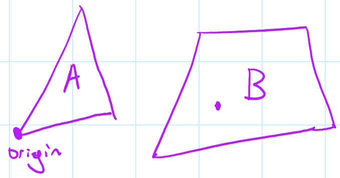
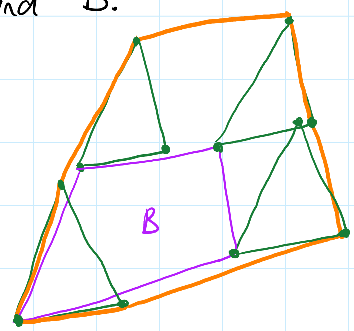
HOW CAN WE COMPUTE THE MINKOWSKI SUM?

Start with convex polygons A and B.

example:





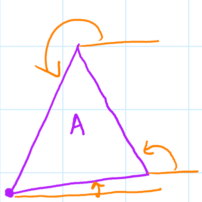
ALGORITHM: For each pair of vertices $v \in A$, $w \in B$, compute the vector sum $v+w$.

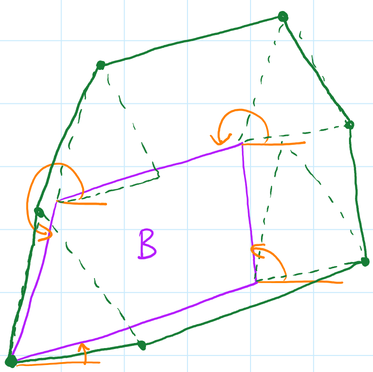
Compute the convex hull of all such vector sums.

If A has m vertices and B has n vertices, then we have to consider mn pairs of vertices.

This could be inneficient. Can we do better?

EXAMPLE:





ALGORITHM

Traversed edges of A and B simultaneously, in order of increasing angle with the horizontal axis.

Concatenate traversed edges together to form a polygon, which is $A \oplus B$.

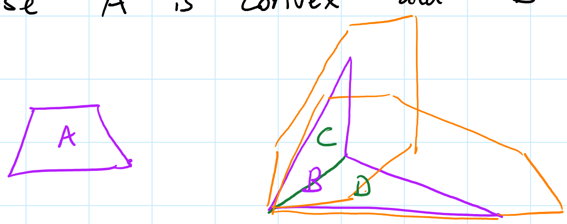
Complexity: $O(m+n)$

WHAT ABOUT NON-CONVEX POLYGONS?

OBSERVE: For any sets A, B, C:

$$\underbrace{A \oplus \underbrace{(B \cup C)}_{\text{all points in either } B \text{ or } C}}_{\text{vector sum of all } x \in A \text{ and } y \in (B \text{ or } C)} = \underbrace{(A \oplus B)}_{\text{vector sum of all } x \in A \text{ and } y \in B} \cup \underbrace{(A \oplus C)}_{\text{vector sum of all } x \in A \text{ and } y \in C}$$

Now, suppose A is convex and B is not:



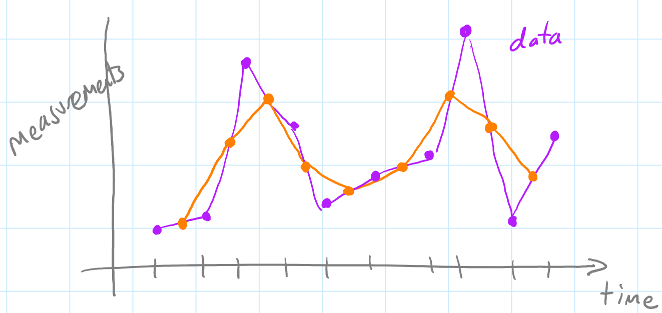
$B = C \cup D$

$$A \oplus B = (A \oplus C) \cup (A \oplus D)$$

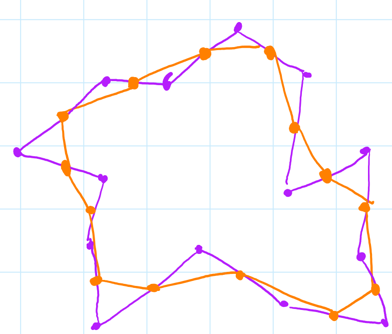
Apply algorithm for convex to each of these, then take union

CURVE SHORTENING

SITUATIONS:



How can we "smooth out" the data?



How can I "smooth out" the polygon?

IDEA: Replace each edge by its midpoint.

MIDPOINT TRANSFORMATION

What happens when we apply the midpoint transformation

... to a polygon?

Same number of vertices.

Length decreases.

... to a polygonal chain?

One less vertex after transformation.

Perimeter is reduced?

Angles get larger — how to make this precise?

If I apply the midpoint transformation to a polygon, do I always get a polygon (without self-intersections)?