatrix} 1 & -\frac{\sqrt{3}}{3} \\ 0 & \frac{2\sqrt{3}}{3} \end{vmatrix} = \frac{2\sqrt{3}}{3} > 0$$

$$\cos\alpha = \frac{PA \cdot PB}{\|PA\|\,\|PB\|} = \frac{-\frac{2}{9}\cdot 3}{\sqrt{1+\frac{3}{9}}\ \sqrt{\frac{4}{9}\cdot 3}} = -\frac{1}{2}, \quad \alpha = \frac{2}{3}\pi$$

$$PB \times PC = \begin{vmatrix} 0 & \frac{2}{3}\sqrt{3} \\ -1 & -\frac{\sqrt{3}}{3} \end{vmatrix} = \frac{2}{3}\sqrt{3} > 0$$

First calculate cross product to determine orientation.

The calculate dot product to determine angle

$$\cos\beta = \frac{PB \cdot PC}{\|PB\|\,\|PC\|} = \frac{-\frac{2}{9}\cdot 3}{\sqrt{\frac{4}{9}\cdot 3}\ \sqrt{1+\frac{3}{9}}} = -\frac{1}{2}, \quad \beta = \frac{2}{3}\pi$$

$$PC \times PA = \begin{vmatrix} -1 & -\frac{\sqrt{3}}{3} \\ 1 & -\frac{\sqrt{3}}{3} \end{vmatrix} = \frac{2}{3}\sqrt{3} > 0$$

$$\cos\gamma = \frac{PC \cdot PA}{\|PC\|\,\|PA\|} = \frac{-\frac{1}{3}}{\sqrt{\frac{4}{3}}\ \sqrt{\frac{4}{3}}} = -\frac{1}{2}. \quad \gamma = \frac{2}{3}\pi$$

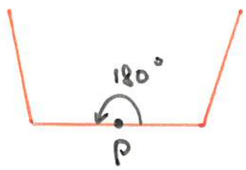
Now consider $P' = (0, -1)$. $PA = (1, 1)$, $PB = (0, \sqrt{3}+1)$, $PC = (-1, 1)$

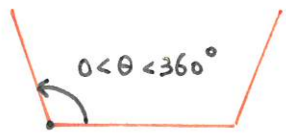
$PA \times PB = \sqrt{3}+1 > 0$ $\quad PB \times PC = \sqrt{3}+1$ $\quad PC \times PA = \begin{vmatrix} -1 & 1 \\ 1 & 1 \end{vmatrix} = -2 < 0$

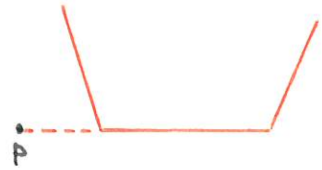
$\frac{PA \cdot PB}{\|PA\|\,\|PB\|} = \frac{\sqrt{3}+1}{\sqrt{2}(\sqrt{3}+1)} \nearrow \alpha' = \frac{\pi}{4}$ $\quad \frac{PB \cdot PC}{\|PB\|\,\|PC\|} = \frac{\sqrt{3}+1}{(\sqrt{3}+1)\sqrt{2}} \nearrow \beta' = \frac{\pi}{4}$ $\quad \frac{PC \cdot PA}{\|PC\|\,\|PA\|} = 0 \nearrow \gamma' = -\frac{\pi}{2}$

$\alpha' + \beta' + \gamma' = 0.$

Boundary Conditions: What if the cross product is zero ?







P locates on the boundary (not on the vertex), $P \subseteq \partial D - \partial^2 D$ dot product < 0

⇓

finished !

P locates on the vertex $P \subseteq \partial^2 D$ dot product $= 0$

⇓

finished !

P locates on the extension of ∂D (P might inside / outside of D) dot product > 0.

⇓

ignore this case

Emperical Remark. Due to squart root operation, inverse trigonometry, floating point numbers, the Angle Method can be 20 times slower than the method using rays.

Inside, Outside, and Orientation. To determine orientation (which direction is considered positive), we first need to determine how to decide inside and outside.

- For the method using rays, inside and outside can be determined by number of intersections with the boundary
- For the method using angles, inside is where points rotating for $\pm 2\pi$ when walking along the polygon.

Then we can define "positive direction" using the inside locates on the left-hand side always.

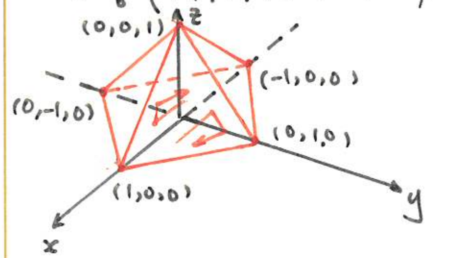
Volume of Polyhedron without Translation. Recall the volume of polyhedron can be computed
$$V = \frac{1}{6}\sum_{i=1}^{m}\sum_{j=2}^{N_i-1}\begin{vmatrix} x_{i,1} & y_{i,1} & z_{i,1} \\ x_{i,j} & y_{i,j} & z_{i,j} \\ x_{i,j+1} & y_{i,j+1} & z_{i,j+1}\end{vmatrix},$$
where $(x_{i,k}, y_{i,k}, z_{i,k})$ are positively oriented.

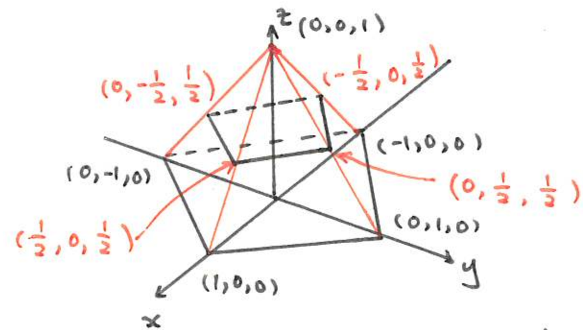
m: number of faces, N_i: number of vertices

Then for a pyramid,

$$V = \frac{1}{6}\left(\begin{vmatrix}1&0&0\\0&1&0\\0&0&1\end{vmatrix} + \begin{vmatrix}-1&0&0\\0&0&1\\0&1&0\end{vmatrix} + \begin{vmatrix}0&0&1\\-1&0&0\\0&-1&0\end{vmatrix} + \begin{vmatrix}1&0&0\\0&0&1\\0&-1&0\end{vmatrix} + \begin{vmatrix}1&0&0\\-1&0&0\\0&1&0\end{vmatrix} + \begin{vmatrix}1&0&0\\0&-1&0\\-1&0&0\end{vmatrix}\right)$$

$$= \frac{1}{6}(1+1+1+1+0+0) = \frac{2}{3}$$





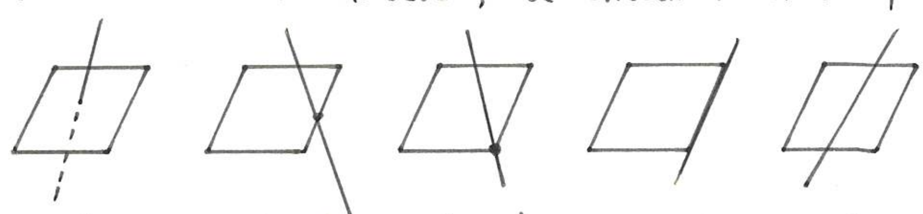
$$V = \frac{1}{6}\left(\begin{vmatrix}1&0&0\\0&1&0\\0&.5&.5\end{vmatrix} + \begin{vmatrix}1&0&0\\0&.5&.5\\.5&0&.5\end{vmatrix} + \begin{vmatrix}0&1&0\\-1&0&0\\-.5&0&.5\end{vmatrix} + \begin{vmatrix}0&1&0\\-.5&0&.5\\0&.5&.5\end{vmatrix} + \begin{vmatrix}0&-.5&.5\\-.5&0&.5\\-1&0&0\end{vmatrix} + \begin{vmatrix}0&-.5&.5\\-1&0&0\\0&-1&0\end{vmatrix}\right.$$
$$\left. + \begin{vmatrix}1&0&0\\.5&0&.5\\0&-.5&.5\end{vmatrix} + \begin{vmatrix}1&0&0\\0&-.5&.5\\0&-1&0\end{vmatrix} + \begin{vmatrix}.5&0&.5\\0&.5&.5\\-.5&0&.5\end{vmatrix} + \begin{vmatrix}.5&0&.5\\-.5&0&.5\\0&-.5&.5\end{vmatrix} + \begin{vmatrix}1&0&0\\-1&0&0\\0&1&0\end{vmatrix} + \begin{vmatrix}1&0&0\\0&-1&0\\-1&0&0\end{vmatrix}\right)$$

$$= \frac{1}{6}\left(\frac{1}{2}+\frac{1}{4}+\frac{1}{2}+\frac{1}{4}+\frac{1}{4}+\frac{1}{2}+\frac{1}{4}+\frac{1}{2}+\left(\frac{1}{8}+\frac{1}{8}\right)+\left(\frac{1}{8}+\frac{1}{8}\right)+0+0\right) = \frac{1}{6}\cdot\frac{7}{2}$$

Alternatively, $V = V_{pyramid} - V_{small} = \frac{1}{3}\cdot 2\cdot 1 - \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2} = \frac{7}{12}$, as above.

Ray Method for Polyhedron

For polyhedron, we generalize the ray method into 3D. We count how many faces a ray in 3D passes through, and use the parity to determine inside and outside. However, we encounter more special cases:

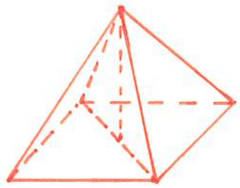


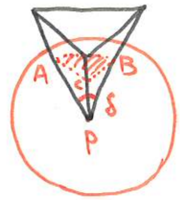
(a) normal case passing through (b) through edge (c) through vertex (d) overlap edge (e) through two edges

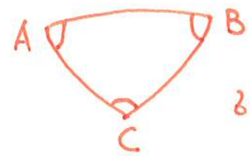
Solution: generate different rays until no special cases (b – e) appear.

Solid Angel and Angel Method for Polyhedron

A pyramid can be divided into tetrahedrons by triangulating the base. The solid angle of the pyramid is the signed sum of the solid angle of the apex of the tetrahedrons.

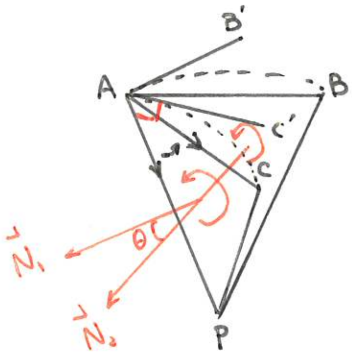






The solid angle of a tetrahedron can be defined as the following: take the apex as the center of a unit ball. The area of its segment on the unit ball is defined as the solid angle.

$$S_{ABC} = R^2 \delta = \delta$$



Let AC' be the tangent line of the curve AC, and AB' be the tangent line of the curve AB. Then $AC' \perp AP$, $AB' \perp AP$, thus $\angle B'AC'$ is the dihedral angle of PAB and PAC.

How do we compute dihedral angle?

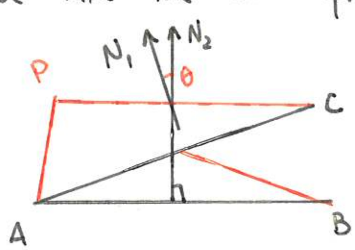
We take the cross product of $\vec{AP}$ and $\vec{AC}$ to get the normal of PAC:

$$\vec{N}_1 = \vec{AP} \times \vec{AC}$$

Then the normal of PAB:

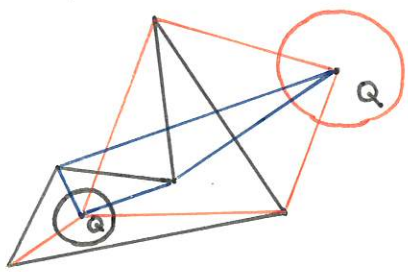
$$\vec{N}_2 = \vec{AP} \times \vec{AB}$$

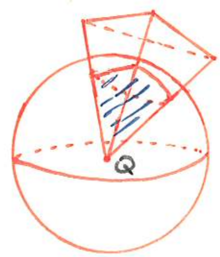
Then we take the dot product $\vec{N}_1 \cdot \vec{N}_2$ to get the dihedral angle.



Angle Method Generalization to High Dimension

- Angle Method for 2D II: set Q as the center of a small circle, add the signed area of the intersection of the small circle and the shape between Q and each boundary. Inside if the sum is πr^2, outside if the sum is 0.

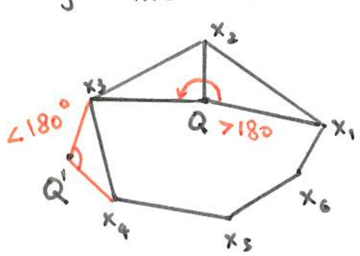




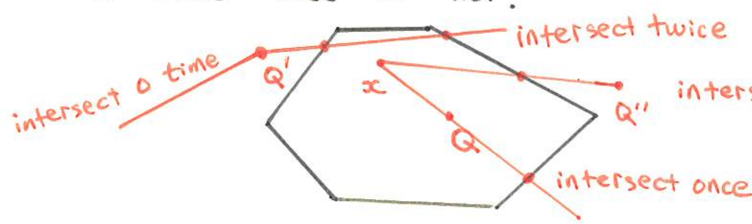
- Angle Method for 3D: set Q as the center of a small ball, add the signed volumn of the intersection of the ball and the solid between Q and the base of the tetrahedron. If the sum is $\frac{3}{4}\pi r^3$, Q is inside. If the sum is 0, Q is outside.
- Generalized Angle Method for high dimension: draw a small ball centered at Q. For each n-1 dimension face, calculate the volume of the intersection between this ball and the pyramid between Q and this face. If the total volume is that of the n-dim ball, then Q is inside the polytope. If the total volume is 0, then Q is outside of the polytope.

Convex Polygon

- For convex polygon, no need to add up the angles. As soon as Qx_i surpass the opposite direction, the iteration is over. However, this does not change the worst case complexity.



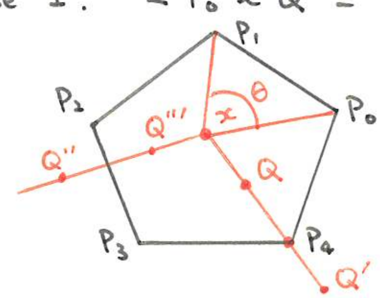
Special Property. Using the ray method, if a point is outside of a convex polygon, then the ray intersects the polygon for 0 time or 2 times; otherwise 1 time. However, if we choose a point x inside the polygon, and draw a ray passing through Q, then this ray intersects the polygon precisely once. Then we only need to determine if x and Q lie on the same side or not.



Binary Search Algorithm. So now the question is how to find the edge that intersects xQ?

- For convex polygon, picking a starting point, then the angle is strictly increasing. So we choose binary search, which is $O(\log n)$.
- Step 1. Set xP_0 as the starting point, i.e., $\theta = 0$, and let $xP_n = xP_0$ as the ending point after iterate through $P_1, \dots, P_{n-1}$.
- Step 2. Use binary search to find the edge P_iP_{i+1} that intersects xQ.

Case 1. $\angle P_0 x Q = \angle P_0 x P_i$, use dot product



$\overrightarrow{xP_i} \cdot \overrightarrow{QP_i} > 0$ inside

$\overrightarrow{xP_i} \cdot \overrightarrow{QP_i} = 0$ i.e., $Q = P_i$

$\overrightarrow{xP_i} \cdot \overrightarrow{QP_i} < 0$ outside

Case 2. xQ falls between xP_i and xP_{i+1}

$\overrightarrow{QP_i} \times \overrightarrow{QP_{i+1}} > 0$ inside (Q''' case)

$\overrightarrow{QP_i} \times \overrightarrow{QP_{i+1}} < 0$ outside (Q'' case)

$\overrightarrow{QP_i} \times \overrightarrow{QP_{i+1}} = 0$ on the edge

Discussions

- Reading the n vertices is already linear.
  - The algorithm complexity analysis excluded I/O time.
  - Realistically, achivable through "Online Inquiry" mechanism, i.e., only read needed (called) vertices. Thus, the complexity is strictly $O(\log n)$.
- How to find x (an arbitrary interior point)

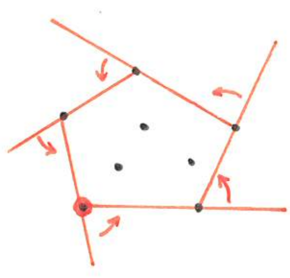
  $\vec{x} = \frac{1}{3}(P_i + P_j + P_k)$ for arbitrary 3 distinct vertices
- How about star polygon

  The complexity is still $O(\log n)$ if the point is inside the kernel of the star polygon. (More will be discussed.)

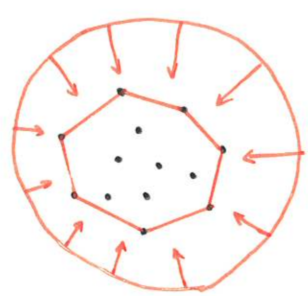
Convex Hull and Applications

Agenda
- Review Convexity
- Find convex hull and corner cases
- Correctness and complexity, connecting with sorting
- Properties and applications

Problem 1. Building a fence to surround woods
- The smaller the better (to reduce cost) → convex hull
- Two ways to analyze it: the lasso and the vacuum



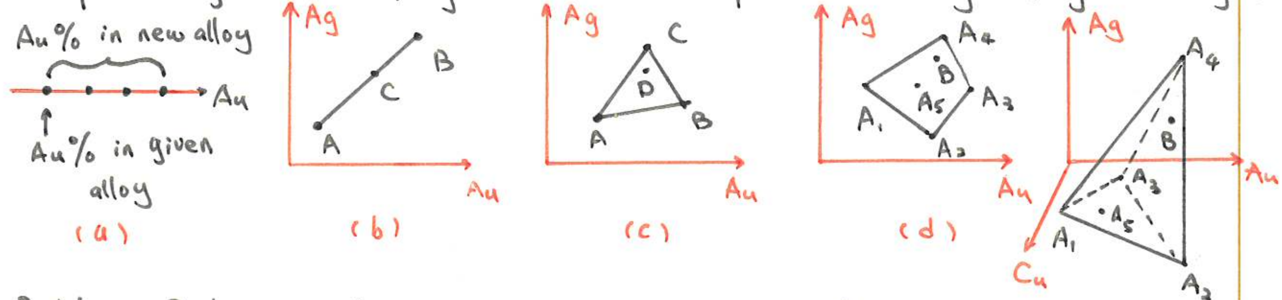
The Lasso



The Vacuum

Problem 2. Alloy Mixture Problem → convex hull

Given some Au - Ag alloys with different Au, Ag composite. What is the percentage of Au, Ag that can be produced using the given alloys?



Problem Reduction (convert to a geometry problem)

(a) Consider Au only, the perrentage of Au in the new alloy is between the min and max of the percentage of Au previously.

(b) Consider Au and Ag, starting with two Au - Ag alloys. The percentage of Au and Ag lies between those in the old alloys:

$$\begin{cases} x_C = \lambda x_A + (1-\lambda) x_B \\ y_C = \lambda y_A + (1-\lambda) y_B \end{cases}, \quad \lambda \in [0, 1]$$

(x_A, y_A) is the percentage of Au and Ag in alloy A
(x_B, y_B) is that in alloy B. Symbolically:

$$C = \lambda A + (1+\lambda) B$$

C denotes the mixture of Au and Ag in the new alloy, i.e., C is between the line segment AB.

(c) Consider 3 alloys A, B, C, then the mixture D lies inside the triangle $\triangle ABC$.

(d) Generalize to n alloys, $A_1, \dots, A_n$, the new alloy lies inside the convex hull of old alloys

(e) Generalize to m metals, each alloy i $(i = 1 \sim n)$ contains x_{ij}% percent of metal j $(j = 1 \sim m)$. Then the composite of the new alloy lies in the convex hull of the previous alloys.

Review: Convexity (generalize to high dimension)

Definition 1: The shape lies on one side of each edge. (Inspired the Gift-Wrapping Algorithm.)

Definition 2: D is convex $\Leftrightarrow \forall x_1, x_2 \in D$, $\frac{x_1+x_2}{2} \in D$.

(This generalize to any point in between.)

Definition 3: D is convex $\Leftrightarrow \forall x_1, x_2 \in D, \lambda \in [0,1], \lambda x_1 + (1-\lambda) x_2 \in D$

Definition 4: D is convex $\Leftrightarrow \forall x_i \in D \ (i=1,\dots,n)$, $\forall \lambda_i \geq 0$ and $\sum_i \lambda = 1$:

$$\sum_i \lambda_i x_i \in D.$$

Convex hull. For a set of points on a plane, or a polygon, its convex hull is the smallest convex region that contains it, i.e., the convex hull is the collection of all convex combination of those points. (convex combination refers to linear combination of points where all coefficients are nonnegative and som up to 1.)

Extremal points. The vertices on the convex hull (excluding middle points between two vertices) are called extremal points.

Theorem. The vertices of a convex hull for a set of points are contained in that set.

Application of Convex Hull

① Number of points reduced. For independently and identically distributed n points in dimension k, the number of vertices in its convex hull $m(k) = O(n^{\frac{k-1}{k+1}})$. → Greatly reduce average-case complexity (not worst-case complexity).

② The convex hull consists of a list of ordered points. The order can induce some algorithm to further decrease worst-case complexity.

Problem 3. The biggest distance between two points in a set of points.

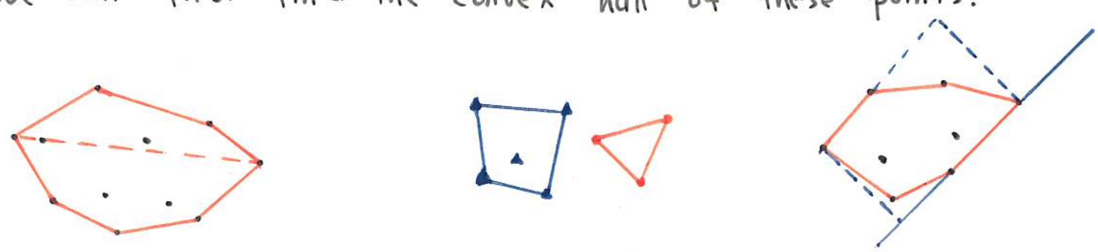
We can use convex hull to reduce the worst-case complexity.

Problem 4. Given a set of red points and a set of blue points, if they can be separated by a straight line.

- Naive approach: test if all pairs of red points intersect with any pair of blue points. $O(n^4)$.
- Convex hull: Determine if any edge from the convex hull of the red points intersects with that of the blue points. Reduced worst-case complexity.

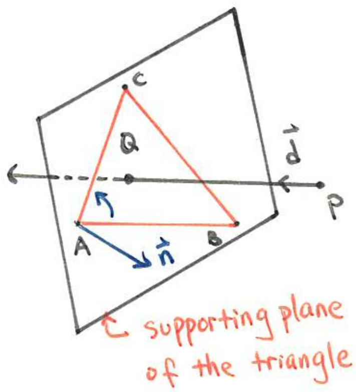
Problem 5. Find the smallest rectangle that contains all the points.

We can first find the convex hull of these points.



Problem 3 Problem 4 Problem 5

Ray - Triangle Intersection



Consider a ray start at P with direction $\vec{d}$, and intersects a triangle ABC at point Q.

Step 1. Determine the point of intersection, Q

↳ The equation of supporting plane

↳ Intersect the ray with the supporting plane

Step 2. Determine if Q is inside ABC.

Ray - plane intersection.

The equation of a plane can be written as:

$$ax + by + cz = d \iff \vec{n} \cdot \vec{x} = d, \quad \vec{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Now consider the ray determined by P and $\vec{d}$:

$$R(t) = P + t\vec{d}.$$

Now solve for the intersection:

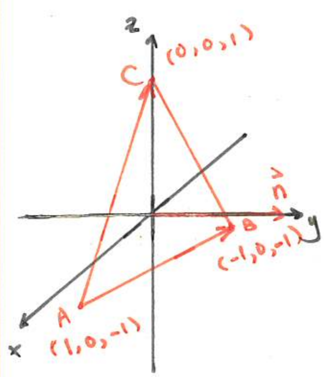
$$\vec{n} \cdot [P + t\vec{d}] = d \Rightarrow \vec{n} \cdot P + t\,\vec{n} \cdot \vec{d} = d$$

$$\Rightarrow t = \frac{d - \vec{n} \cdot P}{\vec{n} \cdot \vec{d}}$$

if $\vec{n} \cdot \vec{d} \neq 0$. Otherwise parallel or contained in the plane.

Then Q can be determined by plugging t.

Solving for Supporting Plane



$\vec{n} = \vec{AB} \times \vec{AC}$, then pick a point on the plane to solve for d, i.e., $d = \vec{n} \cdot \vec{A}$.

Eg. $\vec{AB} = (-2, 0, 0)$, $\vec{AC} = (-1, 0, 2)$.

$$\vec{AB} \times \vec{AC} = \begin{vmatrix} i & j & k \\ -2 & 0 & 0 \\ -1 & 0 & 2 \end{vmatrix} = 4j$$

$$d = \vec{n} \cdot \vec{B} = \begin{pmatrix} 0 \\ 4 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 0 \\ -1 \end{pmatrix} = 0$$

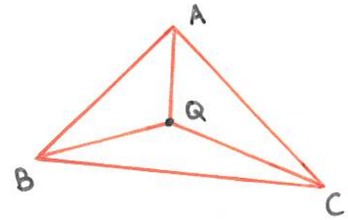
Triangle inside-outside testing

Q is inside of ABC if the following conditions are met simultaneously:

$$(\vec{AB} \times \vec{AQ}) \cdot \vec{n} \geq 0$$

$$(\vec{BC} \times \vec{BQ}) \cdot \vec{n} \geq 0$$

$$(\vec{CA} \times \vec{CQ}) \cdot \vec{n} \geq 0$$



(Equally deals with the boundary case: overlap with vertices or on the edge.)

Eg. Set $R(t) = (1,1,1) + t(1,1,1)$, $\vec{n} = (0,4,0)$, $d = 0$

$$t = \frac{d - \vec{n}\cdot p}{\vec{n}\cdot \vec{d}} = \frac{0 - (0,4,0)\cdot(1,1,1)}{(0,4,0)\cdot(1,1,1)} = \frac{-4}{4} = -1$$

So $Q = (0,0,0)$.

$$(\vec{AB} \times \vec{AQ}) \cdot \vec{n} = \begin{vmatrix} i & j & k \\ -2 & 0 & 0 \\ -1 & 0 & 1 \end{vmatrix} \cdot \vec{n} = 2j \cdot \vec{n} = 8$$

$$(\vec{BC} \times \vec{BQ}) \cdot \vec{n} = \begin{vmatrix} i & j & k \\ 1 & 0 & 2 \\ 1 & 0 & 1 \end{vmatrix} \cdot \vec{n} = j \cdot \vec{n} = 4$$

$$(\vec{CA} \times \vec{CQ}) \cdot \vec{n} = \begin{vmatrix} i & j & k \\ 1 & 0 & -2 \\ 0 & 0 & -1 \end{vmatrix} \cdot \vec{n} = j \cdot \vec{n} = 4$$

(Can project on 2D to accelerate.)

Computing Barycentric Coordinates

If Q is inside of ABC, we will compute the barycentric coordinates of Q with respect to ABC:

$$\alpha = \frac{\text{Area } QBC}{\text{Area } ABC}, \quad \beta = \frac{\text{Area } AQC}{\text{Area } ABC}, \quad \gamma = \frac{\text{Area } ABQ}{\text{Area } ABC}$$

Recall Area $QBC = \frac{1}{2}\|(C-B)\times(Q-B)\|$. since the cross product points in the $\vec{n}$ direction when Q is inside of ABC, then

$$\text{Area } QBC = \frac{1}{2}[(C-B)\times(Q-B)]\cdot \vec{N}, \quad \vec{N} = \frac{\vec{n}}{\|\vec{n}\|}.$$

So

$$\alpha = \frac{[(C-B)\times(Q-B)]\cdot\vec{n}}{[(B-A)\times(C-A)]\cdot\vec{n}}, \quad \beta = \frac{[(A-C)\times(Q-C)]\cdot\vec{n}}{[(B-A)\times(C-A)]\cdot\vec{n}}, \quad \gamma = \frac{[(B-A)\times(Q-A)]\cdot\vec{n}}{[(B-A)\times(C-A)]\cdot\vec{n}}$$

Note that $\vec{n}$ does not need to be normalized because they cancel out.

- Eg. $\vec{AB} \times \vec{AC} = 4j = \vec{n}$, $(\vec{AB} \times \vec{AC}) \cdot \vec{n} = 16$

$$\left.\begin{aligned} (\vec{BC} \times \vec{BQ}) \cdot \vec{n} &= 4 \\ (\vec{CA} \times \vec{CQ}) \cdot \vec{n} &= 4 \\ (\vec{AB} \times \vec{AQ}) \cdot \vec{n} &= 8 \end{aligned}\right\} \Rightarrow \begin{aligned} \alpha &= \tfrac{1}{4} \\ \beta &= \tfrac{1}{4} \\ \gamma &= \tfrac{1}{2} \end{aligned}$$

$$Q = \tfrac{1}{4}A + \tfrac{1}{4}B + \tfrac{1}{2}C = \tfrac{1}{4}(1,0,-1) + \tfrac{1}{4}(-1,0,-1) + \tfrac{1}{2}(0,0,1)$$

$$= (0, 0, 0)$$ as expected.

Convex Hull Algorithm

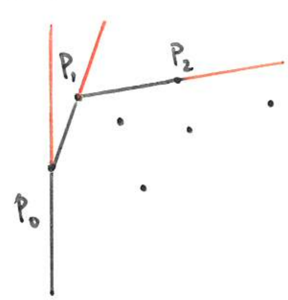
1. Gift-Wrapping

A set of points $\subset$ 2D $\rightarrow$ [convex hull algorithm] $\rightarrow$ a set of points (unordered)

a polygon $\subset$ 2D $\rightarrow$ [convex hull algorithm] $\rightarrow$ a polygon (ordered points)

※ Assume no three points are colinear.

- Starting with P_0 known to be on the convex hull - the leftmost point
- select P_{i+1} such that all the points are to the right of $P_i P_{i+1}$



- Selecting a point P_j, determining if all other points are to the right: $O(n^2)$.
- Simple improvement: compare polar angles between $P_i P_j$: $O(n)$

- Total time: $O(nh)$, h is the number of points on the convex hull.
  - polar angle is hard to compute
  - use cross product: $P_i P_j \times P_i P_{j+1} > 0$, P_{j+1} is to the left

※ Three points are colinear: including extremal points (vertices of convex hull) or not and output accordingly. cross product $\rightarrow 0$

Graham - Scan Algorithm

Step 1. Find the point with lowest y-coordinate. If not unique, pick the one with lowest x-coordinate. $O(n)$

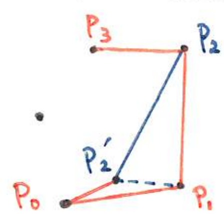
Step 2. Sort the set of points in increasing order of the angle they make with P and the x-axis. Heapsort : $O(n \log n)$.

- Comparison function used by the sorting algorithm can use cross product
- Ties can be broken by increasing distance or delete all but the furthest point from the current point.
- Distance : L^1 : $dist(a,b) := \sum_{i=1}^{n} |a_i - b_i|$

  L^∞ (Chebyshev) : $dist(a,b) := \max_i |a_i - b_i|$

Step 3. Consider each point in the sorted array.

- Determine traveling from the two points immediately preceding the current point makes a left or right turn.
- If a right turn, the second-to-last point is not part of the convex hull. Proceed with its preceding two points



$P_2' P_2$ is a right turn from $P_1 P_2'$ so P_2' is an inside point. Then procede with the next two points after P_2'

- If a left turn, move to next point that is not labeled as inside.

Colinear: report it or discard it, depending on the problem.

Pseudocode:
```
push(P1); push(P2);
i = 3
while i ≤ n
    if Pi on the left of P_next_to_top P_top
        push(Pi), i++
    else pop()
```

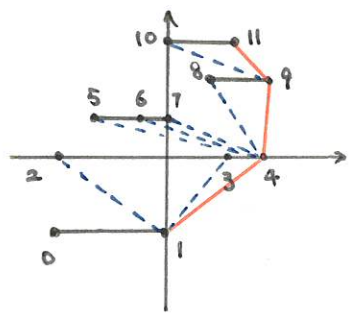
Amortized Analysis

- Sorting the points : $O(n \log n)$
- Finding the convex hull : $O(n)$ because each point can only appear once as a left point and once as a right point.

Horizontal Order for Graham - Scan

step 1. Rank by horizontal order, ties break by vertical order.

step 2. Right chain: from 0 to the last point.



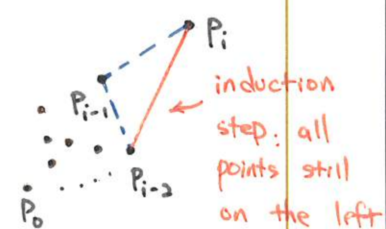
step 3. Left chain: from the last point to the first point.

Correctness of Graham - Scan

We use induction. Let CH(i) denote the local convex hull for the set of points $P = \{P_0, P_1, \dots, P_i\}$. Without loss of generality, we only consider the right chain. We need to prove:

(1) The chain is convex

(2) It is the smallest that containing all the points

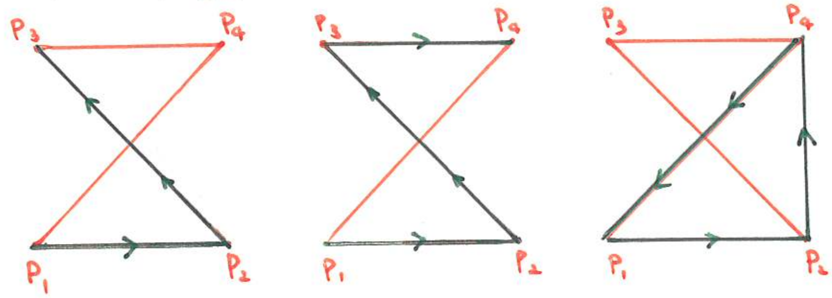


For (1) it is guarateed by the "left-turn" criteria. For (2), being the smallest is ensured by $CH(i) \subset P$. To show the convex hull contains all points, we show all points are to the left for the right chain.

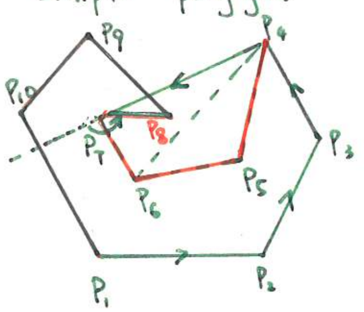
- Base case: P_0, P_1 are to the left (contained on the right chain)
- Hypothesis : $\{P_0, \dots, P_{i-1}\}$ are to the left of $CH(i-1)$.
- Induction Step: If $P_{i-1}P_i$ is a left turn, $CH(i) = CH(i-1) + P_{i-1}P_i$, valid. If it is a right turn, $\{P_0, \dots, P_{i-1}\}$ also on the left of $CH(i)$.

Convex Hull of Polygons

Complex polygon

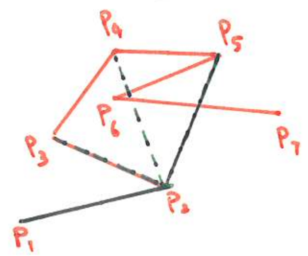


Simple polygon



So Graham-Scan does not work for simple polygon, and there is no easy fix. For example, if we exclude new points inside of the existing polygon, then by the time adding P_8, the convex hull is $P_1 P_2 P_3 P_4 P_7$, inside.

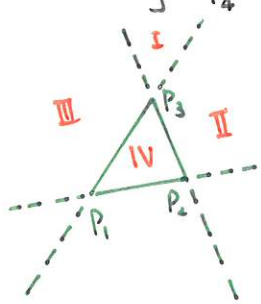
Counter example: Unfortunately, this is not a fix of Graham-Scan for polygons.



At P_5, the convex hull is $P_1 P_2 P_5$, then $P_5 P_6$ is outside of the convex hull (connecting P_1 and P_5), and turning left, so will be added. So we need a new algorithm.

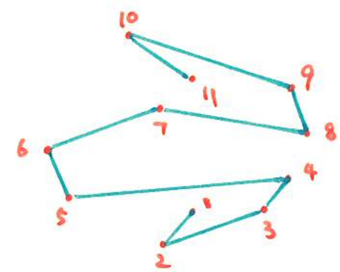
Melkman Algorithm

- deque. initialization: positively oriented three points P_1, P_2, P_3
- Adding P_4:
  - If $P_4 \in$ Region IV: inside the convex hull. skip.
  - If $P_4 \in$ Region II: add into both ends of deque pop top until all turn left. (i.e., "convex turns")
  - If $P_4 \in$ Region III: add into both ends of deque pop bottom until all turn right. ("convex turn")
  - If $P_4 \in$ Region I: repeat both two steps above.



Example. Start with 1, 2, 3

		4	1	2	3	4						III
	5	4	1	2	3	4	5					III
	6	5	2	3	4	5	6					II
	7	6	5	2	3	4	6	7				II
8	7	6	5	2	3	4	7	8				II
9	8	7	6	5	2	3	4	8	9			III
10	9	7	6	5	2	3	4	8	9	10		III
11	10	6	5	2	3	4	8	9	10	11		IV

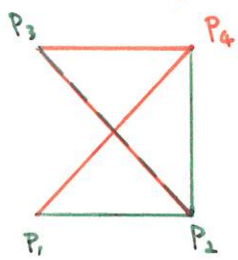


Two ways to view Melkman Algorithm

- top of the deque remain turning left, bottom turning right. i.e., convex
- the sequence in the deque remain positively oriented.

This is an online algorithm. Can start with any point and update interactively.

Does Melkman Algorithm work for complex polygon?



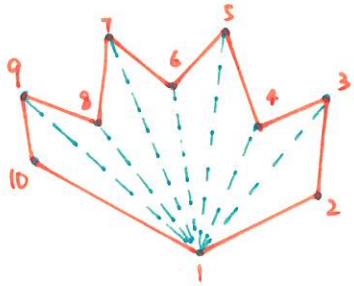
4 1 2 3 4 II

No.

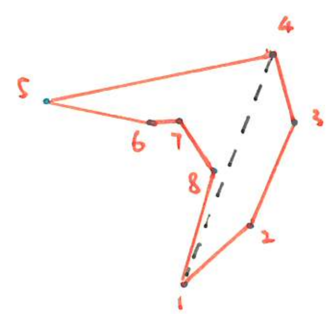
Melkman Algorithm Complexity: $O(n)$

- Can we use Melkman Algorithm for the convex hull of points? $O(n \log n)$

  No, because it does not apply to complex polygon, so we have to sort. $O(n \log n)$



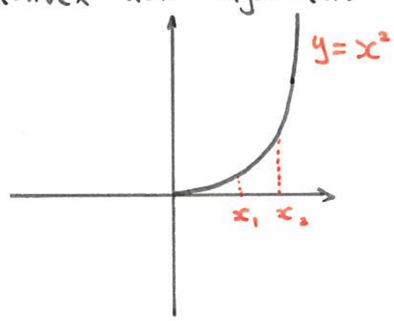
polar order



horizontal order

Find the min/max points, and then proceed from the right and left.

- $\Omega(n\log n)$ is a lower bound of convex hull algorithms for points
  - The lower bound of sorting is $\Omega(n\log n)$
  - If convex hull algorithms can go below $\Omega(n\log n)$, so does sorting



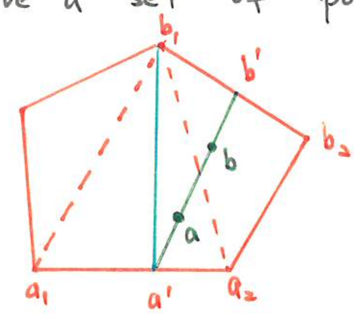
Finding the convex hull of $(x_1, x_1^2), \dots, (x_n, x_n^2)$ can convert to sorting.

  - This lower bound can be achieved (using Graham-Scan)
- Converting a set of points to a simple polygon has lower bound $\Omega(n\log n)$.
  - Melkman algorithm: $O(n)$ for simple polygon
  - So $\Omega(n\log n)$ is the lower bound for points to simple polygon
- Generalized Graham-scan: $O(n\log n)$
  - Convert the set of points to simple polygon. $O(n\log n)$
  - Melkan algorithm: $O(n)$

Applications: use complex hull to improve average-case complexity

Example 1. The diameter of a set of points

Give a set of points $S=\{\vec{p}_0, \dots, \vec{p}_{n-1}\}$, $D_s := \max\limits_{i,j} |p_i - p_j|$.



Brute Force: $O(n^2)$.

Can we reduce to points on its convex hull?

↳ Is p_i, p_j with max distance locate on the convex hull as vertices?

- It is clear that a, b must be on the boundary.
- If a', b' (intersection of the extension of ab with the boundary) is not on the vertex, then $a'b_1 > a'b'$ or $a'b_2 > a'b'$, and

$a_1b_1 > a'b_1$ or $a_2b_1 > a'b_1$. $\quad a_1b_2, a_1b_1 > a'b_1 > a'b' \Rightarrow$ must be the extreme points

$a_1b_2 > a'b_2$ or $a_2b_2 > a'b_2$ $\quad a_2b_2, a_2b_1 > a'b_2$

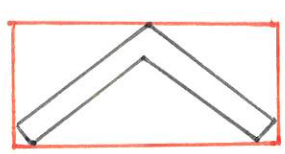
Applications: Use convexity to improve worst-case complexity

Example 2. Smallest Enclosing Box (UVA 10173)

- If the enclosing box is parallel to the coordinates, then only need to compute the horizontal and vertical span.
- We can enumerate $\theta \in [0, \frac{\pi}{2})$, recalculate the coordinates, then get the span. But there are precision issues (and complexity issues).

A smarter discretization: choose θ to be the angle between two points.

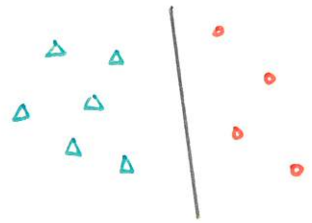
- There are C_n^2 choices
- sketch of the proof:
  1. each edge of the enclosing box has at least one point.
  2. at least one edge has two points. If not, rotate the box around each of the points in two direction. Then compute the area of the two new boxes. It can be proved that one of the new boxes will be smaller than the original one.
- Therefore, for each angle θ_i, compute the horizontal and vertical span $\Theta(C_n^2 \cdot n) = \Theta(n^3)$.
- Suffice to compute angle θ_i formed by edges on the convex hull: $\Theta(n \log n)$ for convex hull $+ \Theta(n \cdot n) = \Theta(n^2)$.
- If the input is already a polygon, we still treat as point set, because we can only prove the enclosing box contains two points, not an edge. A counter example:



The smallest enclosing box does not share an edge with the polygon.

Example 3. The Great Divide (UVA 10256)

Given a set of red points and blue points, if there exists a line to separate them:



Straightforward approach: compute the clique of red and blue respectively, and see if any pair of red points intersect blue. $O(n^4)$

Instead, we can investigate if the convex hull of each set intersects.

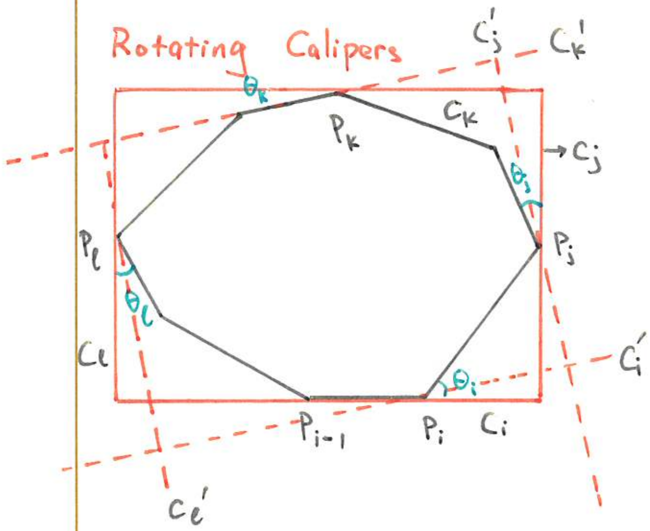
- If one blue edge intersects one red edge $\Rightarrow$ can not separate
- Otherwise $\Rightarrow$ $\begin{cases} \text{can separate} \\ \text{one is contained in another} \end{cases}$ $\xrightarrow{\text{test}}$ take a red point, is it inside of the blue hull? / take a blue point, is it inside of the red hull?

Complexity: $2O(n\log n) + O(n^2) = O(n^2)$

Here we assumed that convex hull can be separated

$\Leftrightarrow$ two sets of points can be separated

Rotating Calipers



- Start from the edge $P_{i-1}P_i$ on the box
- Select the smallest angel to rotate (θ_k)
- Because θ_k is the smallest, C_i', C_j', C_l' still contains P_i, P_j, P_l. This implies for each time, we only need to select the smallest θ and roll ahead.
- We accumulate the rotation until sum $> \frac{\pi}{2}$.

- Summary.

Approximating $\longrightarrow$ enumerate angle $O(n^3)$ $\longrightarrow$ enumerate convex hull edges $O(n^2)$ $\longrightarrow$ rotating caliper $O(n\log n)$ convex hull $+ O(n)$ rotation