Straight Skeletons and their Relation to Triangulations

EuroCG2010, Dortmund, Germany

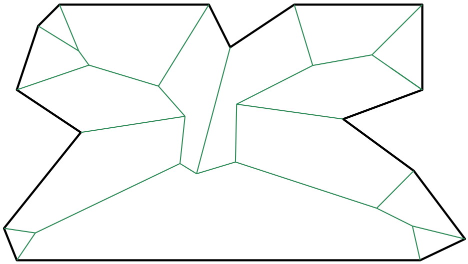
Stefan Huber Martin Held

Universität Salzburg, Austria

March 22, 2010

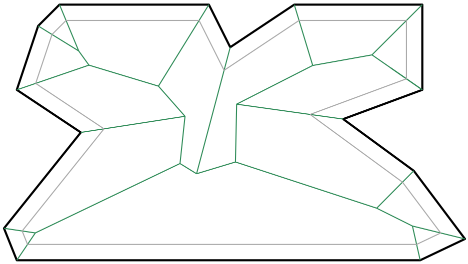
Straight skeleton of simple polygons

- Aichholzer et alii [1995]: straight skeleton of simple polygons.
- Similar to Voronoi diagram, but consists only of straight-line segments.



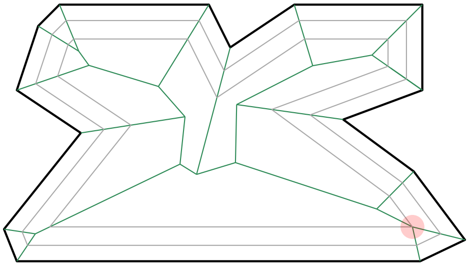
Straight skeleton of simple polygons

- Aichholzer et alii [1995]: straight skeleton of simple polygons.
- Similar to Voronoi diagram, but consists only of straight-line segments.
- Self-parallel wavefront propagation process.



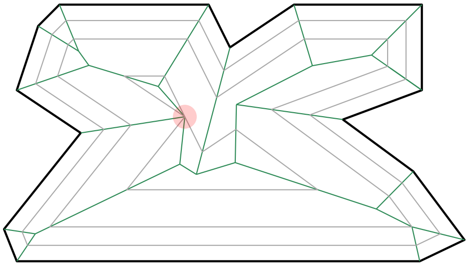
Straight skeleton of simple polygons

- Aichholzer et alii [1995]: straight skeleton of simple polygons.
- Similar to Voronoi diagram, but consists only of straight-line segments.
- Self-parallel wavefront propagation process.
- Topological changes:
  - edge events



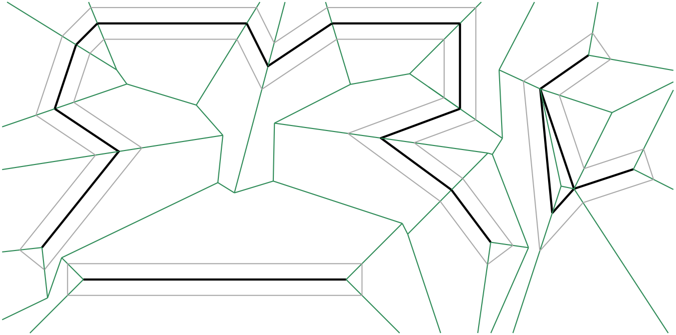
Straight skeleton of simple polygons

- Aichholzer et alii [1995]: straight skeleton of simple polygons.
- Similar to Voronoi diagram, but consists only of straight-line segments.
- Self-parallel wavefront propagation process.
- Topological changes:
  - edge events
  - split events



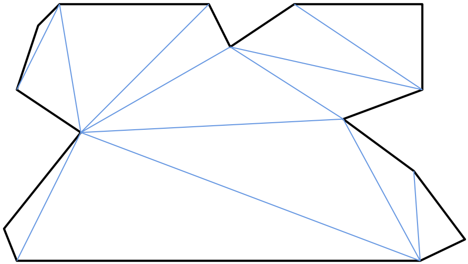
Straight skeleton of planar straight-line graphs

- Aichholzer and Aurenhammer [1998]: generalization to planar straight-line graphs.



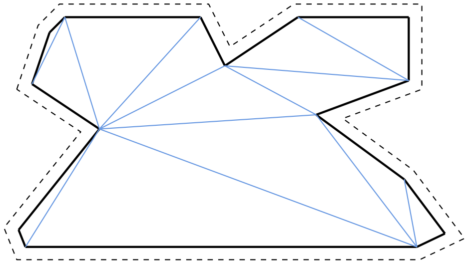
Triangulation-based algorithm

- Aichholzer and Aurenhammer [1998]: for planar straight-line graphs, but let us consider simple polygons.
- Basic idea:
  - Triangulate polygon.



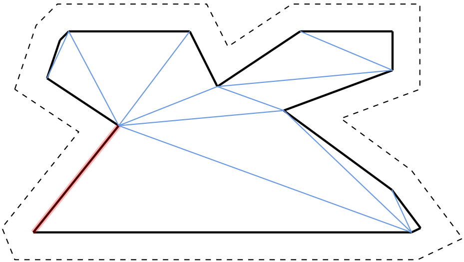
Triangulation-based algorithm

- Aichholzer and Aurenhammer [1998]: for planar straight-line graphs, but let us consider simple polygons.
- Basic idea:
  - Triangulate polygon.
  - Simulate wavefront propagation.



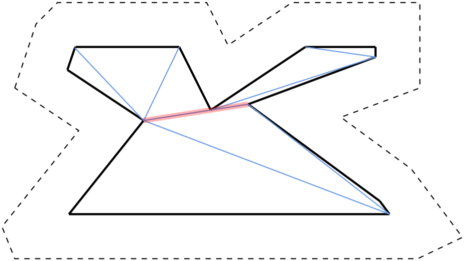
Triangulation-based algorithm

- Aichholzer and Aurenhammer [1998]: for planar straight-line graphs, but let us consider simple polygons.
- Basic idea:
  - Triangulate polygon.
  - Simulate wavefront propagation.
  - Edge and split events correspond to collapses of triangles.



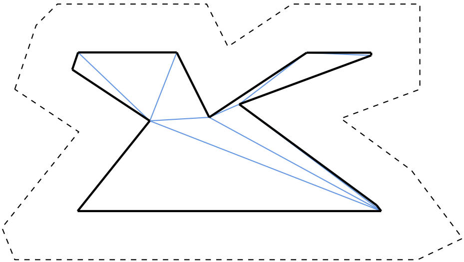
Triangulation-based algorithm

- Aichholzer and Aurenhammer [1998]: for planar straight-line graphs, but let us consider simple polygons.
- Basic idea:
  - Triangulate polygon.
  - Simulate wavefront propagation.
  - Edge and split events correspond to collapses of triangles.
  - Additional events: **flip events**. We have to maintain the triangulation by an edge flip.



Triangulation-based algorithm

- Aichholzer and Aurenhammer [1998]: for planar straight-line graphs, but let us consider simple polygons.
- Basic idea:
  - Triangulate polygon.
  - Simulate wavefront propagation.
  - Edge and split events correspond to collapses of triangles.
  - Additional events: **flip events**. We have to maintain the triangulation by an edge flip.



Triangulation-based algorithm

- Priority queue contains collapsing times of triangles.
- A single flip event costs $O(\log n)$ time.

Central question

How many flip events can occur?

- Best known bound: $O(n^3) \implies O(n^3 \log n)$ runtime.
- Still open problem: is the actual bound $O(n^2)$?

How often can diagonals reappear?

Question

How often can a diagonal reappear?

- Not every collinearity of three vertices results in a flip event.
- If a single diagonal would reappear at most $O(1)$ times, the $O(n^2)$ bound for the number of flip events would follow.

How often can diagonals reappear?

Question

How often can a diagonal reappear?

- Not every collinearity of three vertices results in a flip event.
- If a single diagonal would reappear at most $O(1)$ times, the $O(n^2)$ bound for the number of flip events would follow.

- Unfortunately, one can prove the following. . .

Lemma

There exists a polygon P with n vertices and a corresponding triangulation such that $\Omega(n)$ diagonals reappear $\Omega(n)$ times each during the wavefront propagation.

Goto proof

Can we always find good triangulations?

- Byproduct of the last lemma: polygons and corresponding triangulations with $\Omega(n^2)$ flip events.
- Choosing “better” triangulations often leads to $O(n)$ flip events.

Question

Can we always find, for any given polygon P, a triangulation that leads to $o(n^2)$, say $O(n)$, flip events?

Can we always find good triangulations?

- Byproduct of the last lemma: polygons and corresponding triangulations with $\Omega(n^2)$ flip events.
- Choosing "better" triangulations often leads to $O(n)$ flip events.

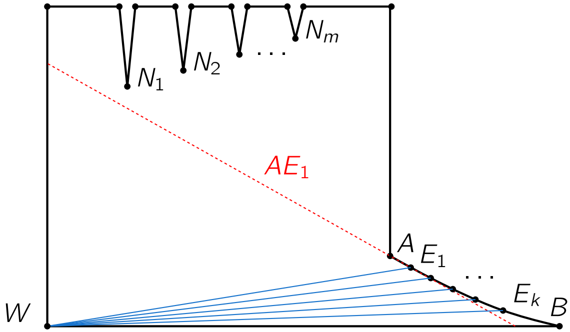
Question

Can we always find, for any given polygon P, a triangulation that leads to $o(n^2)$, say $O(n)$, flip events?

Lemma

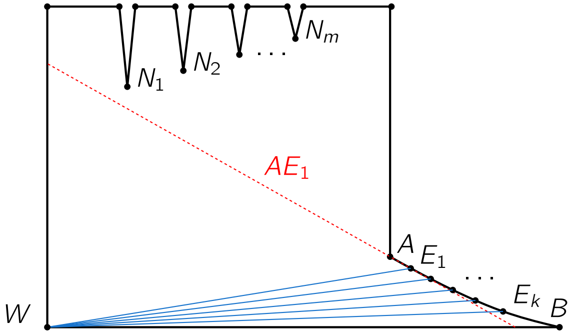
There exists a polygon with n vertices for which **every triangulation** *leads to $\Omega(n^2)$ flip events.*

Can we always find good triangulations?



- Let N_i cause $\Omega(n)$ flip events before $N_{i+1}, \dots, N_m$ cross AE_1. $\square$

Can we always find good triangulations?



- Let N_i cause $\Omega(n)$ flip events before $N_{i+1}, \ldots, N_m$ cross AE_1. $\square$
- Note: **Retriangulating** at specific favorable moments does not seem to help either! Retriangulating once saves at most $O(n)$ flip events.

Steiner triangulations without flip events

Hence, there are polygons where every triangulation leads to $\Omega(n^2)$ flip events. Even retriangulating at favorable moments does not help.

Question

Do Steiner points help to reduce the number of flip events? If yes, how should Steiner points behave during the propagation process?

Steiner triangulations without flip events

Hence, there are polygons where every triangulation leads to $\Omega(n^2)$ flip events. Even retriangulating at favorable moments does not help.

Question

Do Steiner points help to reduce the number of flip events? If yes, how should Steiner points behave during the propagation process?

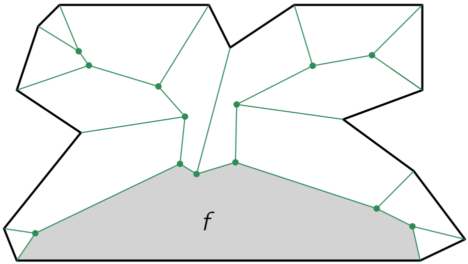
Lemma

Every simple polygon P with n vertices admits a triangulation that employs at most $n-2$ Steiner points and which is **free of flip events**.

Steiner triangulations without flip events

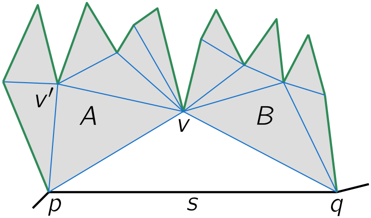
Proof.

- Constructive, but employs straight skeleton S of P.
- Insert nodes of S as Steiner points and arcs of S as initial diagonals.
- Vertices of P are prevented to cause flip events!
- It remains to triangulate each face f. Do this carefully such that Steiner points do not cause flip events!

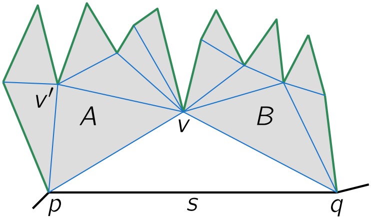


Steiner triangulations without flip events

- Note that face f is monotone w.r.t. to s.
- Reflex vertices of f only appear in the upper monotonic chain.
- If f is convex: triangulate arbitrarily.
- If f is non-convex:
  - Let v be the reflex vertex with minimum orthogonal distance to s.
  - Insert diagonals pv and qv.
  - Face f is decomposed by the triangle pvq into two parts A and B. Proceed recursively.



Steiner triangulations without flip events



- During the wavefront propagation the segments s move inwards and may be split when reaching reflex vertices of f.
- Steiner points stay in place and wait until a segment s reaches them.
- A triangle in f collapses when s reaches a node of f and hence an edge or a split event occured.
- However, note that no diagonal crosses a Steiner point. □

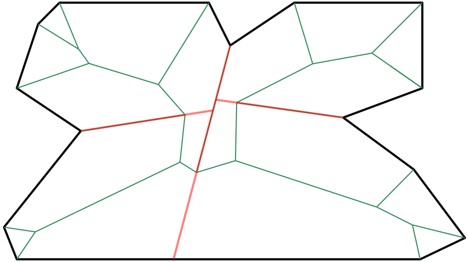
Straight skeletons based on motorcycle graphs

- Proof of last lemma does not result in a new straight skeleton algorithm.
- Convex vertices never cause flip events.
- Reflex vertices are barred from causing flip events: they move along triangulation diagonals.

Straight skeletons based on motorcycle graphs

- Proof of last lemma does not result in a new straight skeleton algorithm.
- Convex vertices never cause flip events.
- Reflex vertices are barred from causing flip events: they move along triangulation diagonals.

- The same properties hold if we replace the straight skeleton with the motorcycle graph M induced by the moving reflex vertices.



Straight skeletons based on motorcycle graphs

- Adopt the assumption of Cheng and Vigneron [2002]: no split events of higher degree.[1]
- Cheng and Vigneron [2002] showed: reflex arcs of the straight skeleton are not longer than corresponding motorcycle graph trace.
- Note that M always decomposes P into convex parts during the shrinking process.

[1] No two or more reflex events meet simultaneously in a common point.

Straight skeletons based on motorcycle graphs

- Adopt the assumption of Cheng and Vigneron [2002]: no split events of higher degree.[1]
- Cheng and Vigneron [2002] showed: reflex arcs of the straight skeleton are not longer than corresponding motorcycle graph trace.
- Note that M always decomposes P into convex parts during the shrinking process.

A simple straight skeleton algorithm:

- We do not (need to) consider a triangulation.
- But we maintain the intersection of M with the shrinking polygon P.
- We call the trace end points Steiner points.

[1] No two or more reflex events meet simultaneously in a common point.

Straight skeletons based on motorcycle graphs

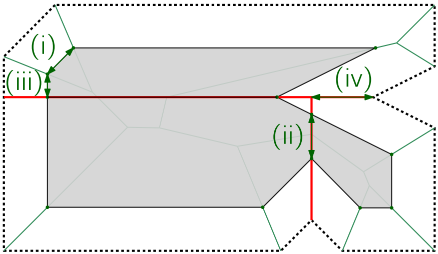
The following events may occur:

(i) edge event: two neighboring convex vertices meet.

(ii) split event: a reflex vertex meets its corresponding Steiner point.

(iii) switch event: a convex vertex meets a Steiner point an hence migrates to a neighboring convex part of P.

(iv) start event: a reflex vertex or a moving Steiner point meets a resting Steiner point, which has to start moving now.



Straight skeletons based on motorcycle graphs

- Note that **only neighboring**[2] **vertices meet** in the propagation process. Hence, it suffices to check only collisions among neighboring vertices.
- We use a priority queue, and process events in chronological order.

- In the worst case we have $O(n^2)$ switch events. All other events occur $\Theta(n)$ times.
- Every event is handled in $O(\log n)$ time since only a constant number of neighbors are affected.
- $O(n^2 \log n)$ worst case complexity.
- Still as simple as the triangulation-based algorithm.

- In practice $O(n^2)$ seems very pessimistic for the number of switch events. If only $O(n)$ switch events actually occur we get an $O(n \log n)$ runtime.

[2] Adjacent on P or on M.

Conclusion

Regarding the triangulation-based algorithm, we saw that

- $\Omega(n)$ diagonals may reappear $\Omega(n)$ times and
- there a polygons where every triangulation leads to $\Omega(n^2)$ flip events,
- but employing Steiner points allows us to eliminate all flip events.

This motivated a straight skeleton algorithm which is

- as simple as the triangulation-based algorithm;
- expected to perform as fast as the triangulation-based algorithm in practice;
- not conceptionally bounded to simple polygons and hence can be easily extended to planar straight-line graphs.

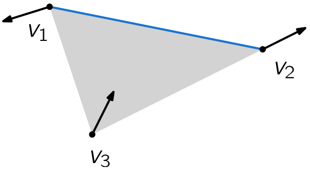
Finish

The $O(n^3)$ bound

- Flip event: a vertex v_3 crosses a diagonal between v_1 and v_2.
- Vertices v_i move constantly along a straight line, say $v_i = s_i + t \cdot u_i$.
- Vertex v_3 crosses the diagonal only if

$$\begin{vmatrix} s_{1x} + t \cdot u_{1x} & s_{2x} + t \cdot u_{2x} & s_{3x} + t \cdot u_{3x} \\ s_{1y} + t \cdot u_{1y} & s_{2y} + t \cdot u_{2y} & s_{3y} + t \cdot u_{3y} \\ 1 & 1 & 1 \end{vmatrix} = 0.$$

- Quadratic polynomial in t: at most two crossings possible for every triple v_1, v_2, v_3. □



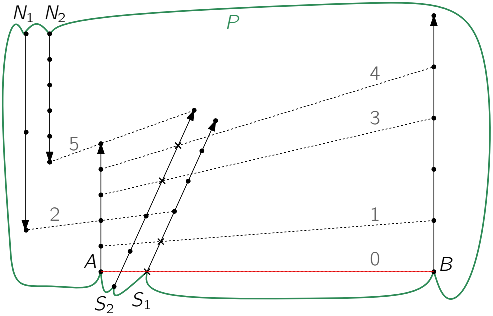
How often can diagonals reappear?

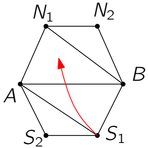
Proof.

- We construct an **appropriate geometric configuration** of moving vertices that **realizes a sequence of topological transitions** such that diagonals reappear as often as claimed.
- The proof is split into three parts:
  1. One diagonal reappears twice.
  2. One diagonal reappears $\Omega(n)$ times.
  3. $\Omega(n)$ diagonals reappear $\Omega(n)$ times each.

How often can diagonals reappear?

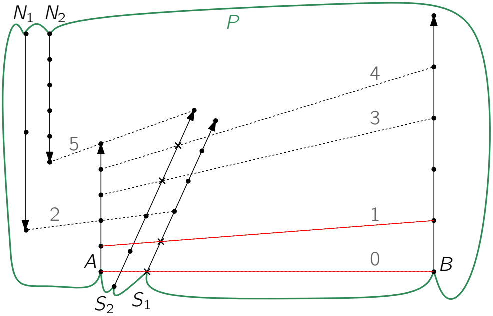
Part 1: Diagonal AB reappears twice.

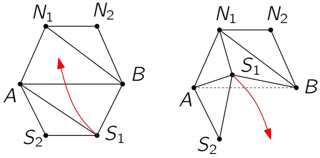




How often can diagonals reappear?

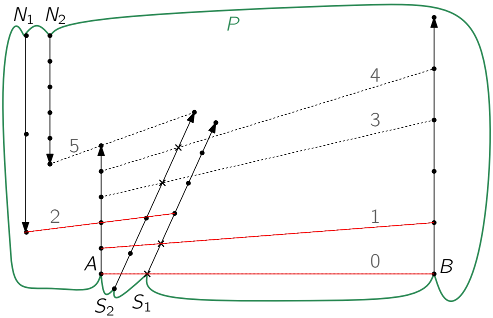
Part 1: Diagonal AB reappears twice.

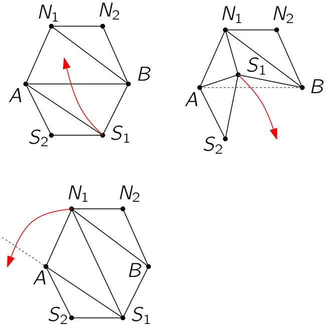




How often can diagonals reappear?

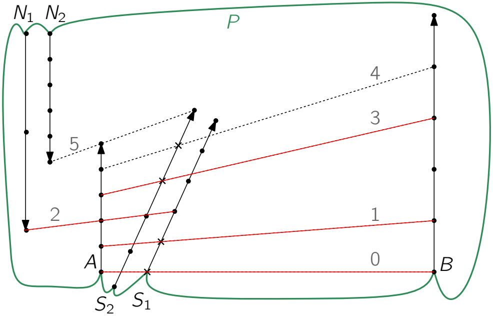
Part 1: Diagonal AB reappears twice.

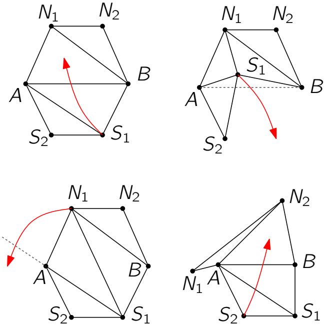




How often can diagonals reappear?

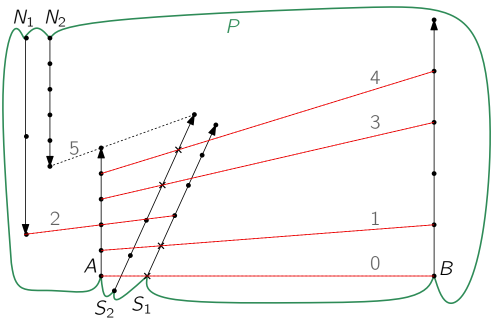
Part 1: Diagonal AB reappears twice.

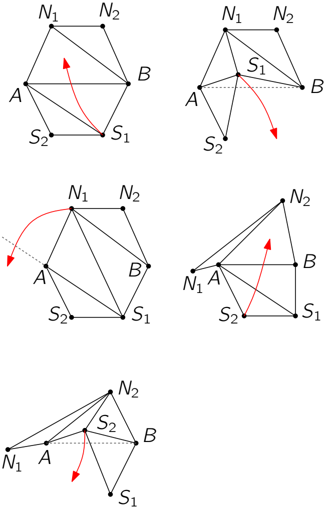




How often can diagonals reappear?

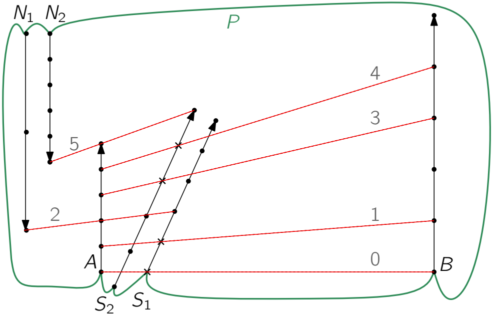
Part 1: Diagonal AB reappears twice.

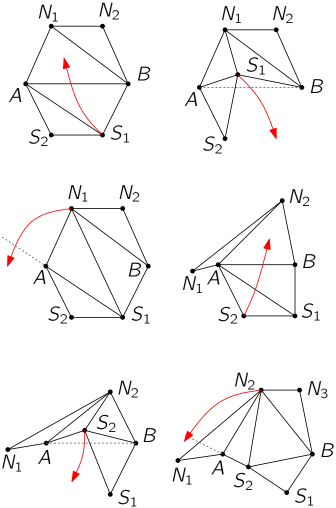




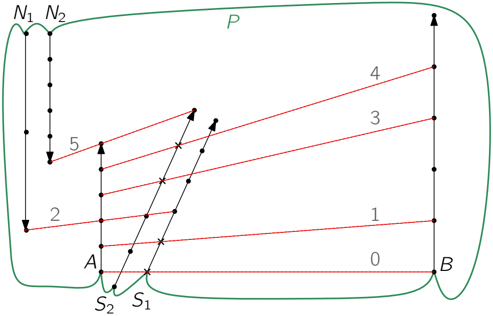
How often can diagonals reappear?

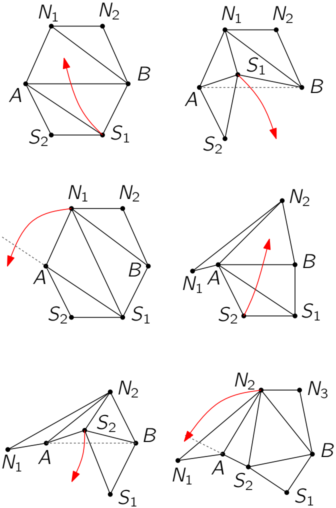
Part 1: Diagonal AB reappears twice.





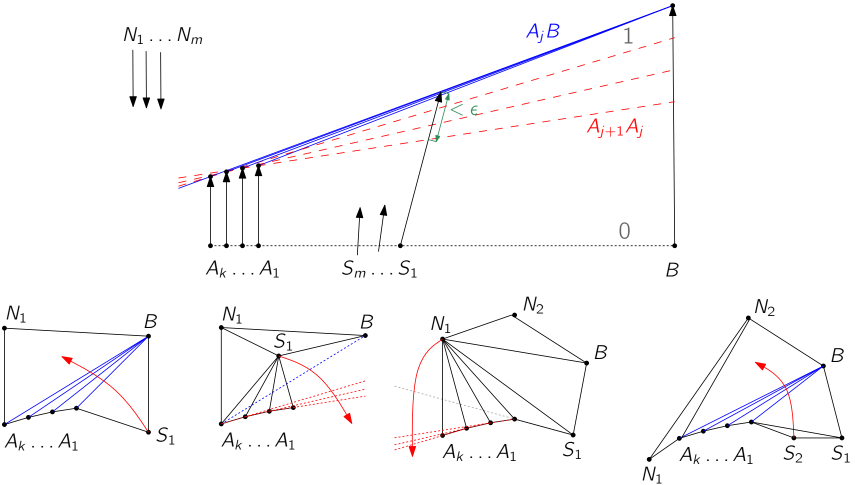
Part 2: Add vertices $S_3, \ldots, S_m$ and $N_1, \ldots, N_m$.





How often can diagonals reappear?

Part 3: Add vertices $A_2, \ldots, A_k$.



□

Return