Bowyer-Watson algorithm on a hyperbolic surface

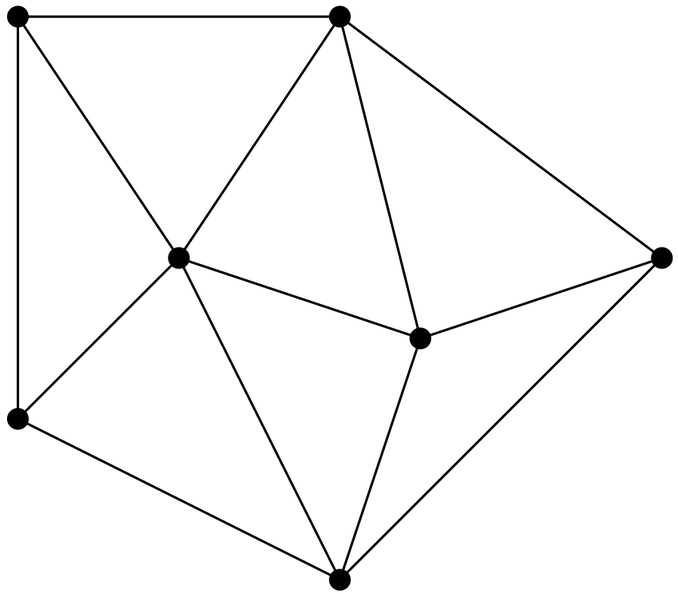
Dorian PERROT
joint work with Vincent DESPRE and Marc POUGET

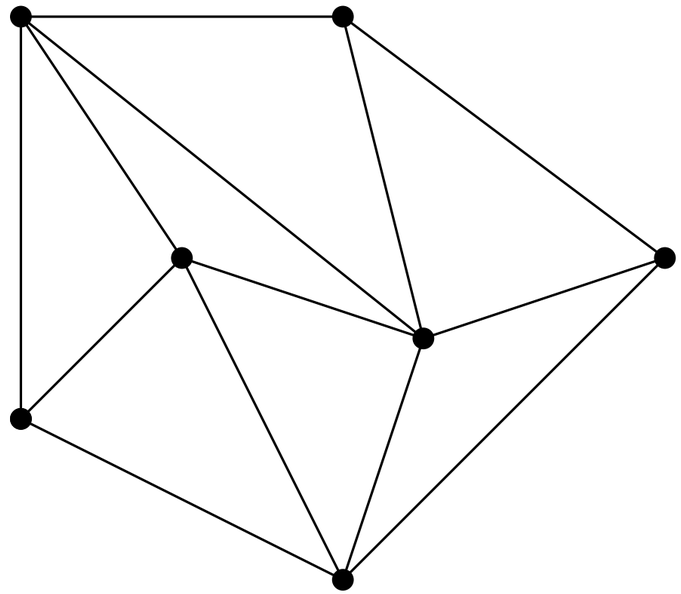
JGA Roscoff November 2025

- **Delaunay triangulation and Bowyer-Watson algorithm**
- Drawing on a hyperbolic surface
- Bowyer-Watson algorithm on a hyperbolic surface

Delaunay triangulation

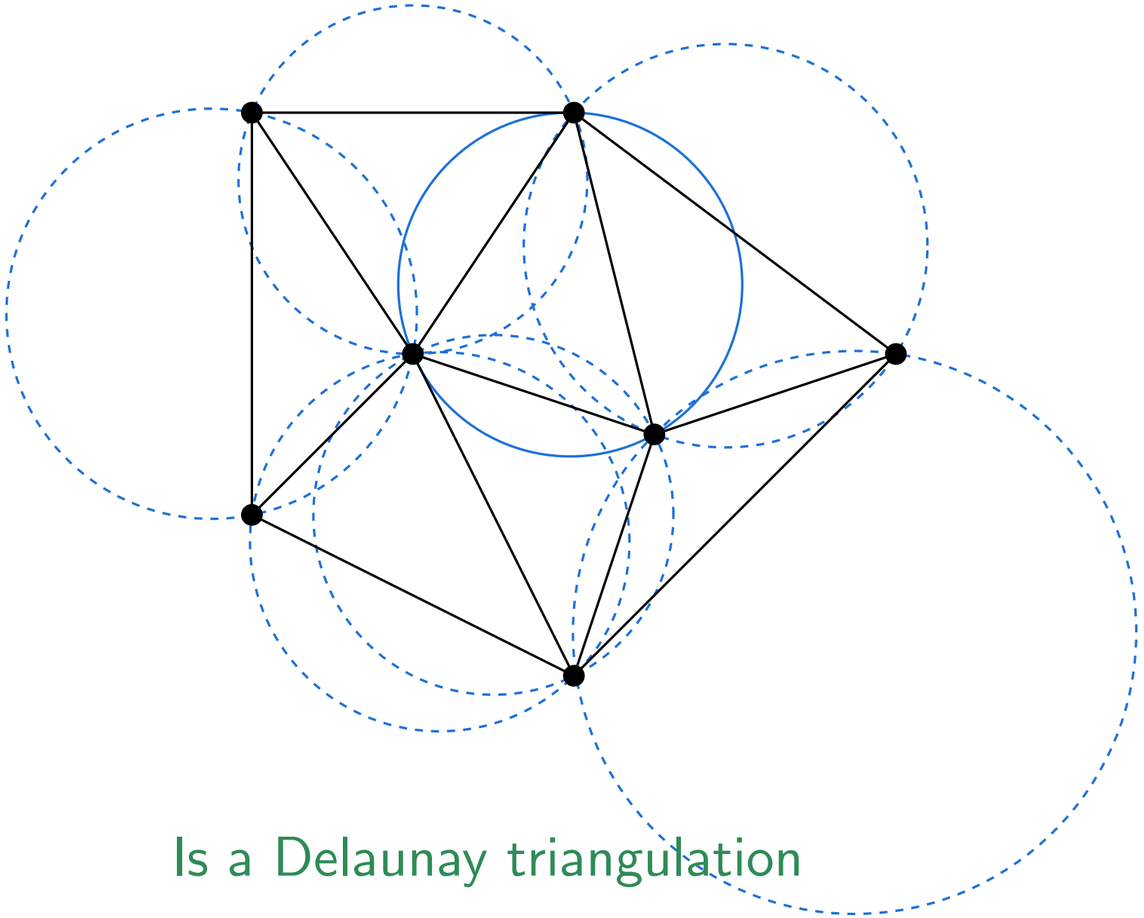
Def : a triangulation in which every circumscribed disk of triangle does not contain any of triangulation's vertices



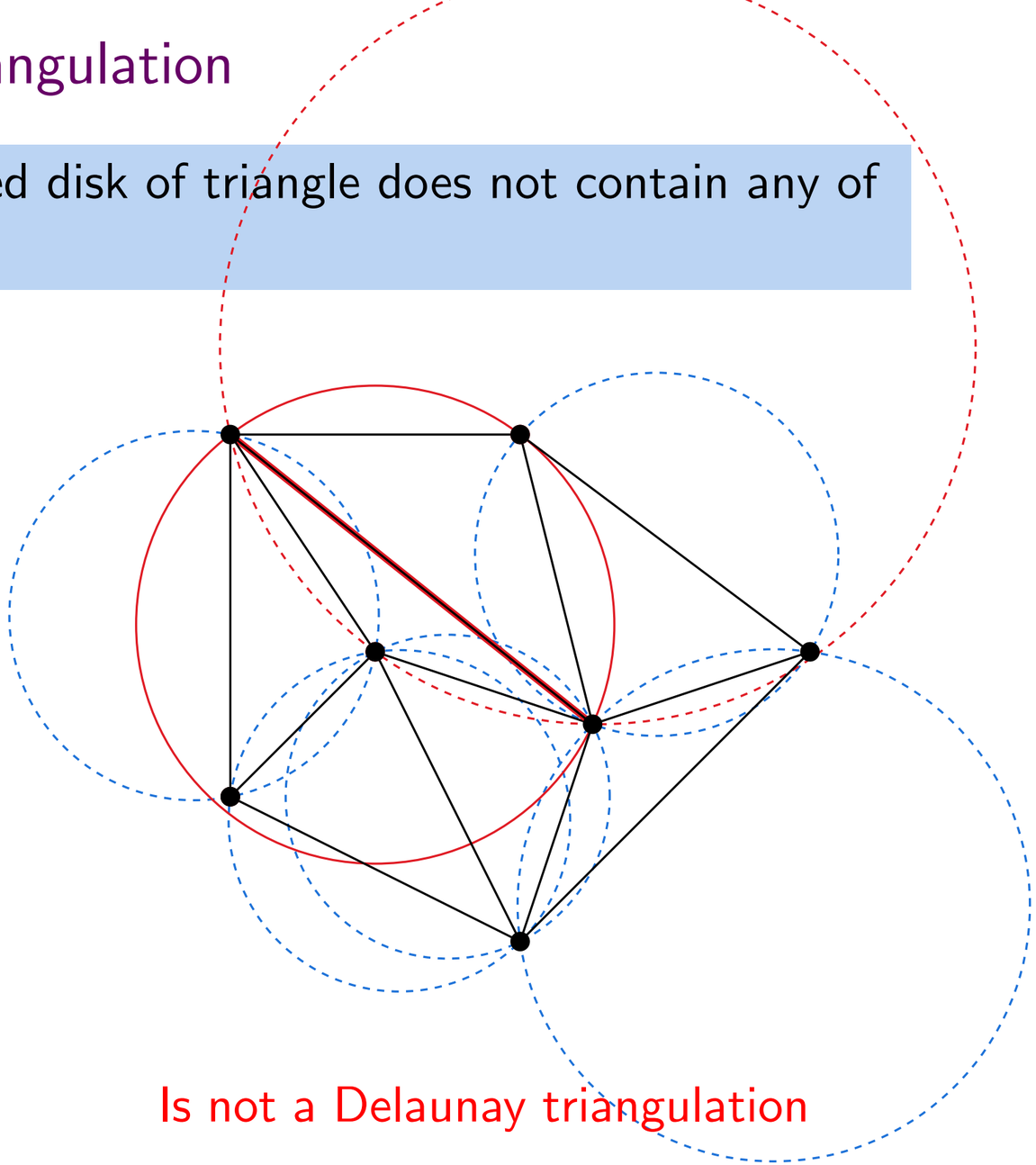


Delaunay triangulation

Def : a triangulation in which every circumscribed disk of triangle does not contain any of triangulation's vertices



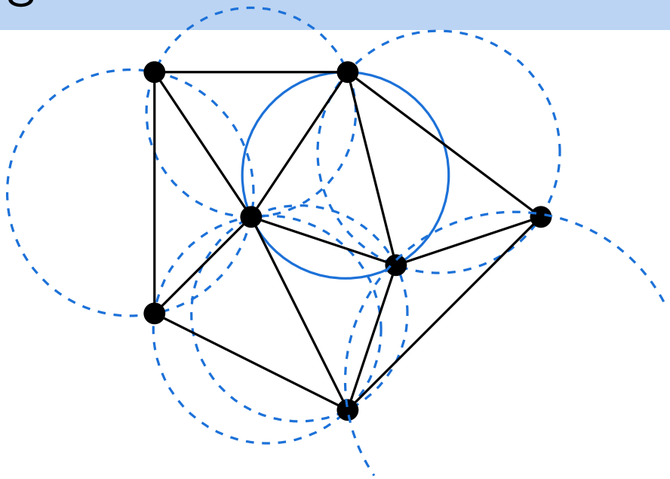
Is a Delaunay triangulation



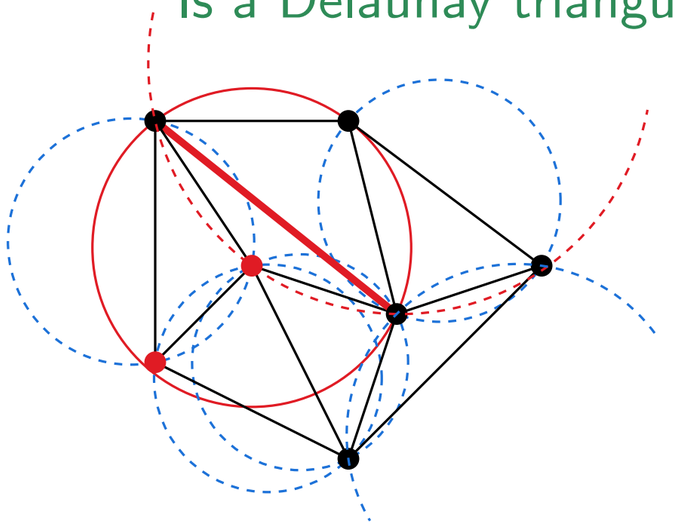
Is not a Delaunay triangulation

Delaunay triangulation

Def : a triangulation in which every circumscribed disk of triangle does not contain any of triangulation's vertices



Is a Delaunay triangulation

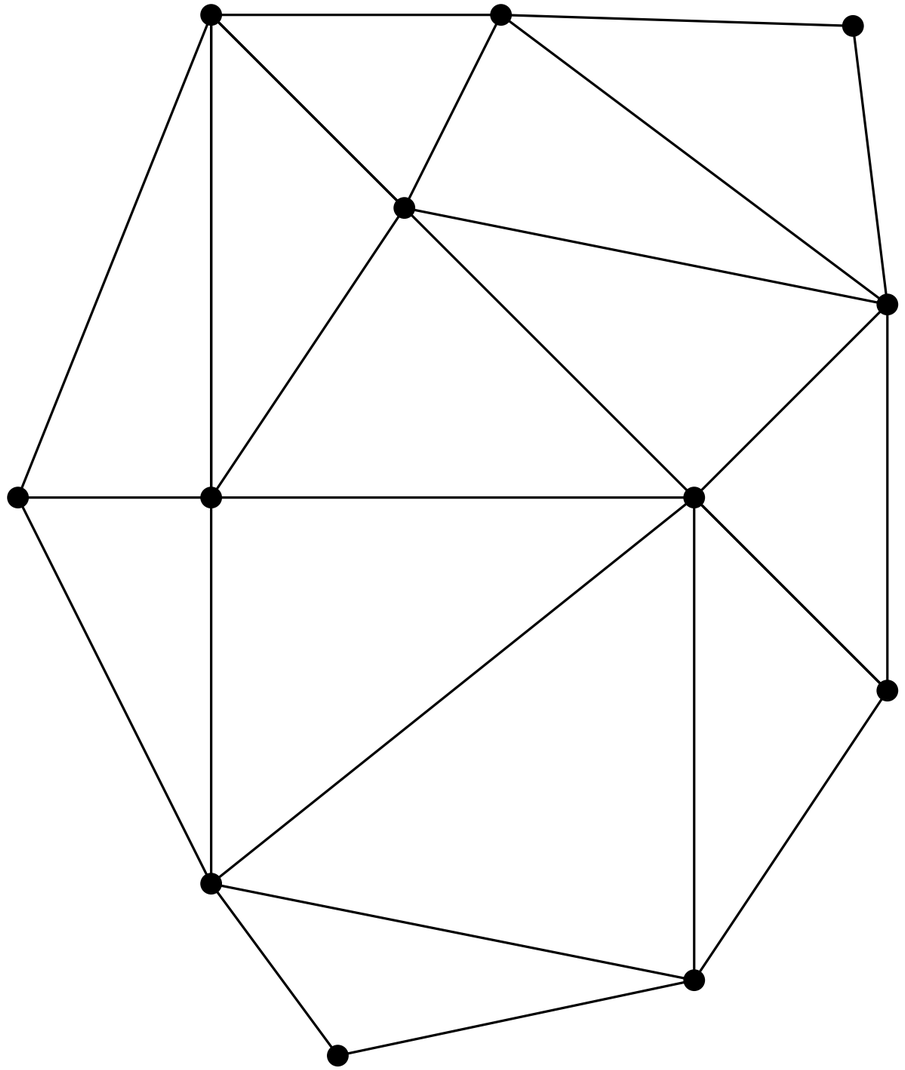


Is not a Delaunay triangulation

- Flip algorithm in Euclidean plane [La71].
  - Generalised to a hyperbolic surface [DST24].
- Bowyer-Watson algorithm. [Bo81].
  - Extended to Bolza surface (most symetric surface of genus 2) [IT17]
  - Generalization of this algorithm to all hyperbolic surfaces.

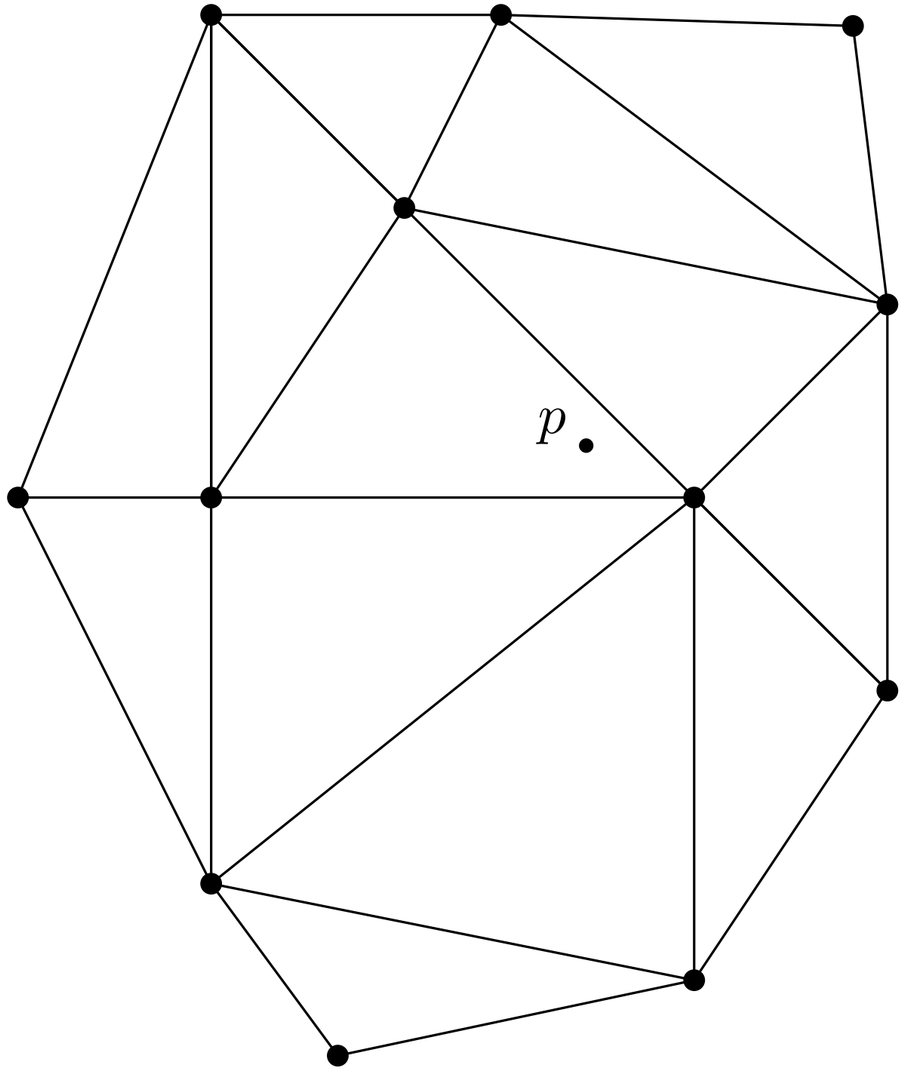
Bowyer-Watson algorithm

→ An incremental algorithm that computes a Delaunay triangulation of a set of points.



Bowyer-Watson algorithm

⟶ An incremental algorithm that computes a Delaunay triangulation of a set of points.

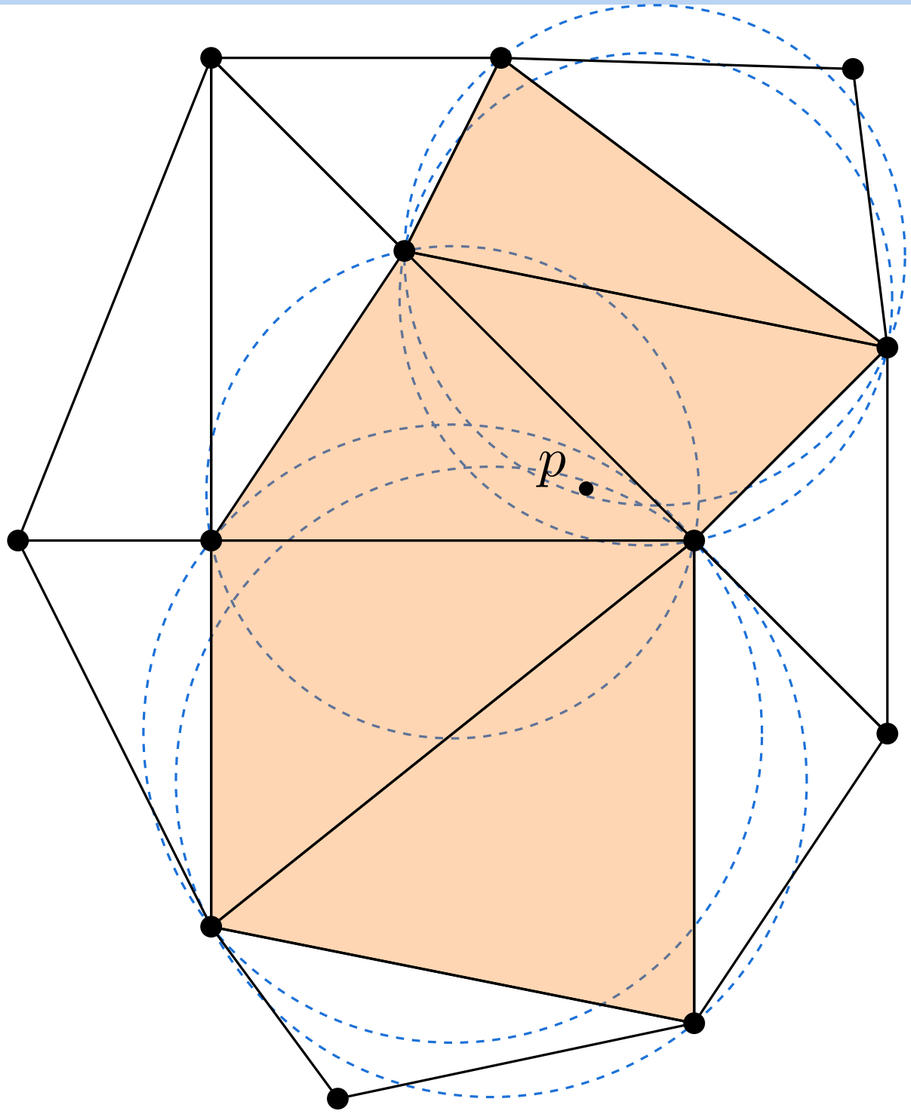


Insertion step :

1) Insert a point p

Bowyer-Watson algorithm

→ An incremental algorithm that computes a Delaunay triangulation of a set of points.



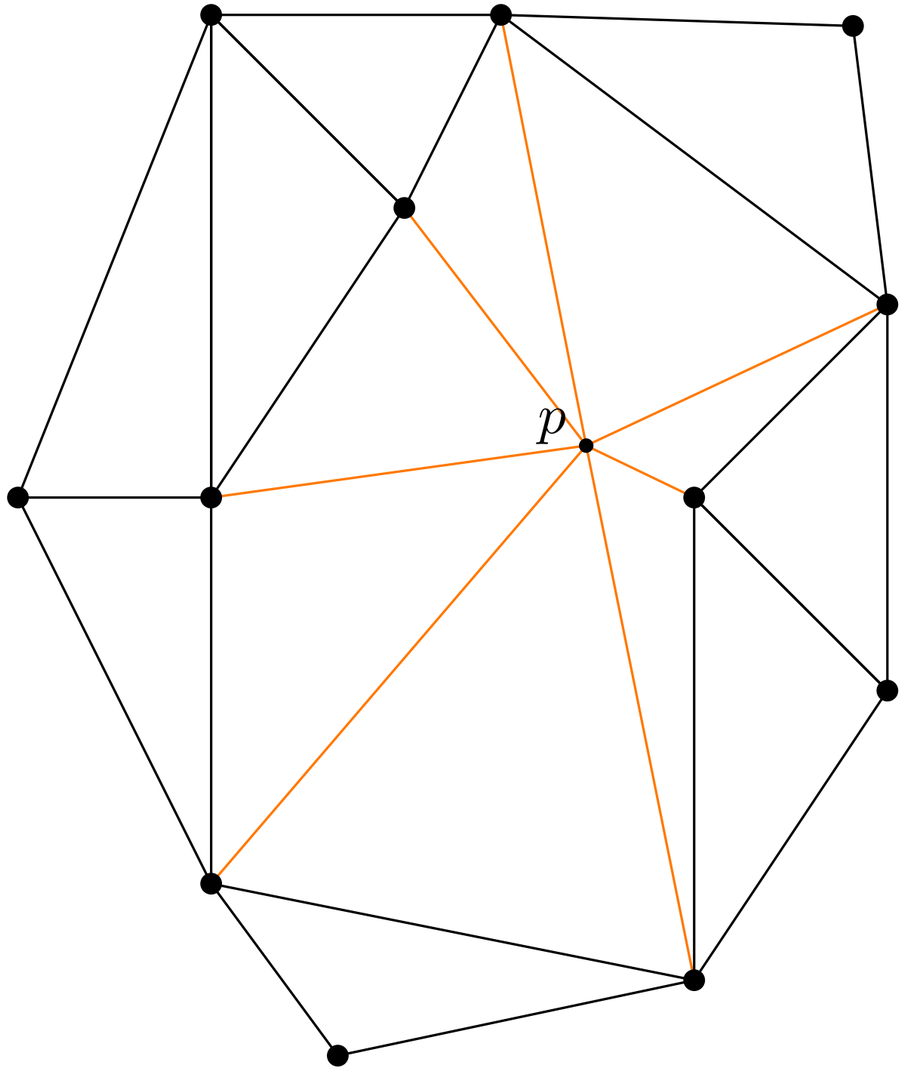
Insertion step :

1) Insert a point p

2) Compute triangles whose circumscribed disk contains p : the conflict zone.

Bowyer-Watson algorithm

⟶ An incremental algorithm that computes a Delaunay triangulation of a set of points.



Insertion step :

1) Insert a point p

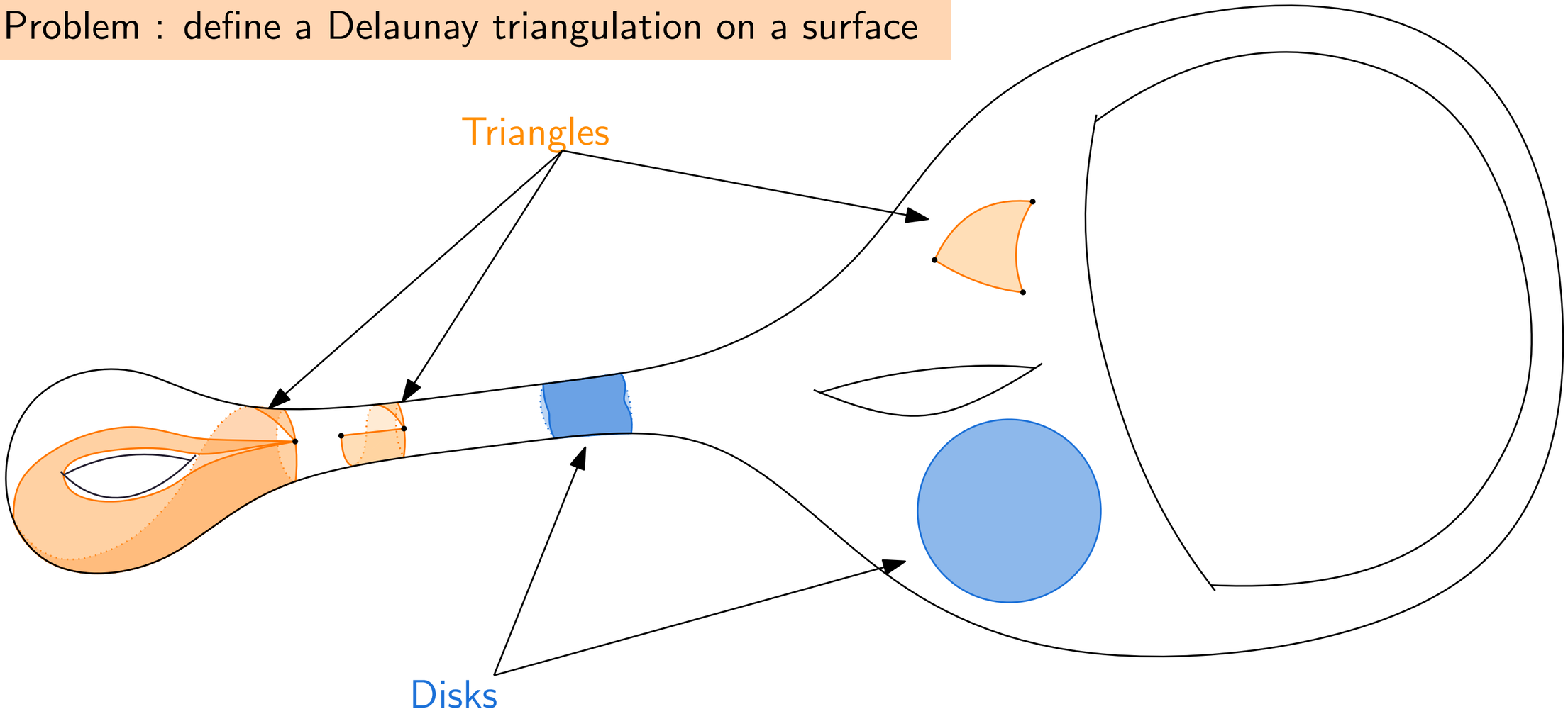
2) Compute triangles whose circumscribed disk contains p : the conflict zone.

3) Connect p to vertices of these triangles.

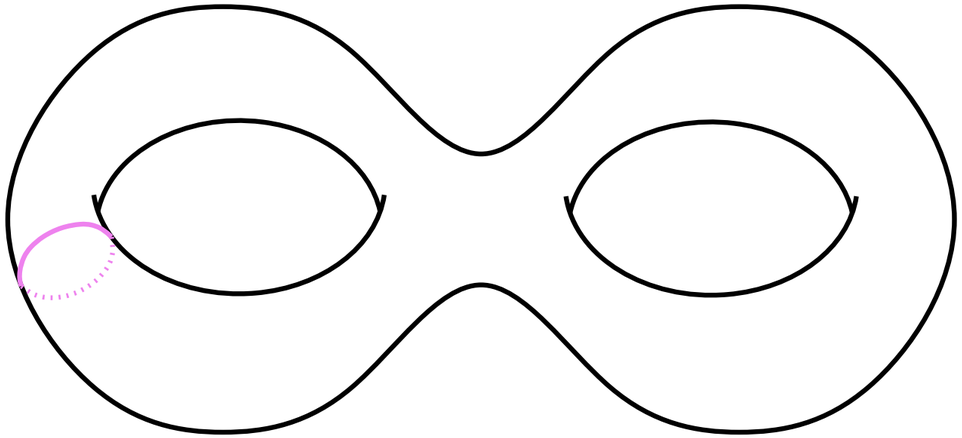
- Delaunay triangulation and Bowyer-Watson algorithm
- **Drawing on a hyperbolic surface**
- Bowyer-Watson algorithm on a hyperbolic surface

Weird triangles and weird circles

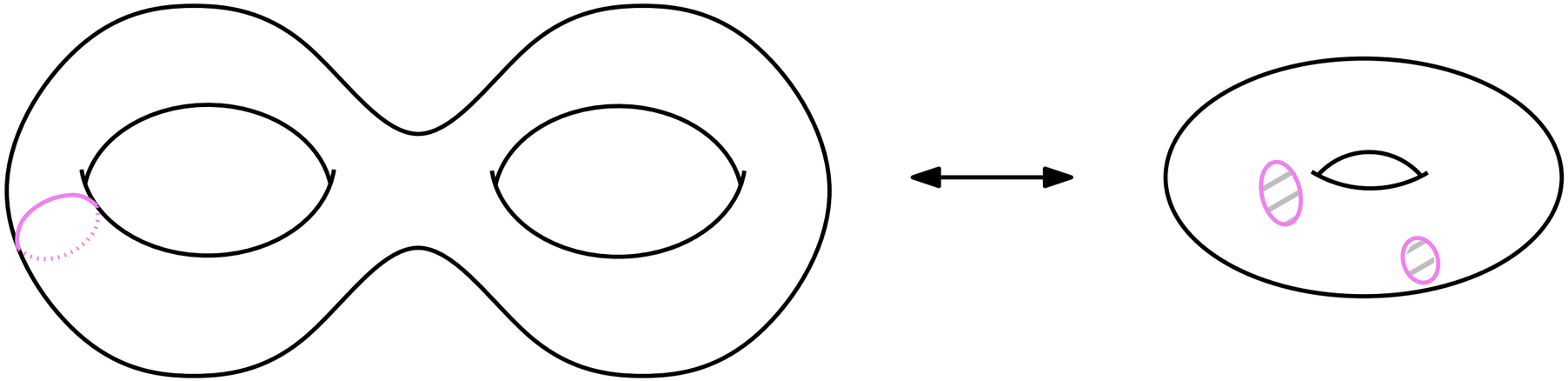
Problem : define a Delaunay triangulation on a surface



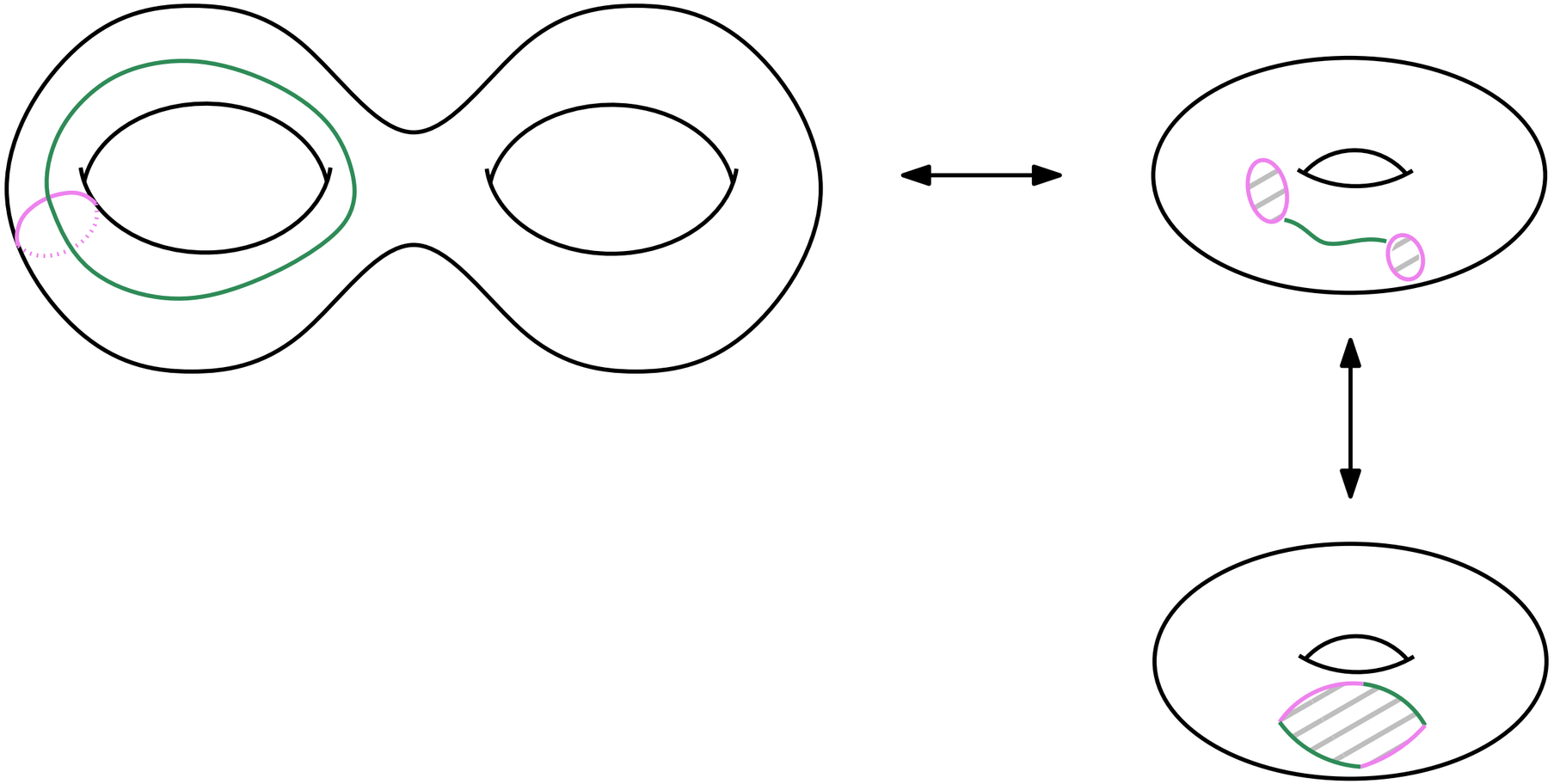
Drawing on a hyperbolic surface



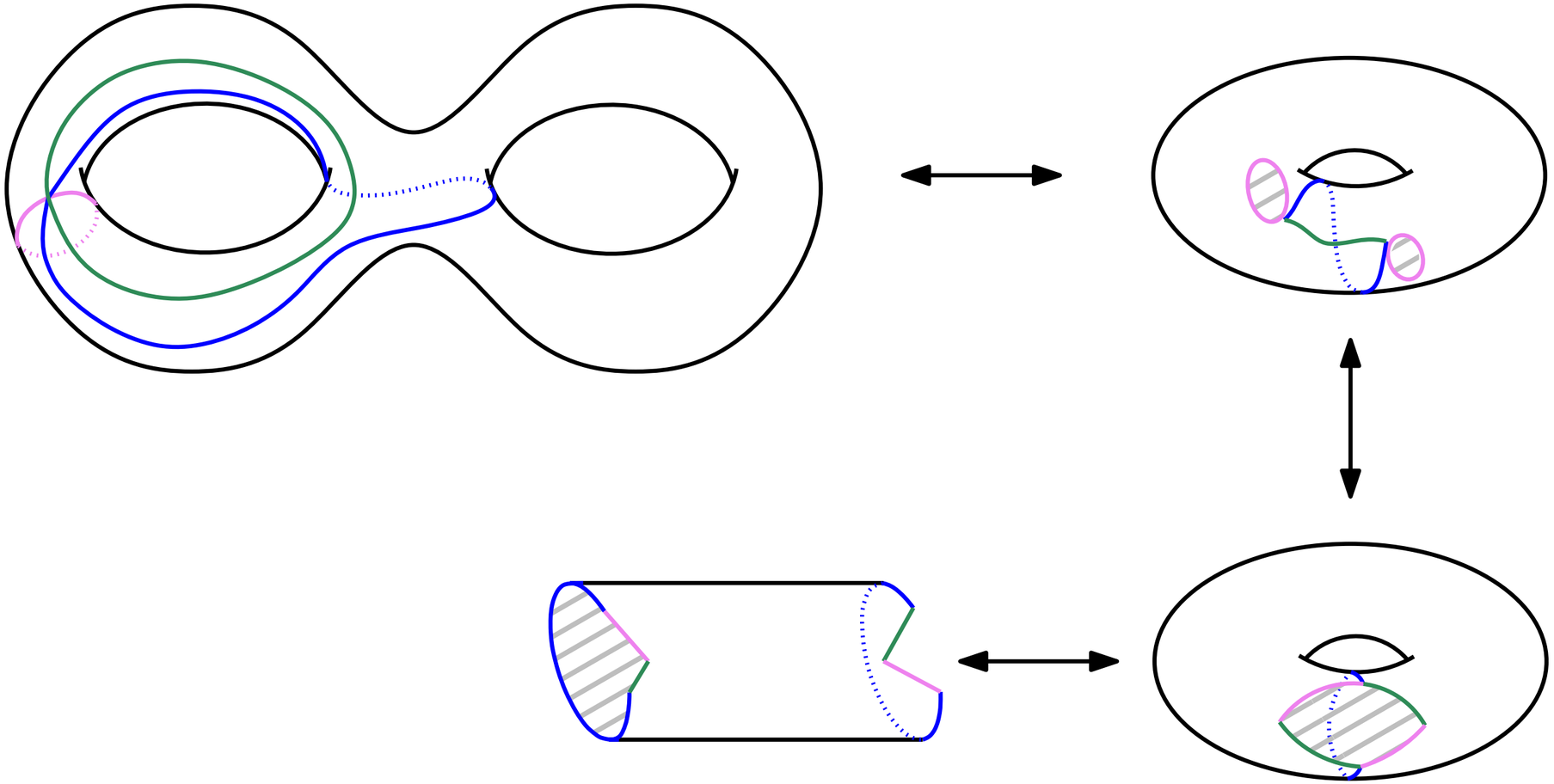
Drawing on a hyperbolic surface



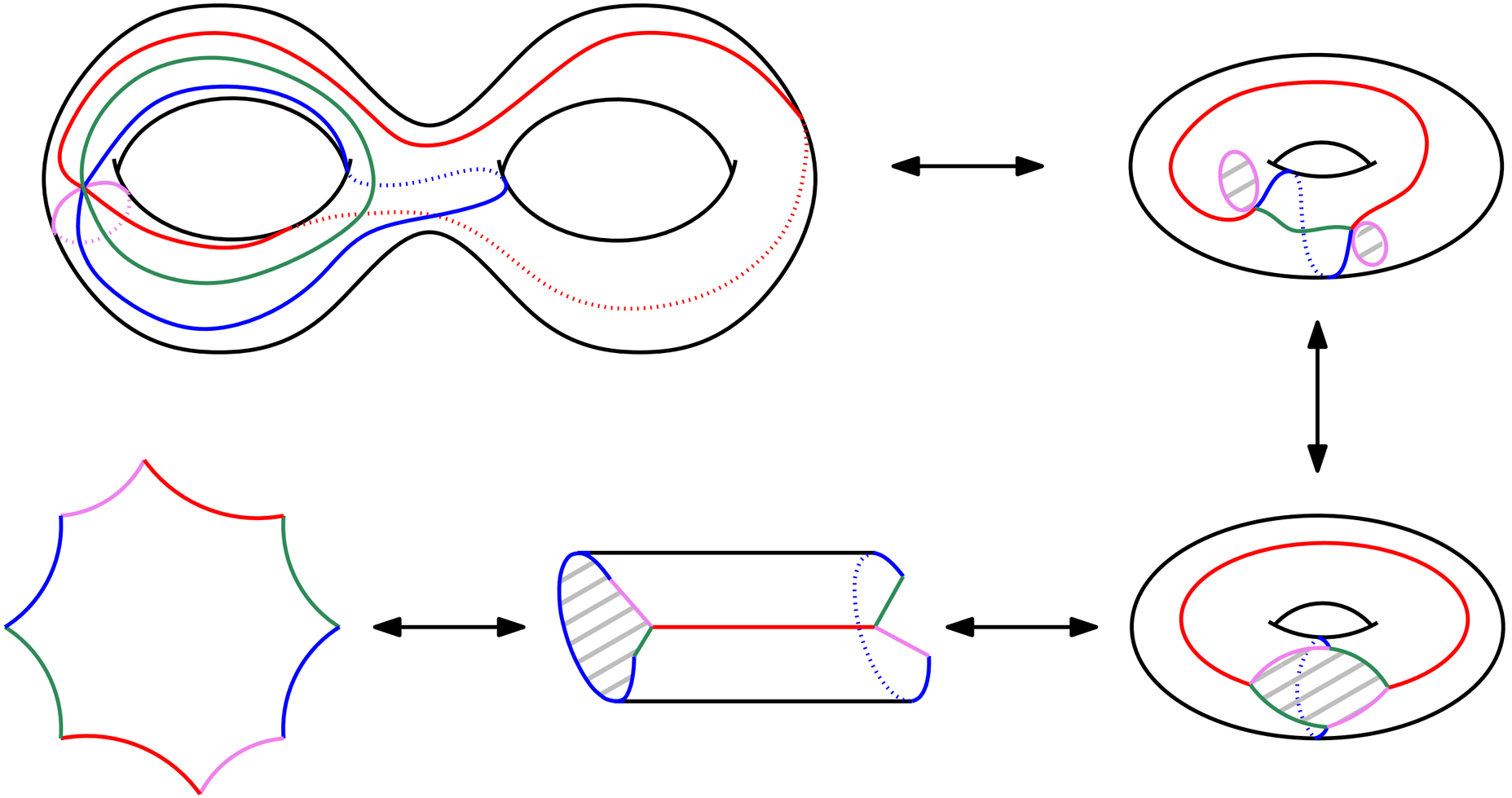
Drawing on a hyperbolic surface



Drawing on a hyperbolic surface

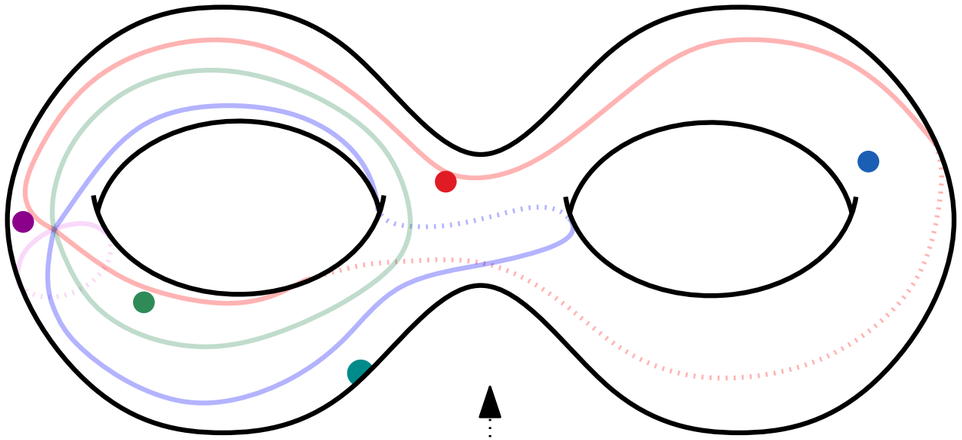


Drawing on a hyperbolic surface

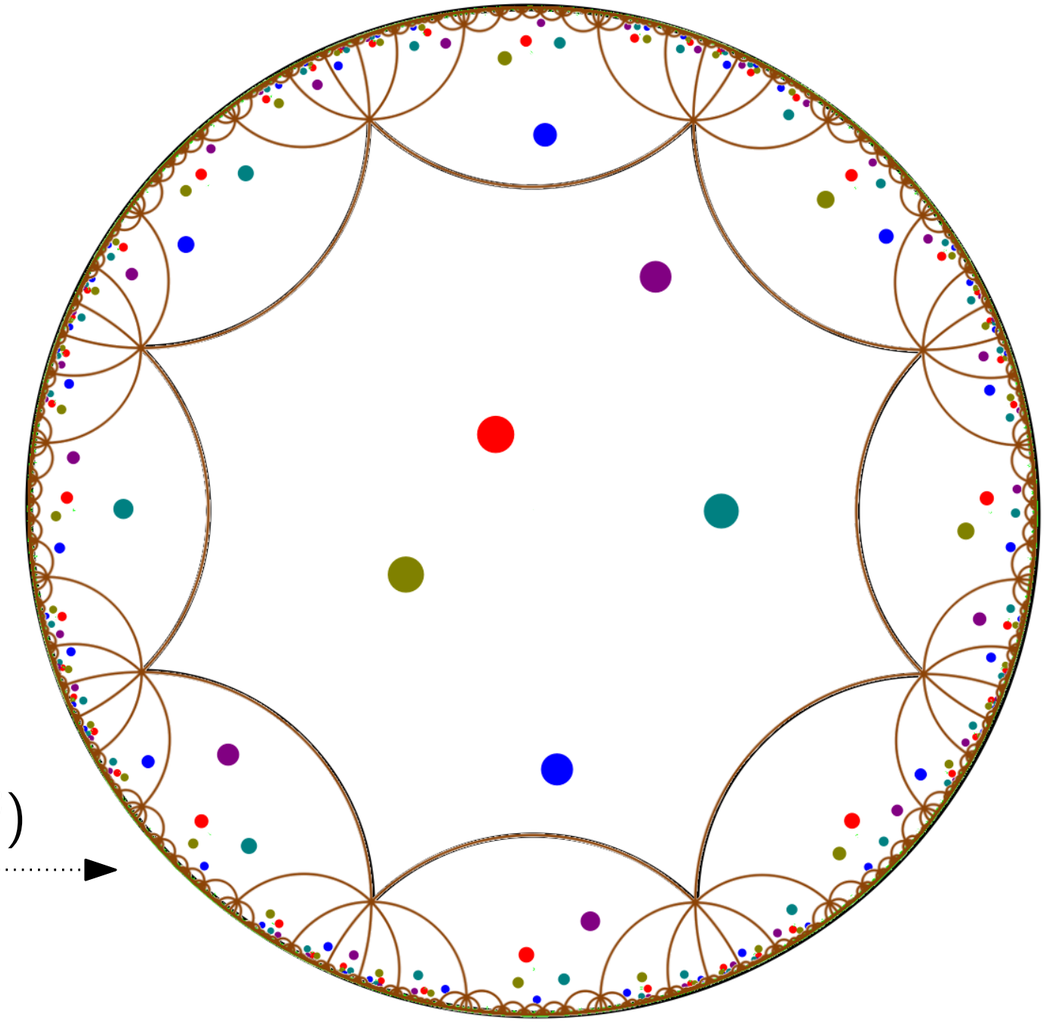


Fundamental polygon

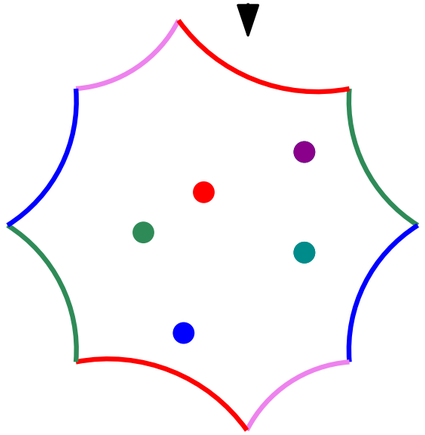
Drawing on a hyperbolic surface



Universal cover

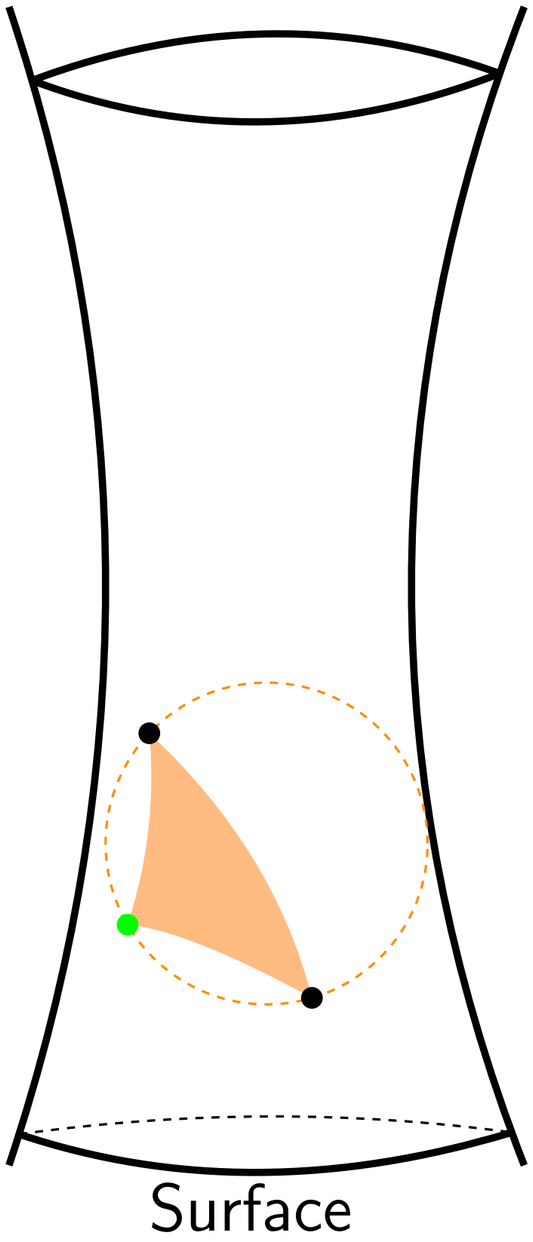


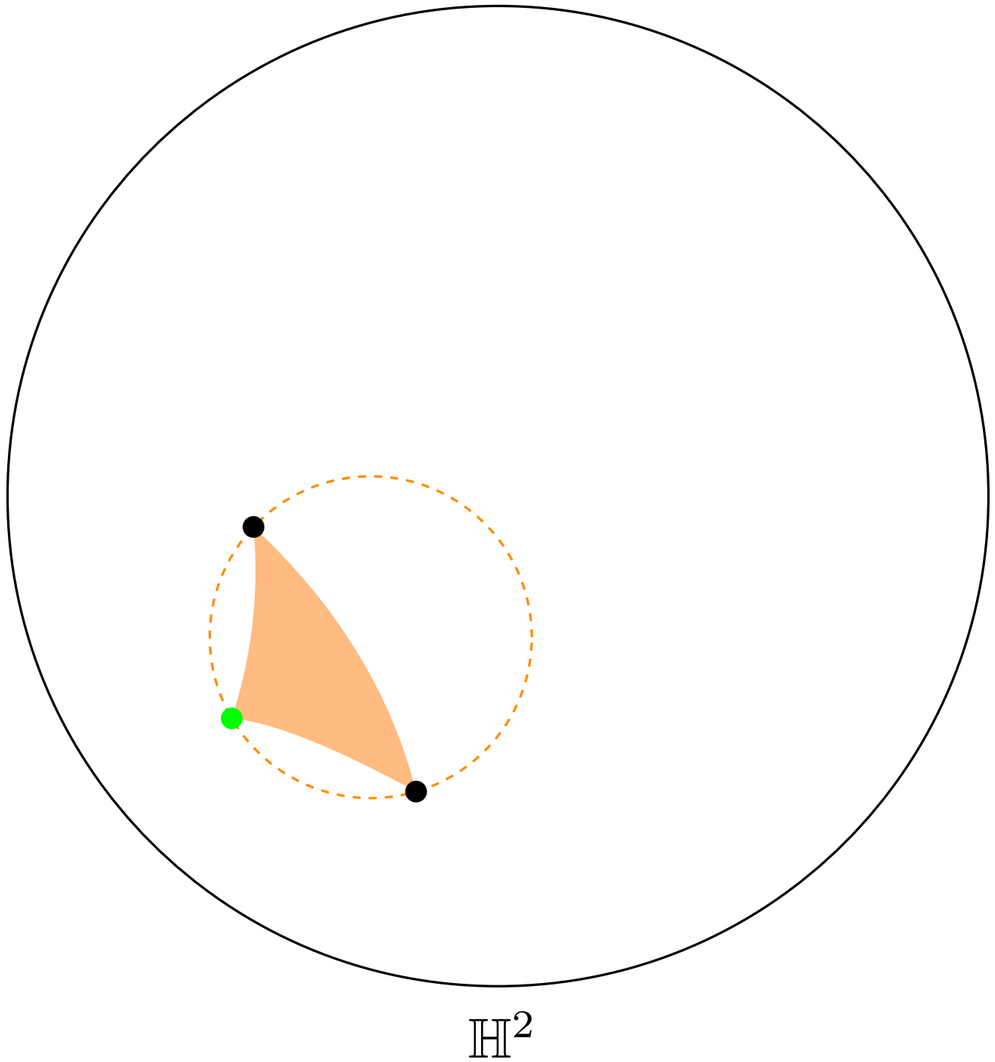
"Unfolding" the surface (Fundamental polygon)



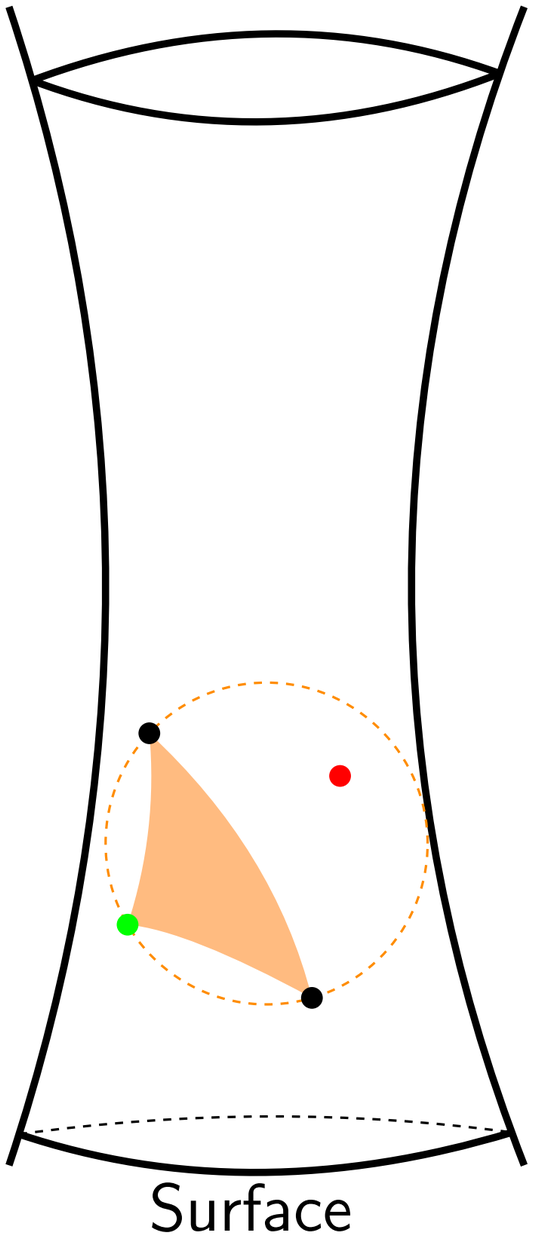
Hyperbolic plane ($\mathbb{H}^2$) tesselation

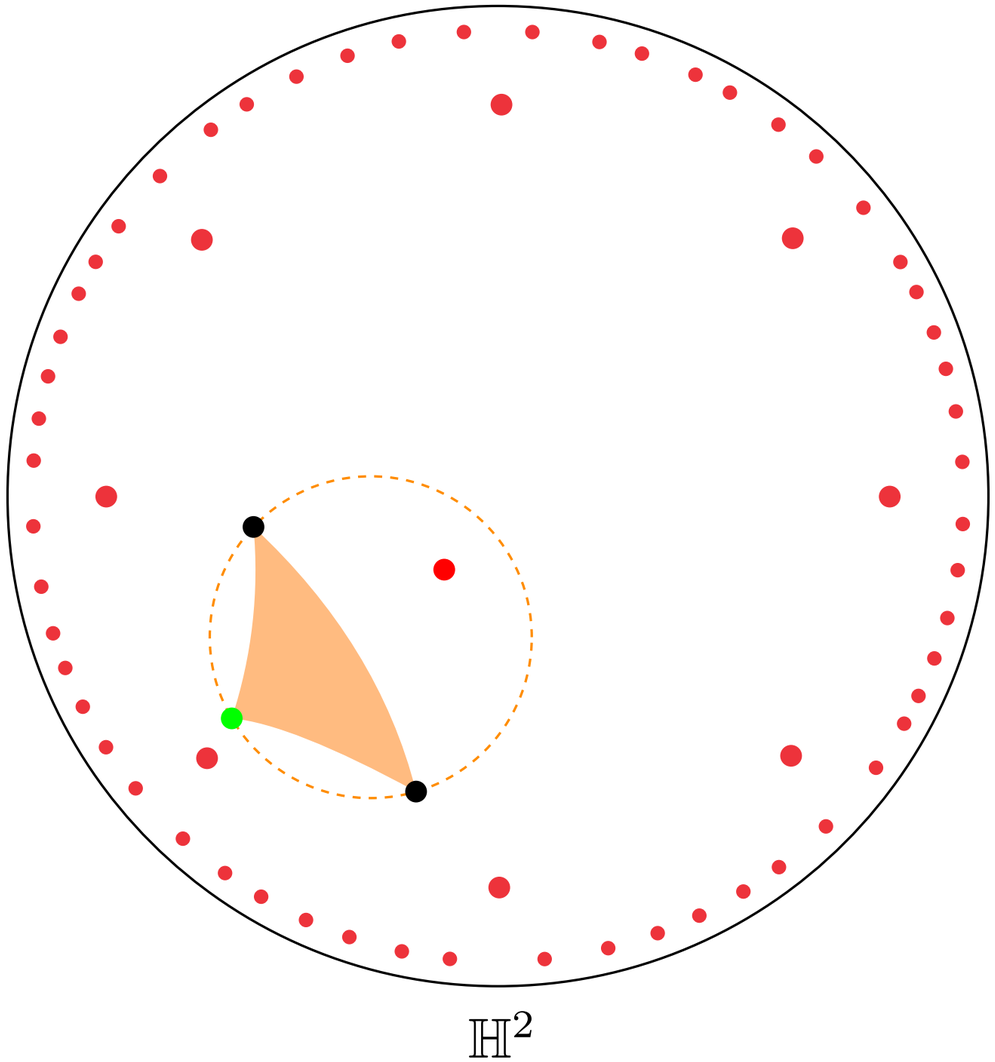
Which way ?



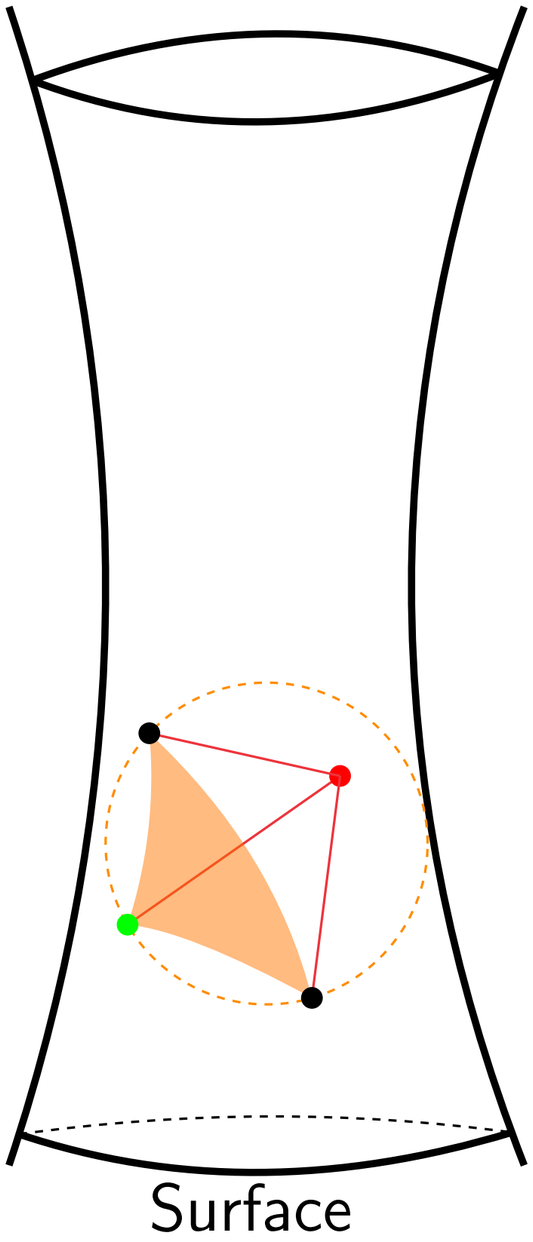


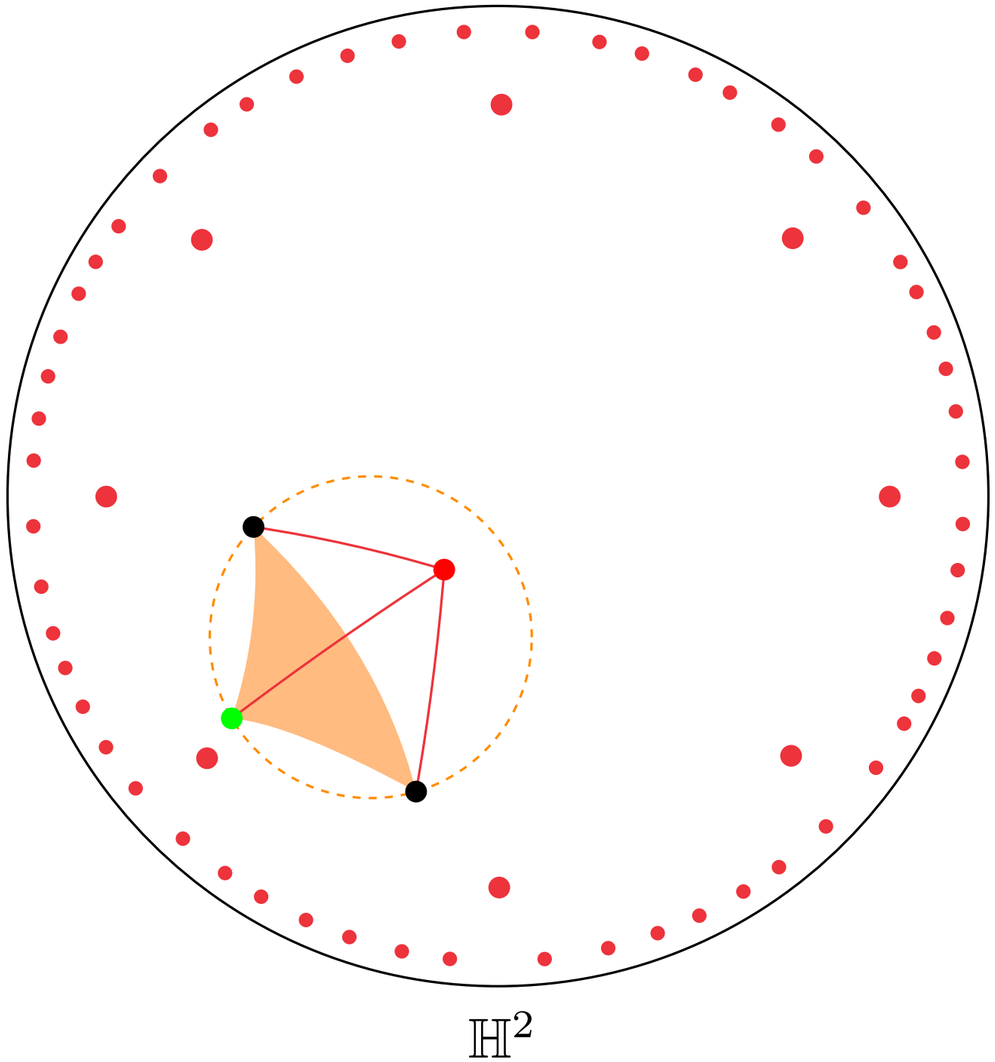
Which way ?



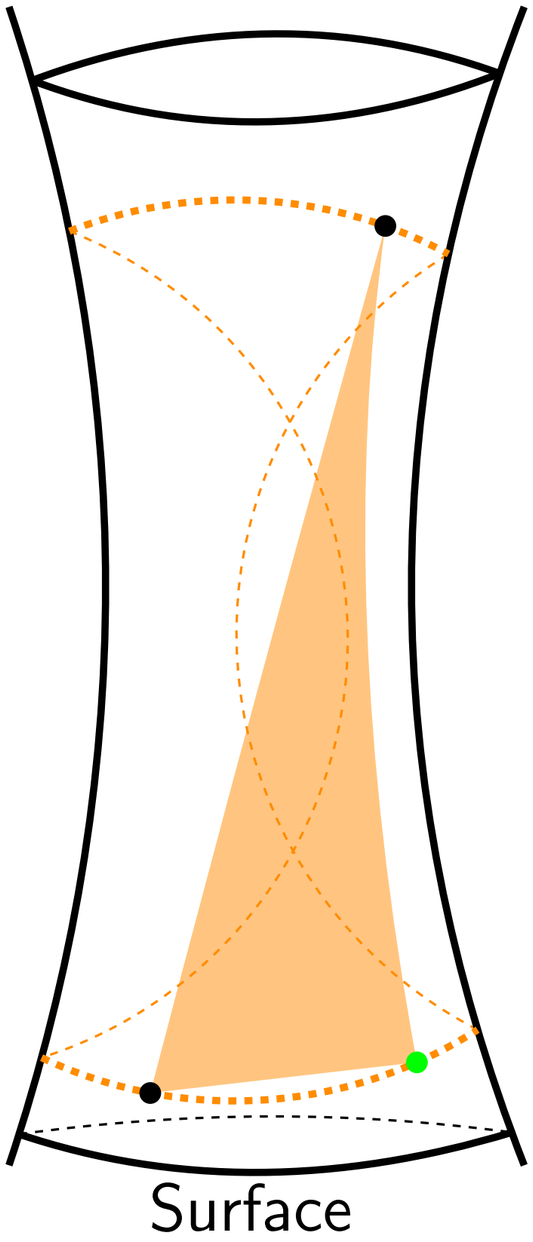


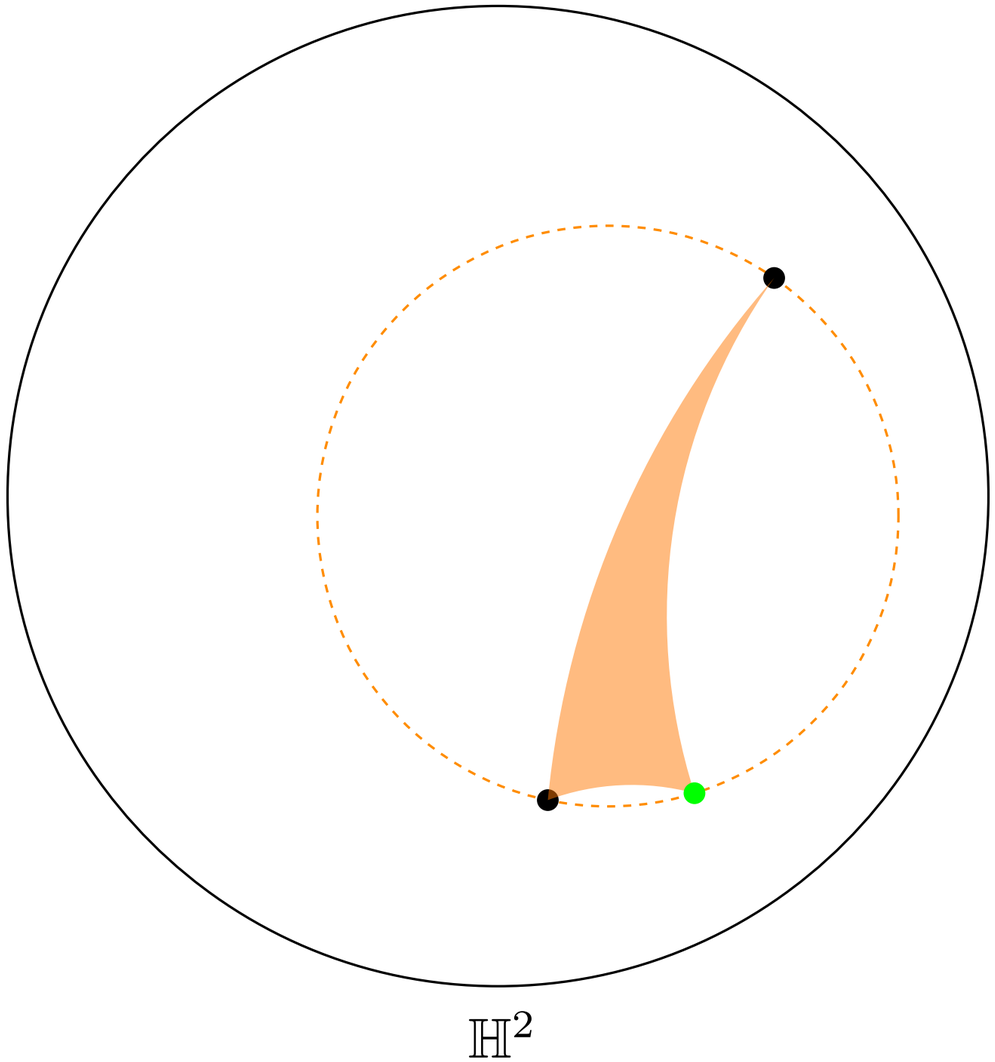
Which way ?



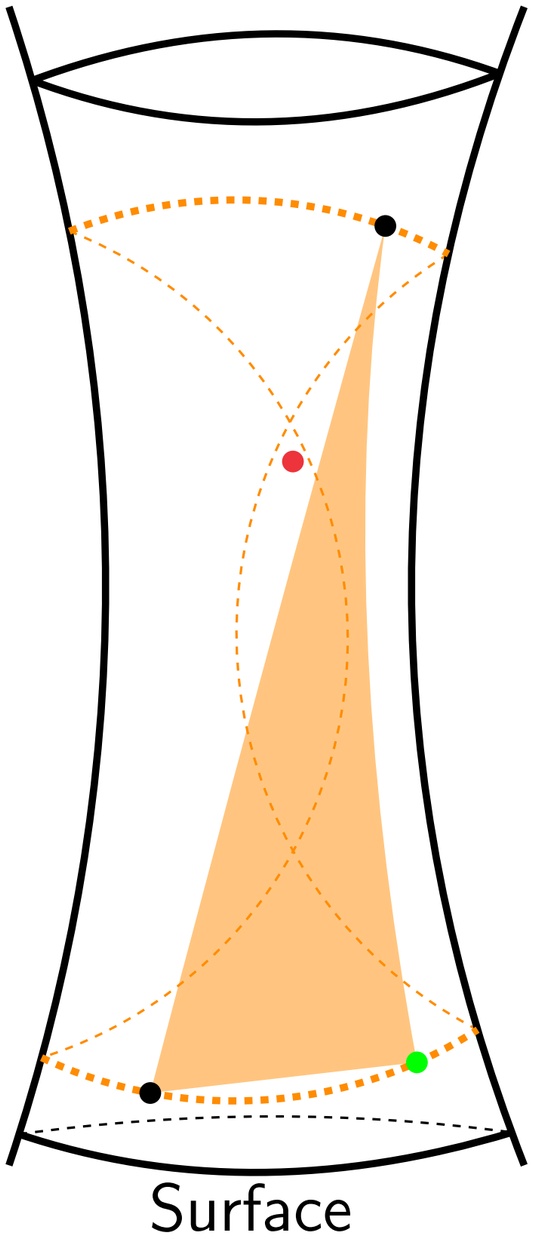


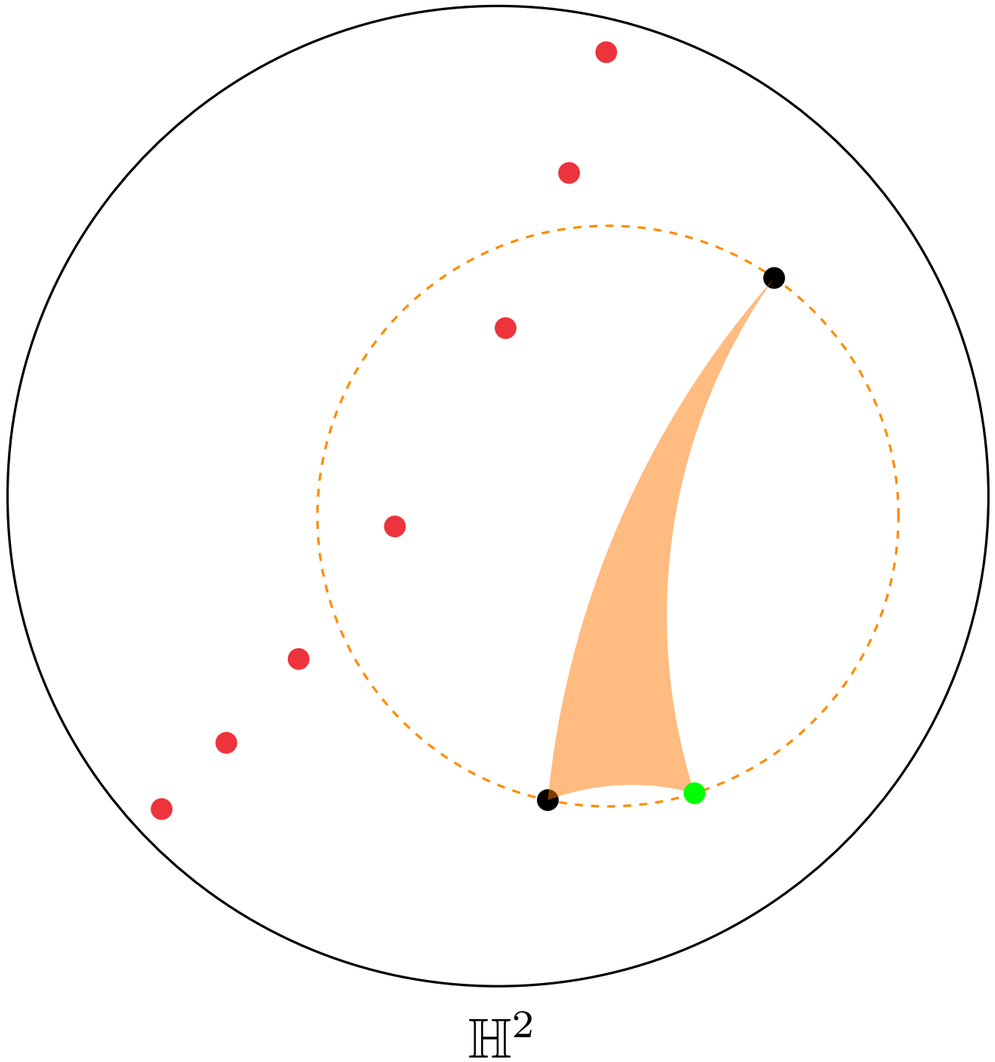
Which way ?



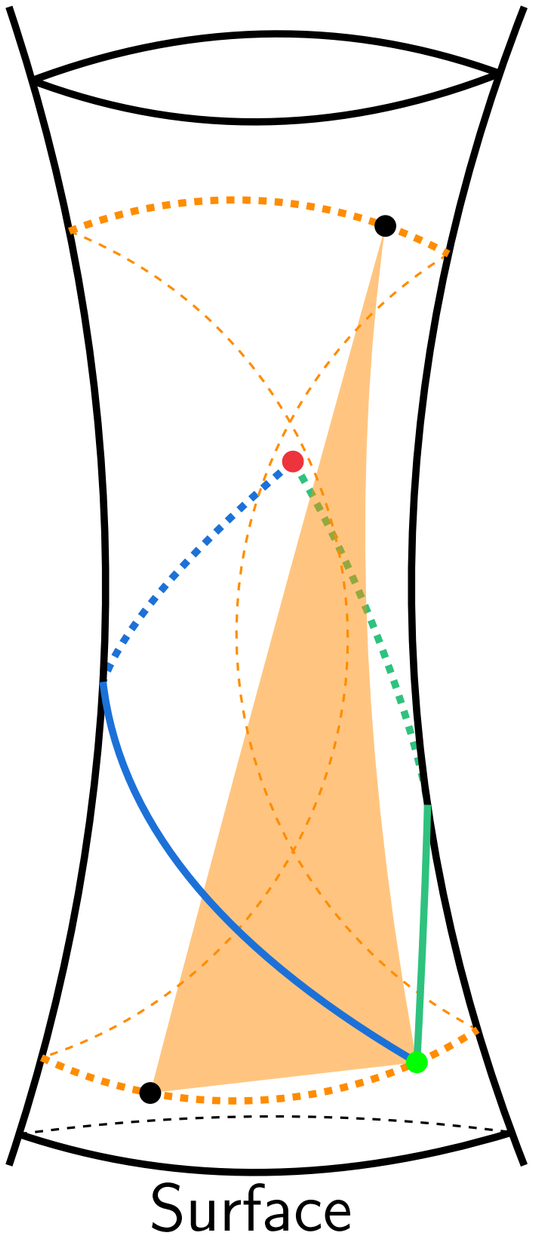


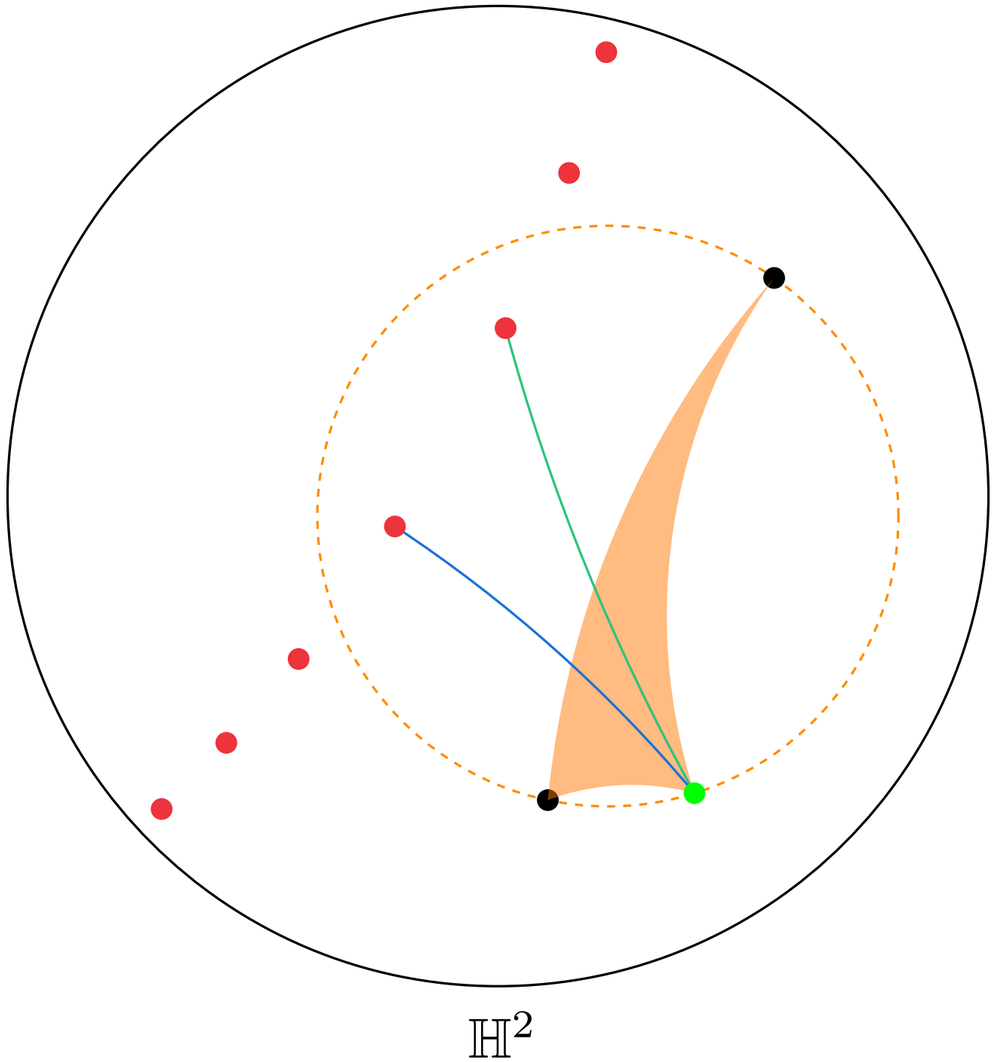
Which way ?





Which way ?

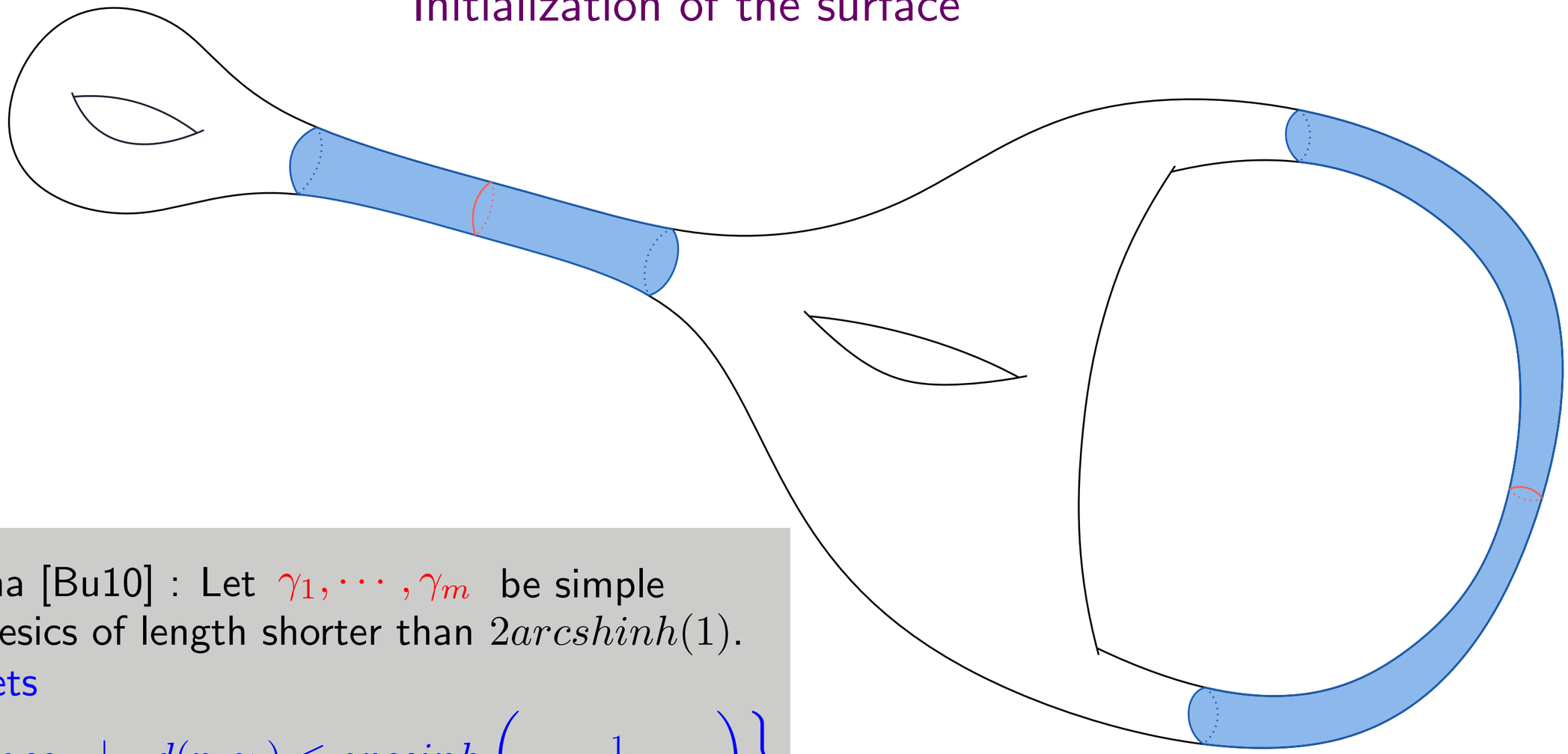




- Delaunay triangulation and Bowyer-Watson algorithm
- Drawing on a hyperbolic surface
- **Bowyer-Watson algorithm on a hyperbolic surface**

Initialization of the surface

Initialization of the surface

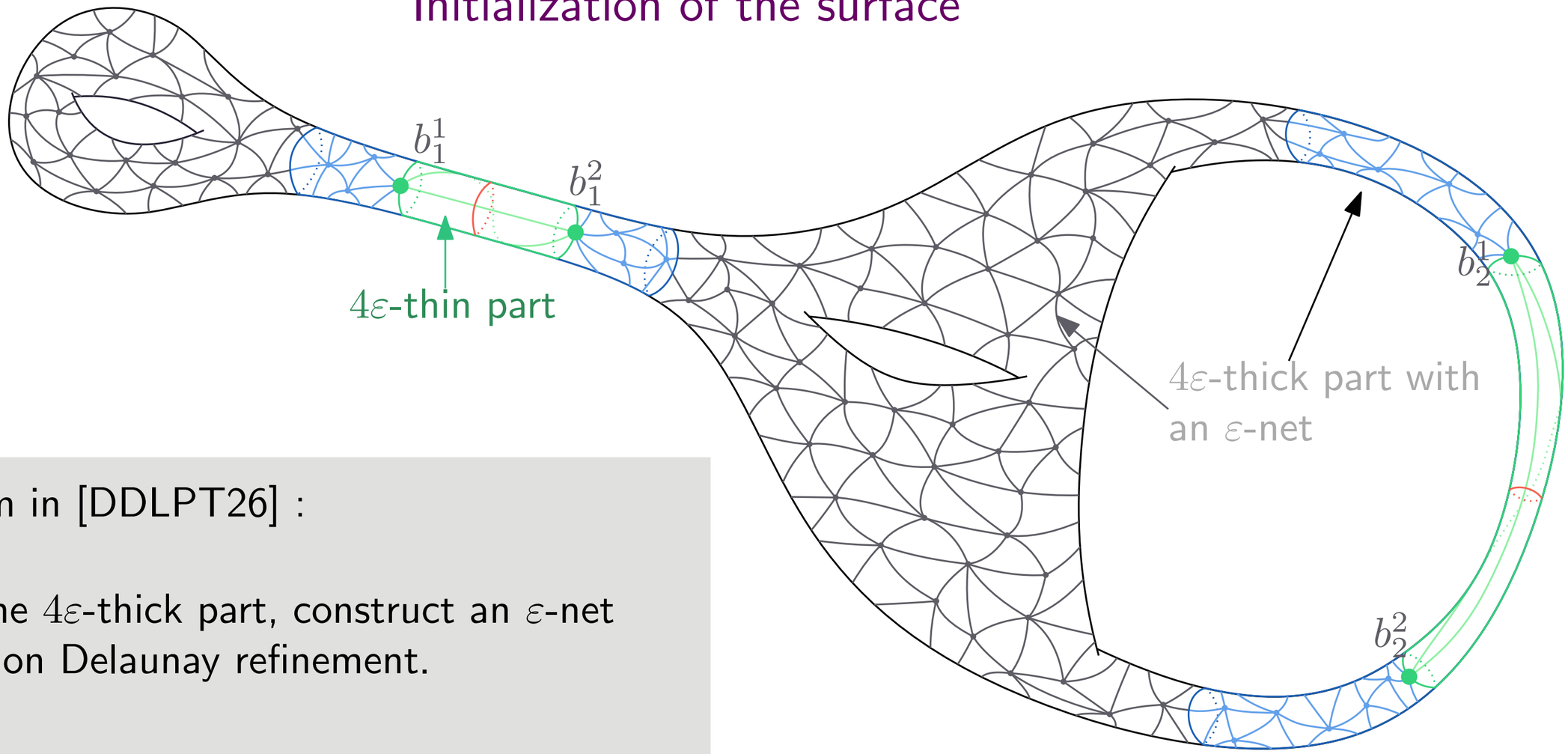


Collar lemma [Bu10] : Let $\gamma_1, \cdots, \gamma_m$ be simple closed geodesics of length shorter than $2arcshinh(1)$. Then the sets

$$\left\{ p \in Surface \quad | \quad d(p, \gamma_i) \leq arcsinh\left(\frac{1}{\sinh\left(\frac{1}{2} l(\gamma_i)\right)} \right) \right\}$$

are pairwise disjoint and homeomorphic to a cylinder.

Initialization of the surface

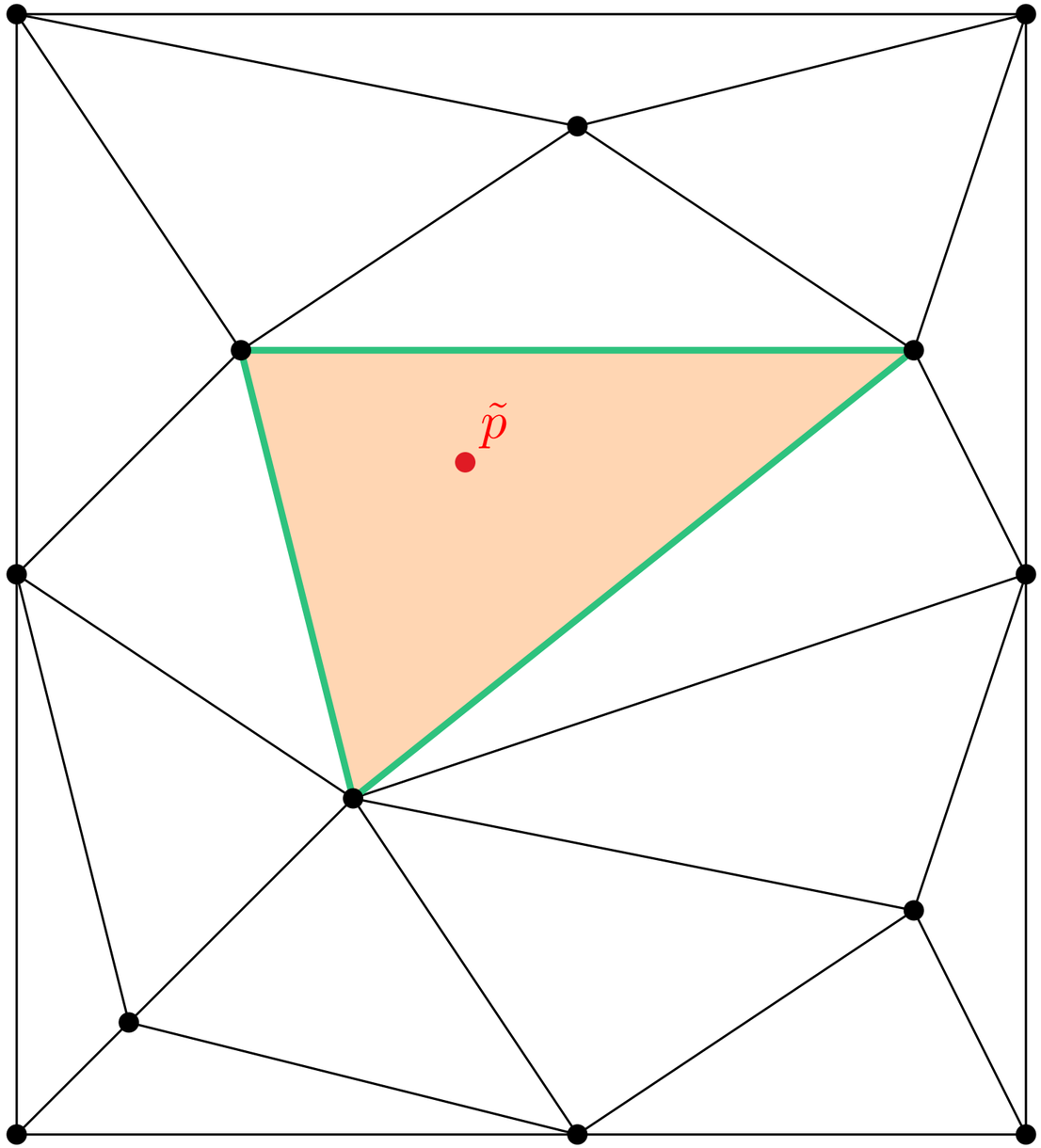


Algorithm in [DDLPT26] :

- In the 4ε-thick part, construct an ε-net based on Delaunay refinement.

- Detect the collars (4ε-thin parts) and place points $b_{1,2}$ on their borders.

Insertion in 4ε-thick part

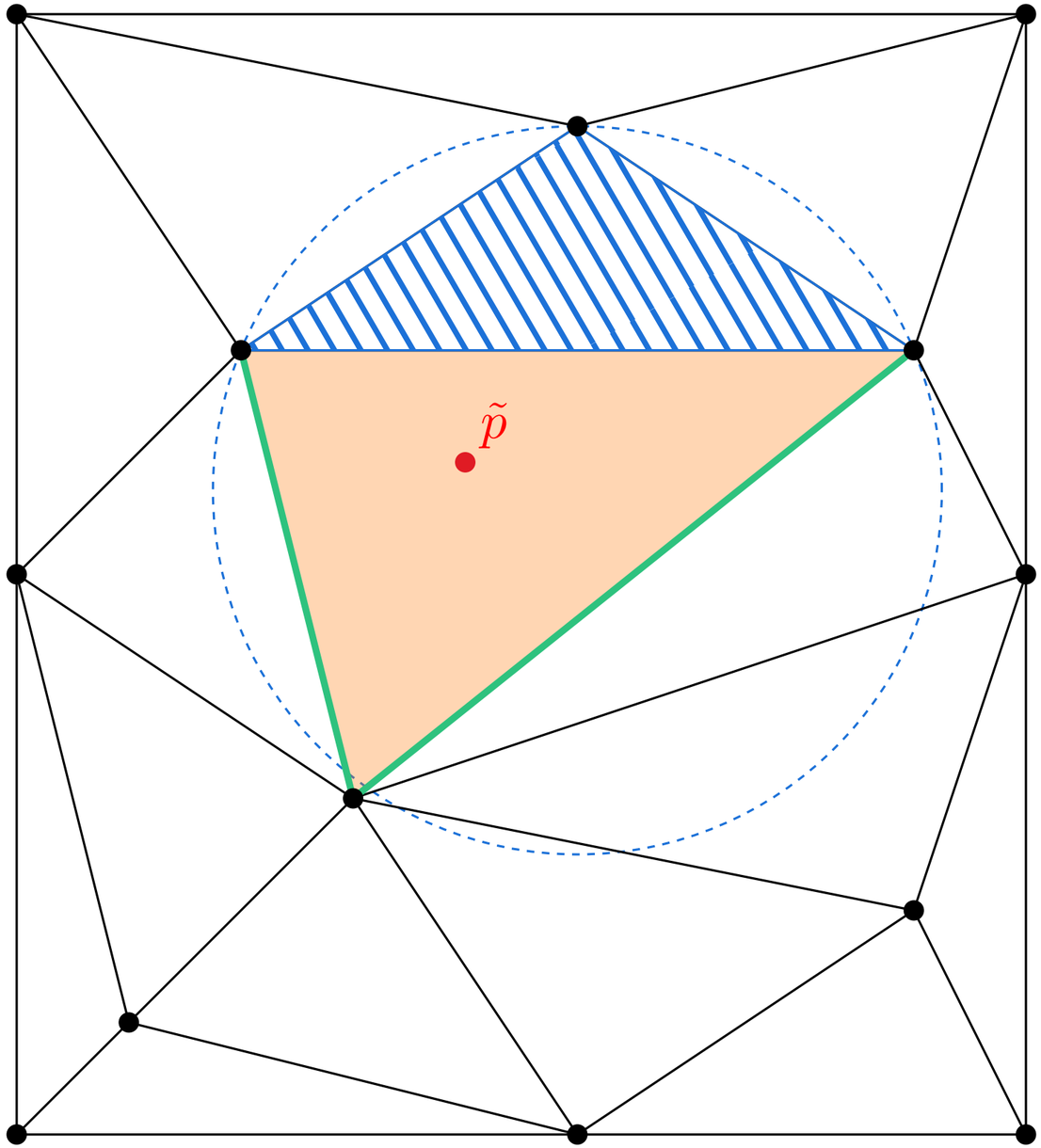


Processing in the universal cover

→ Find a triangle which contains $\widetilde{p}$.

→ Start a DFS to find conflict zone.

Insertion in 4ε-thick part

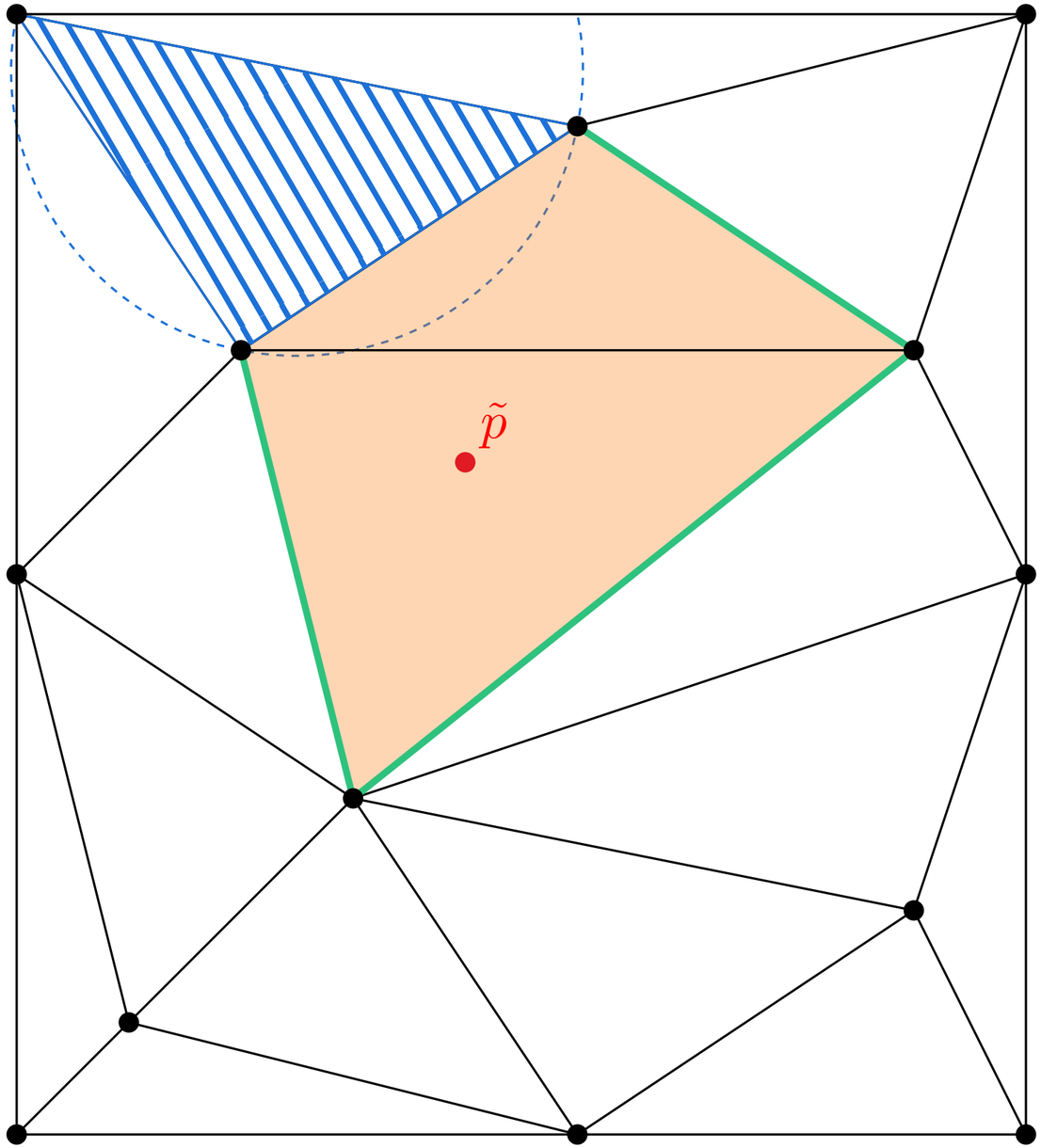


Processing in the universal cover

⟶ Conflict with $\widetilde{p}$.

⟶ Push the two other sides in the stack.

Insertion in 4ε-thick part

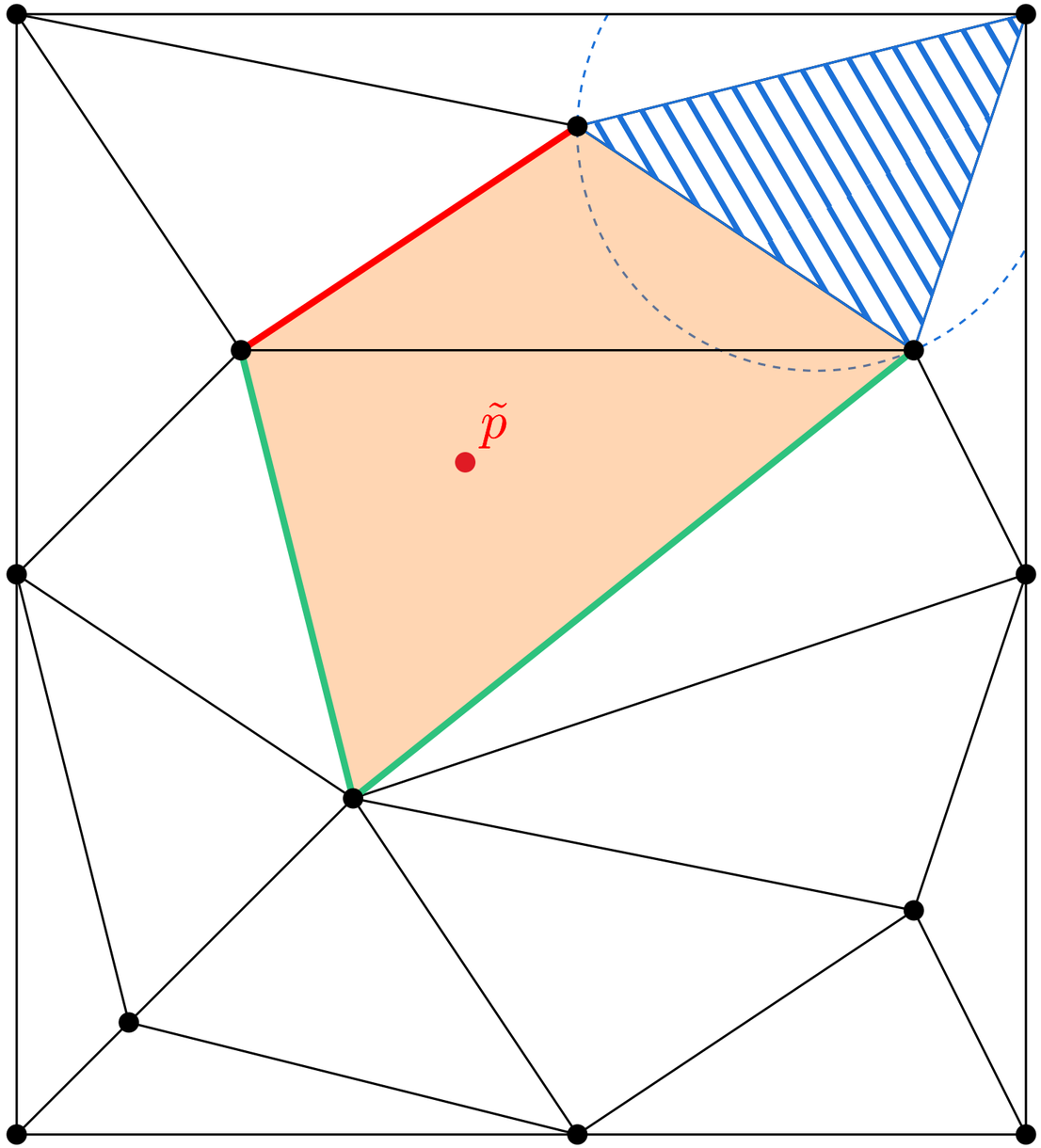


Processing in the universal cover

⟶ Not in conflict with $\widetilde{p}$.

⟶ The side in commun with the conflict zone belongs to the border.

Insertion in 4ε-thick part

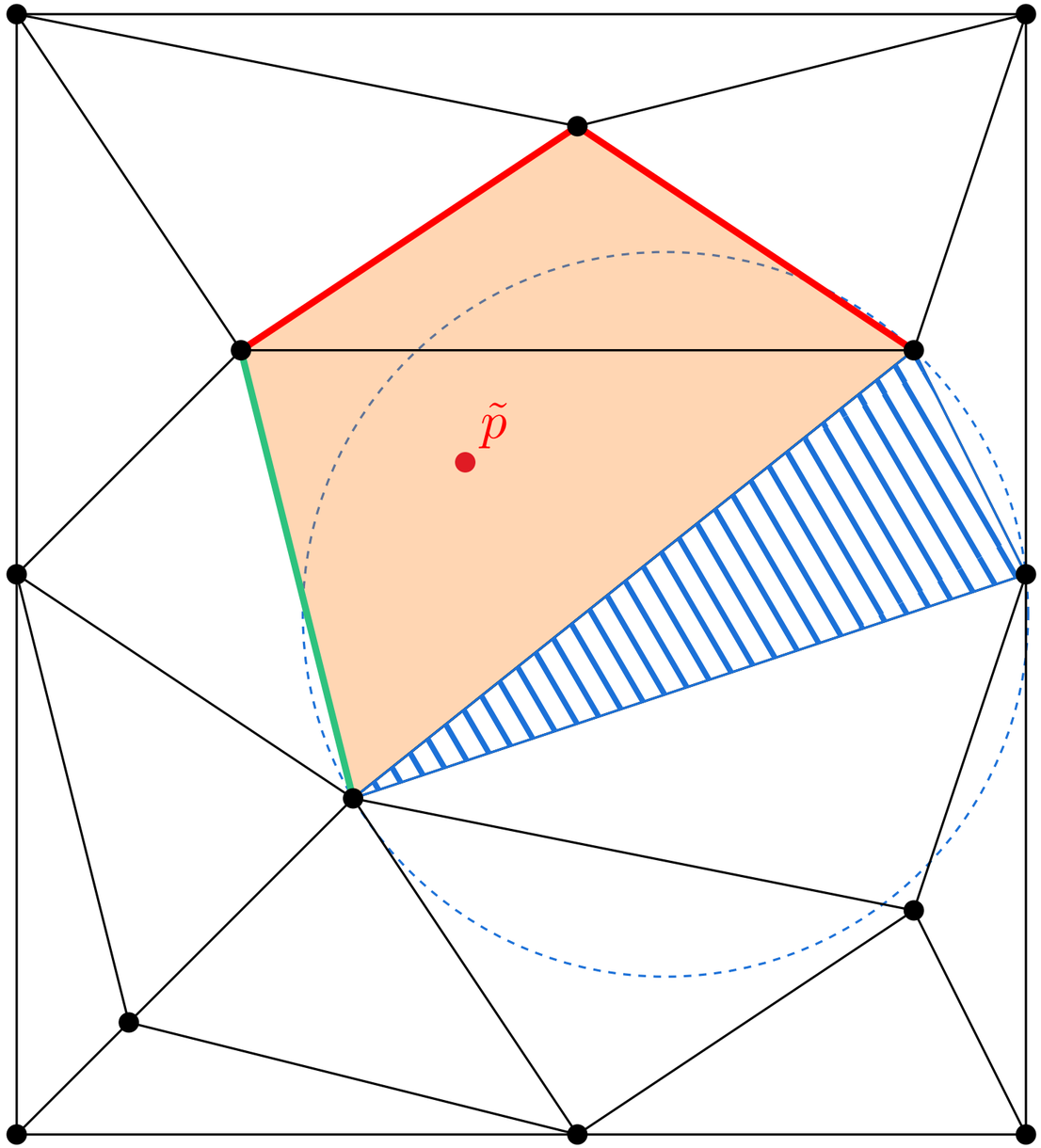


Processing in the universal cover

⟶ Not in conflict with $\widetilde{p}$.

⟶ The side in commun with the conflict zone belongs to the border.

Insertion in 4ε-thick part

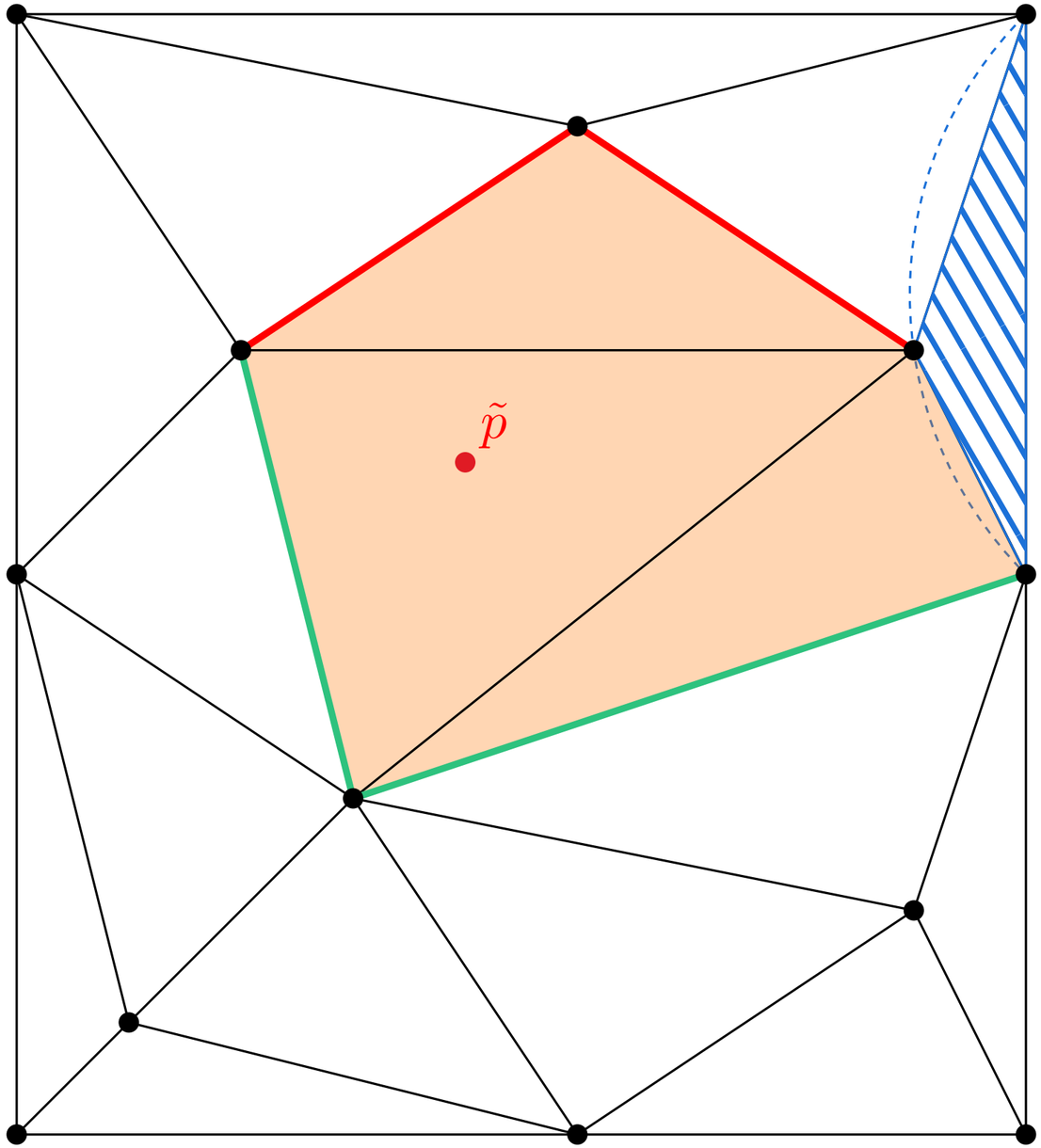


Processing in the universal cover

$\longrightarrow$ Conflict with $\widetilde{p}$.

$\longrightarrow$ Push the two other sides in the stack.

Insertion in 4ε-thick part

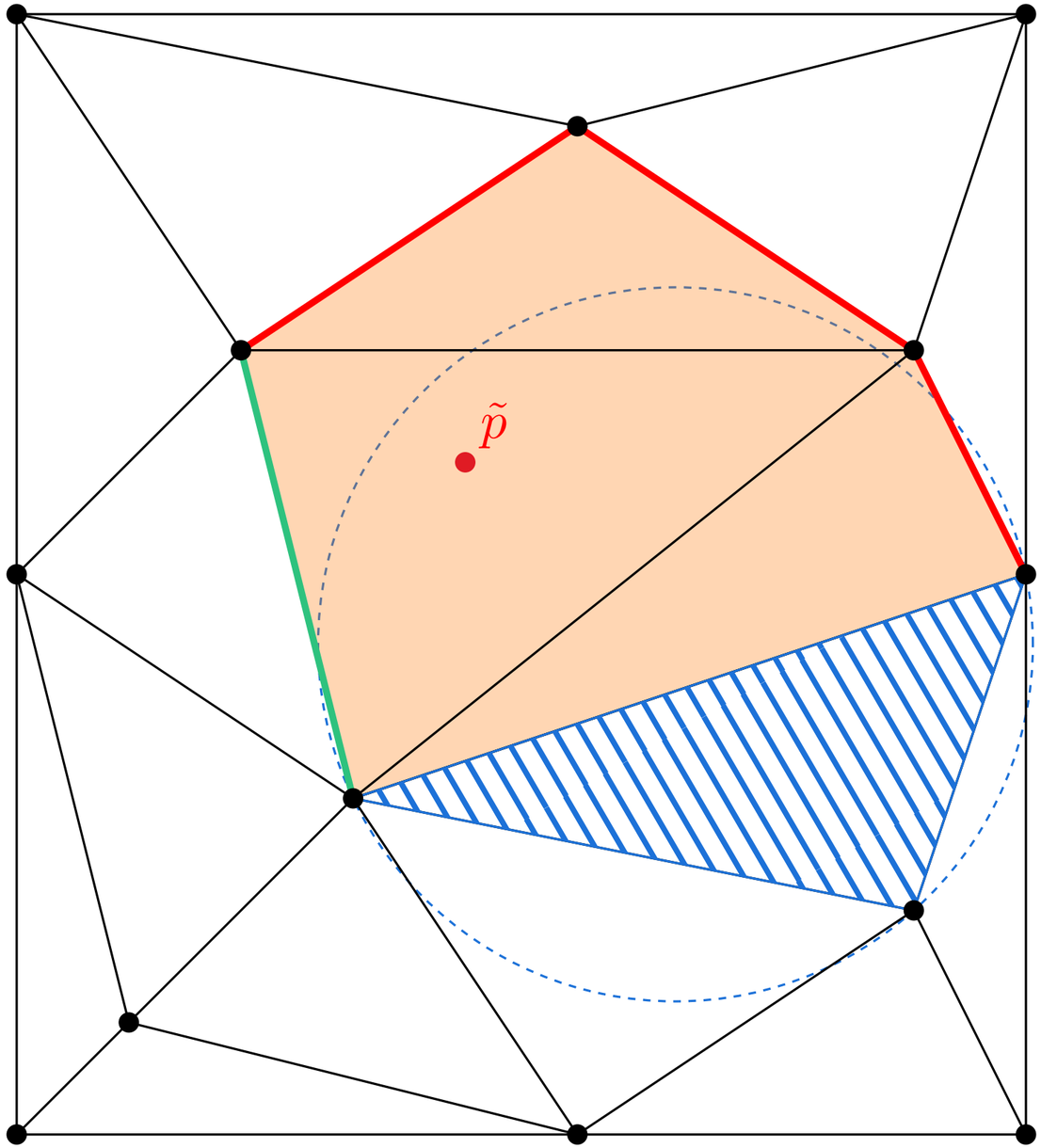


Processing in the universal cover

⟶ Not in conflict with $\widetilde{p}$.

⟶ The side in commun with the conflict zone belongs to the border.

Insertion in 4ε-thick part

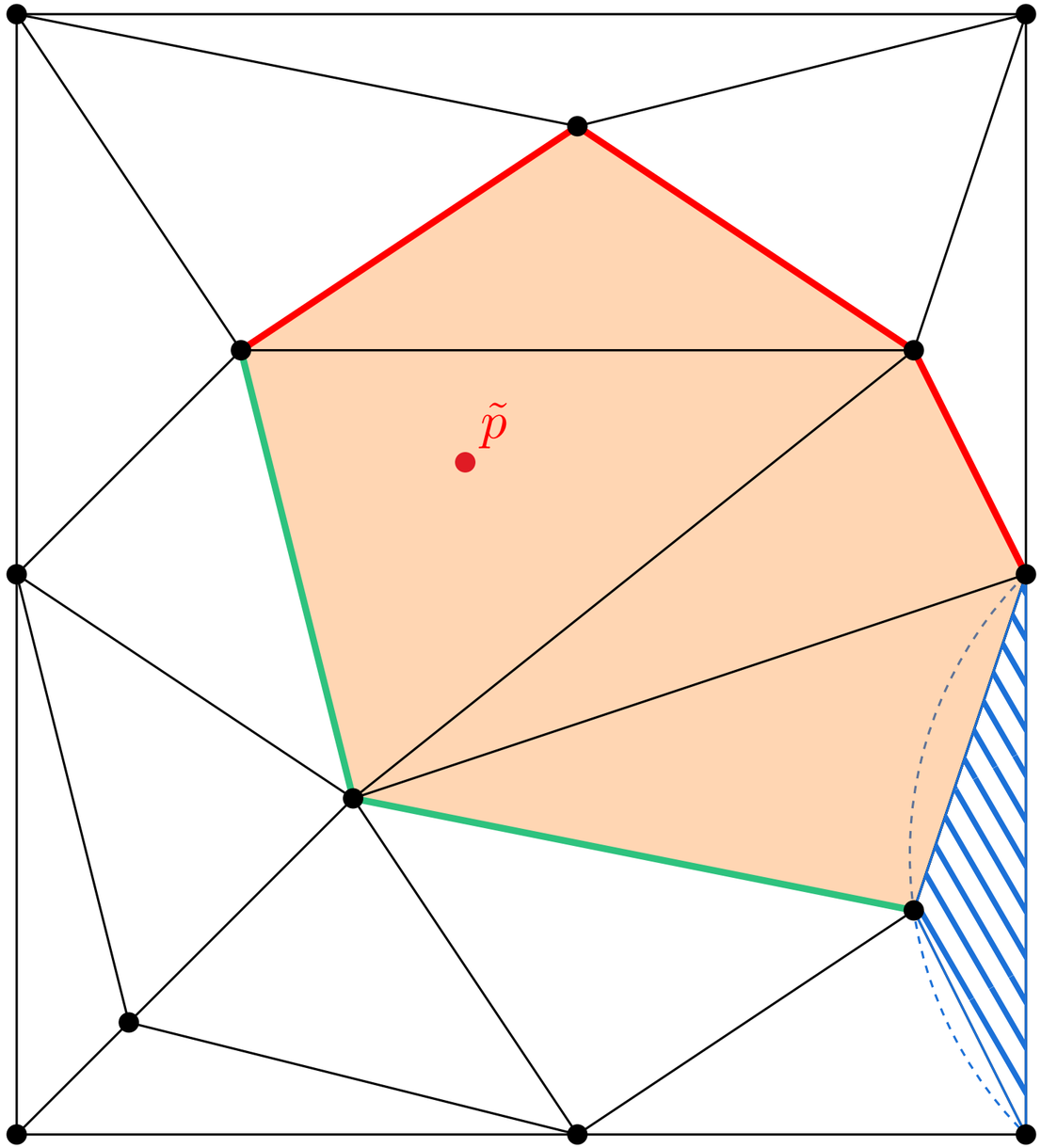


Processing in the universal cover

⟶ Conflict with $\widetilde{p}$.

⟶ Push the two other sides in the stack.

Insertion in 4ε-thick part

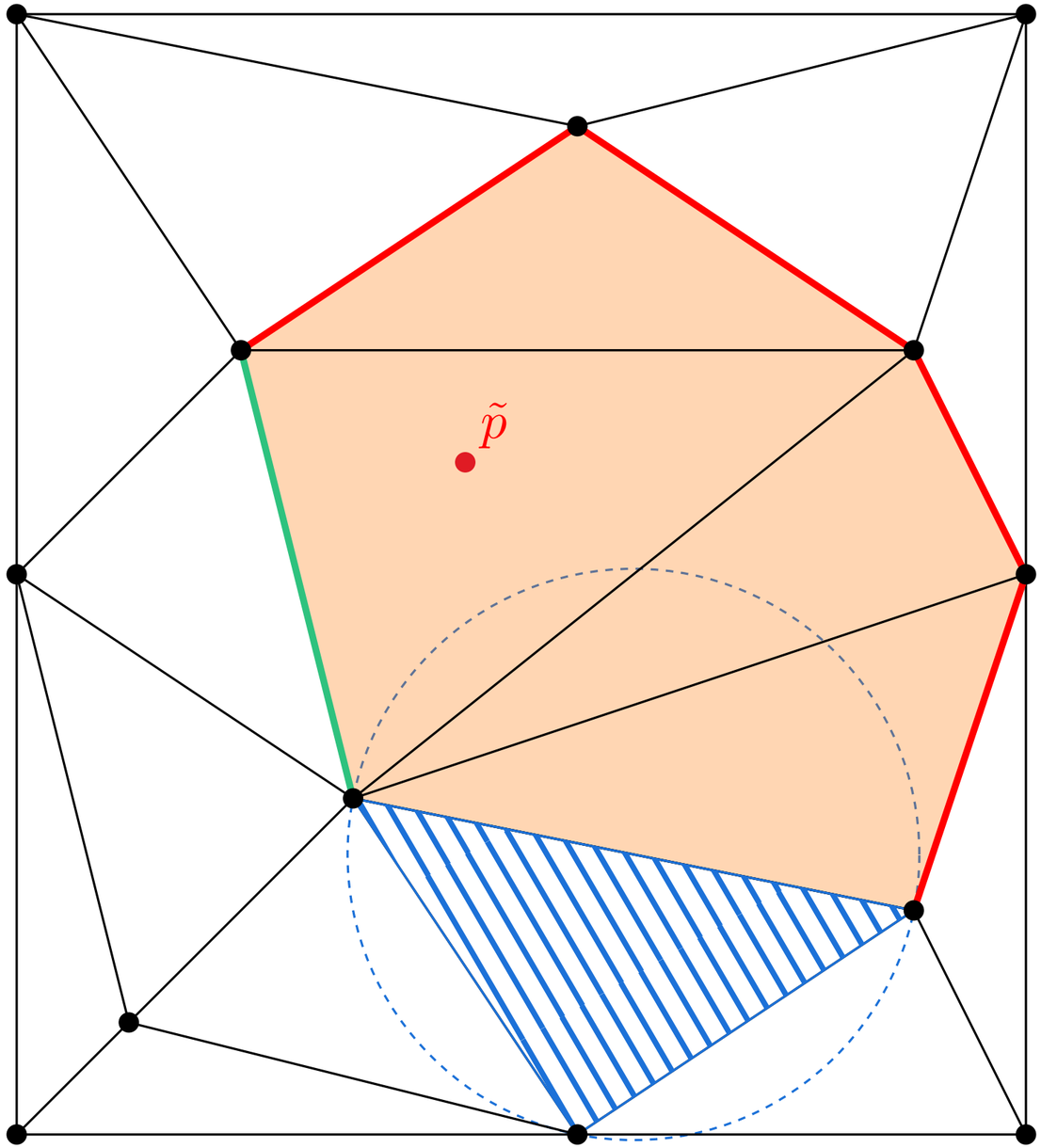


Processing in the universal cover

⟶ Not in conflict with $\widetilde{p}$.

⟶ The side in commun with the conflict zone belongs to the border.

Insertion in 4ε-thick part

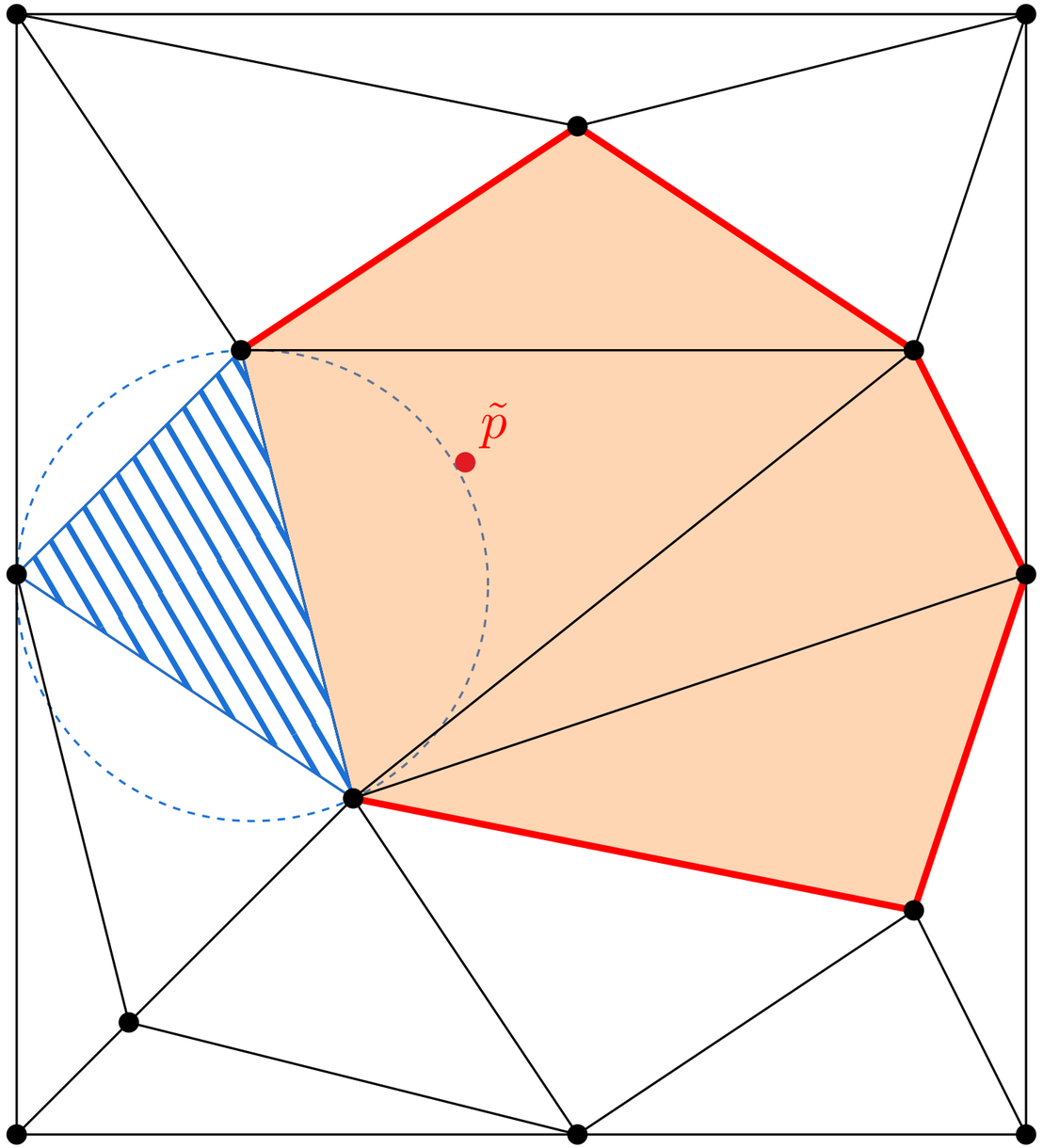


Processing in the universal cover

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in 4ε-thick part

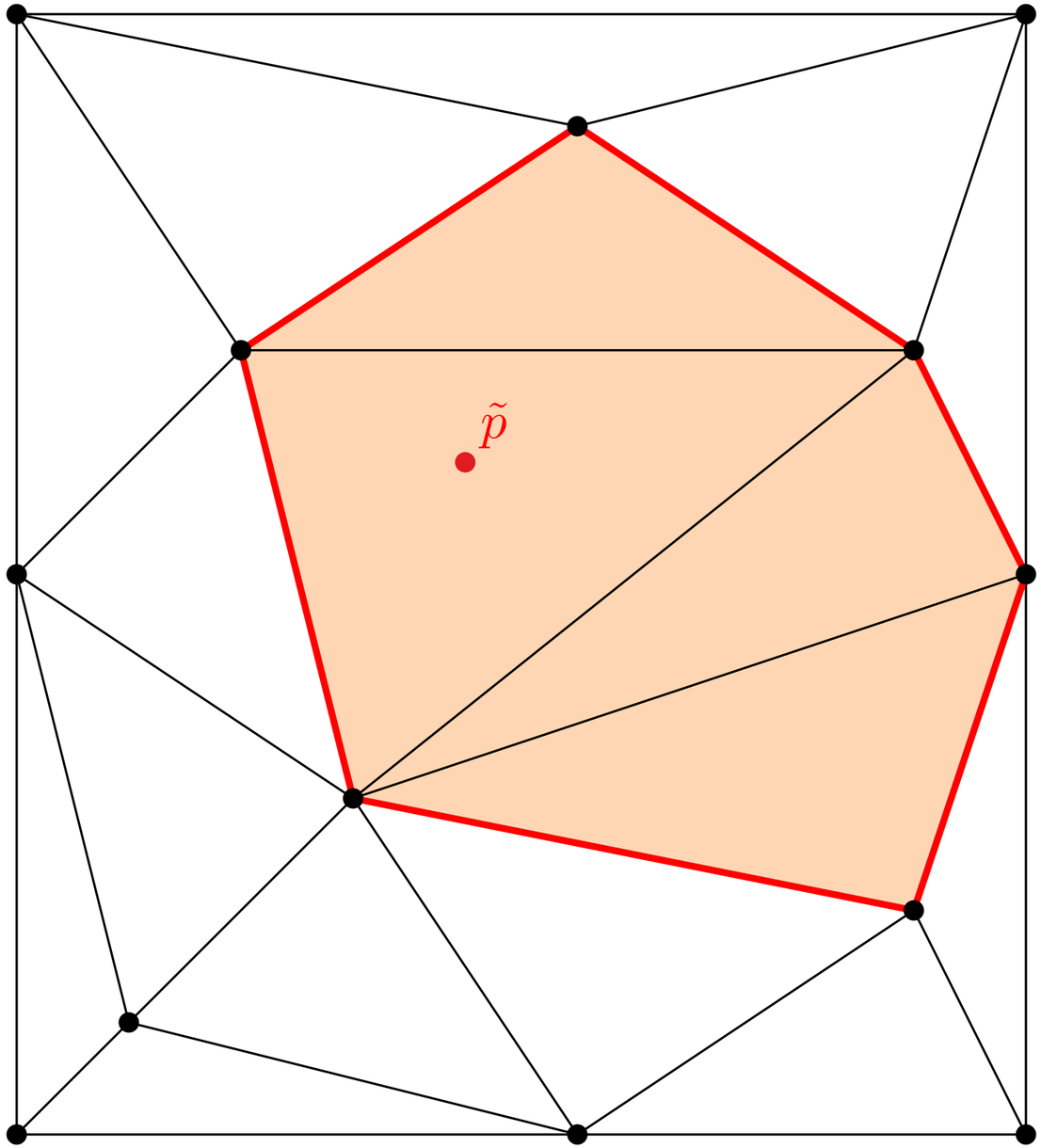


Processing in the universal cover

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

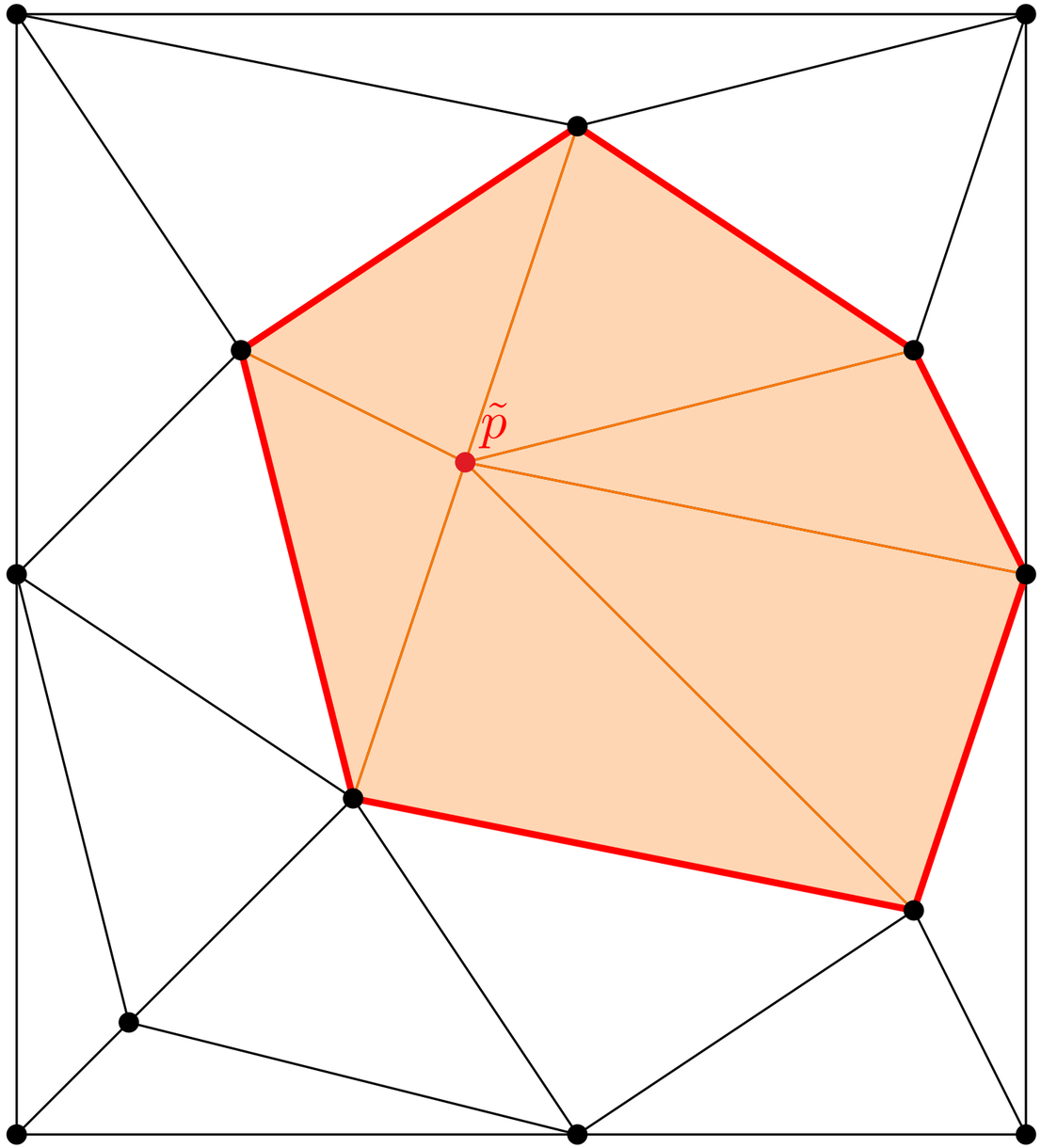
Insertion in 4ε-thick part



Processing in the universal cover

$\longrightarrow$ End of the BFS, we find the conflict zone.

Insertion in 4ε-thick part

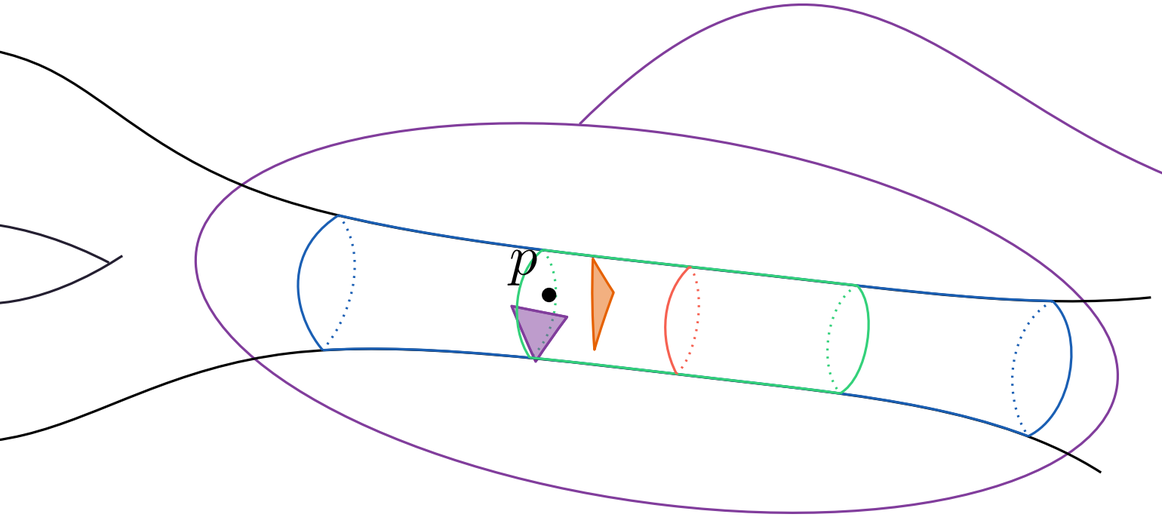


Processing in the universal cover

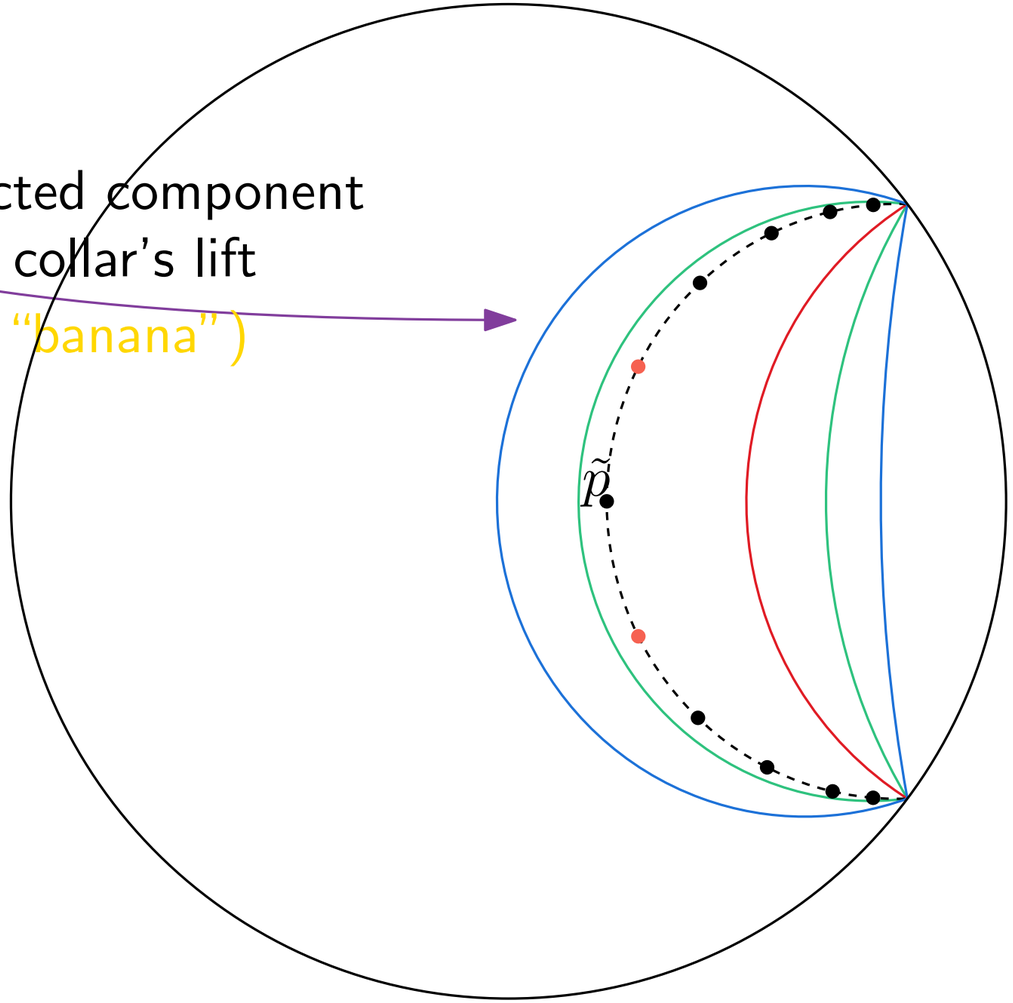
→ Remove all triangles in the conflict zone.

→ Connect $\widetilde{p}$ to the border's vertices.

4ε-thin part : lift of a collar

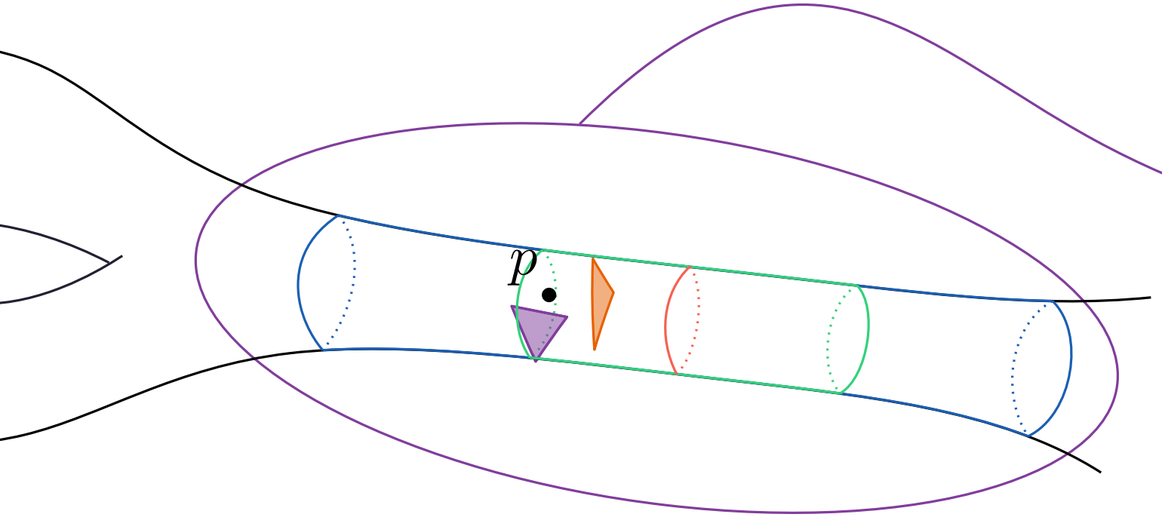


A connected component
of a collar's lift
(a "banana")

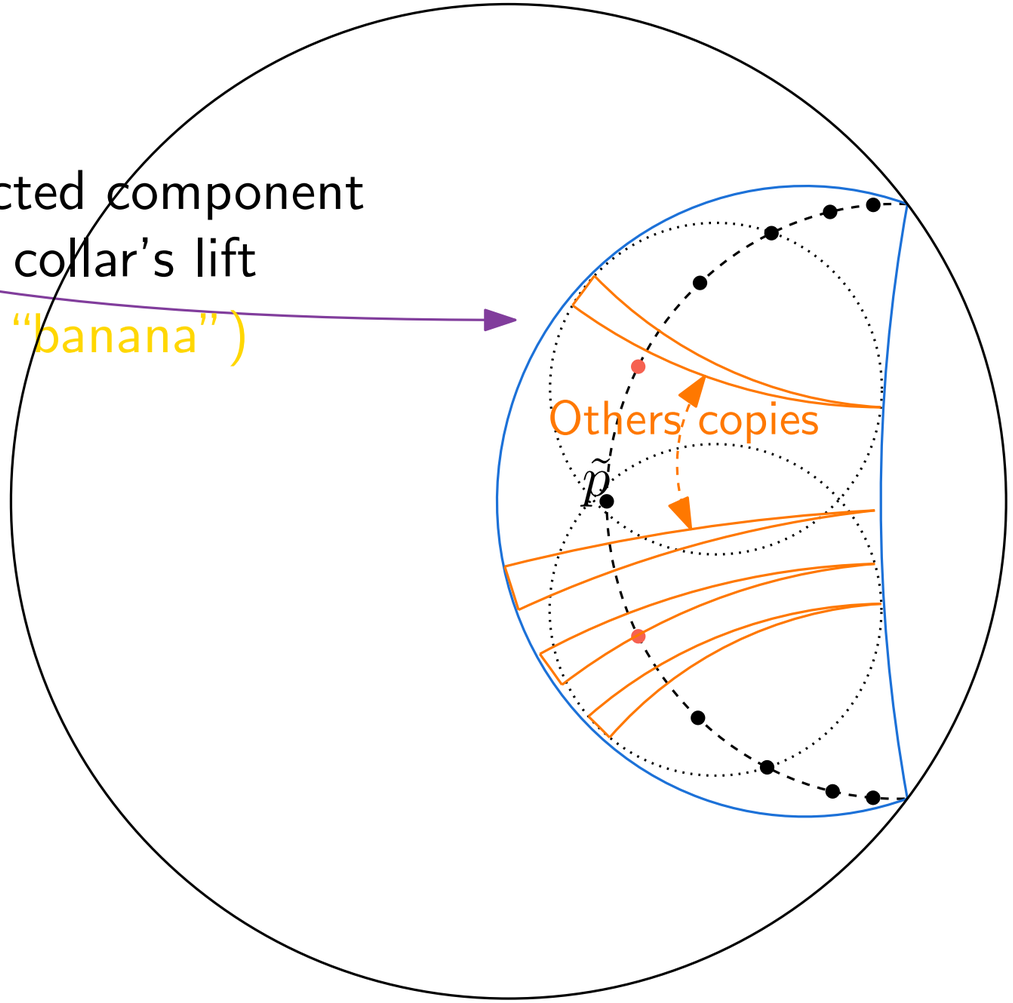


$\mathbb{H}^2$, universal cover

4ε-thin part : lift of a collar



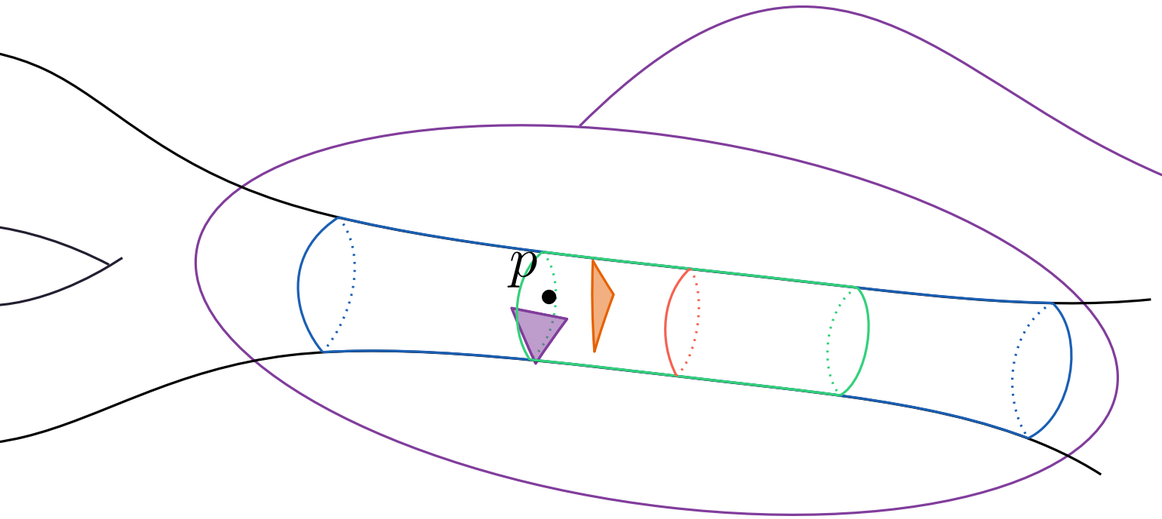
A connected component
of a collar's lift
(a "banana")



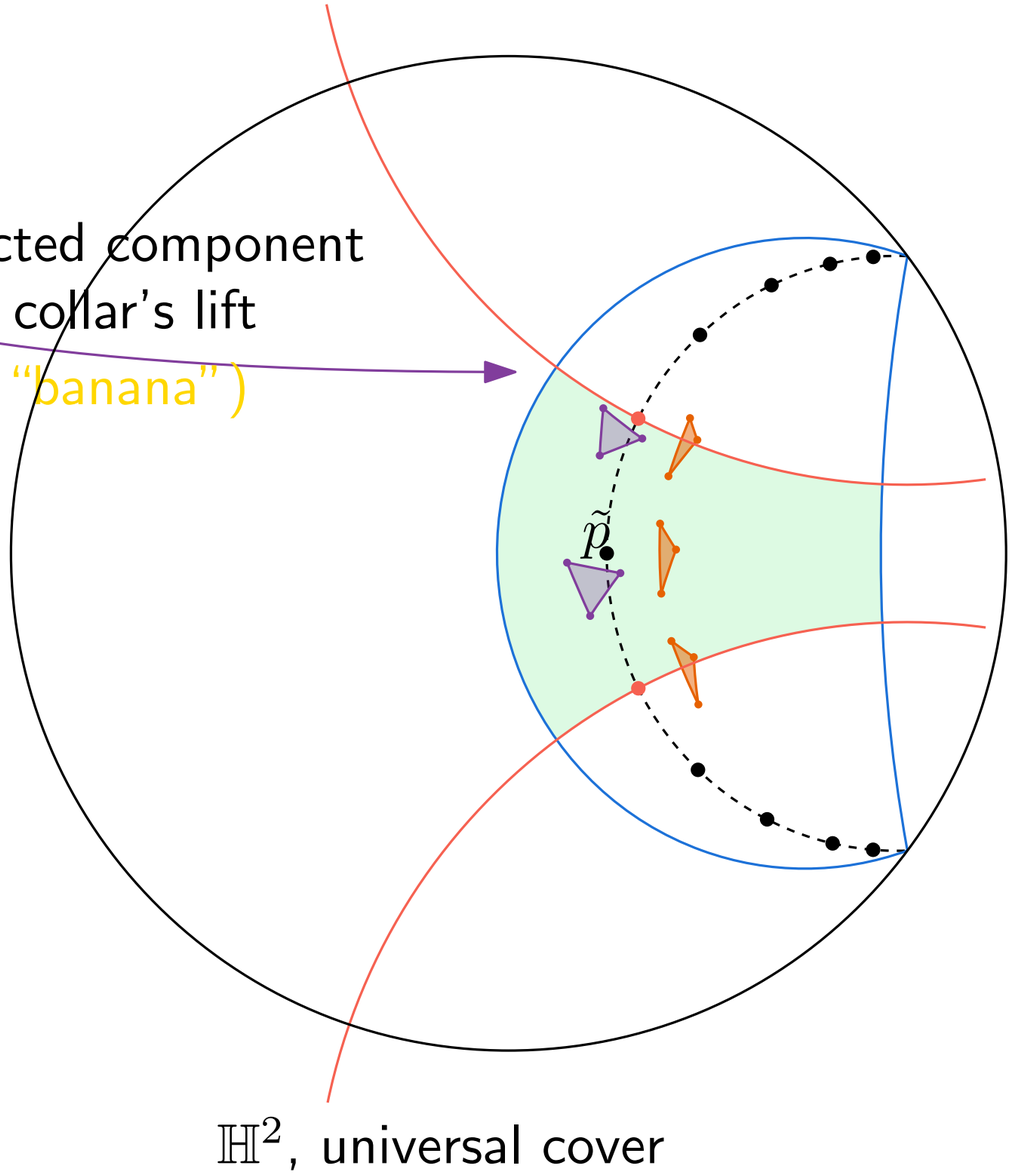
Prop : If p is in the 4ε-thin part,
then the conflict zone is included in the collar.

$\mathbb{H}^2$, universal cover

4ε-thin part : lift of a collar



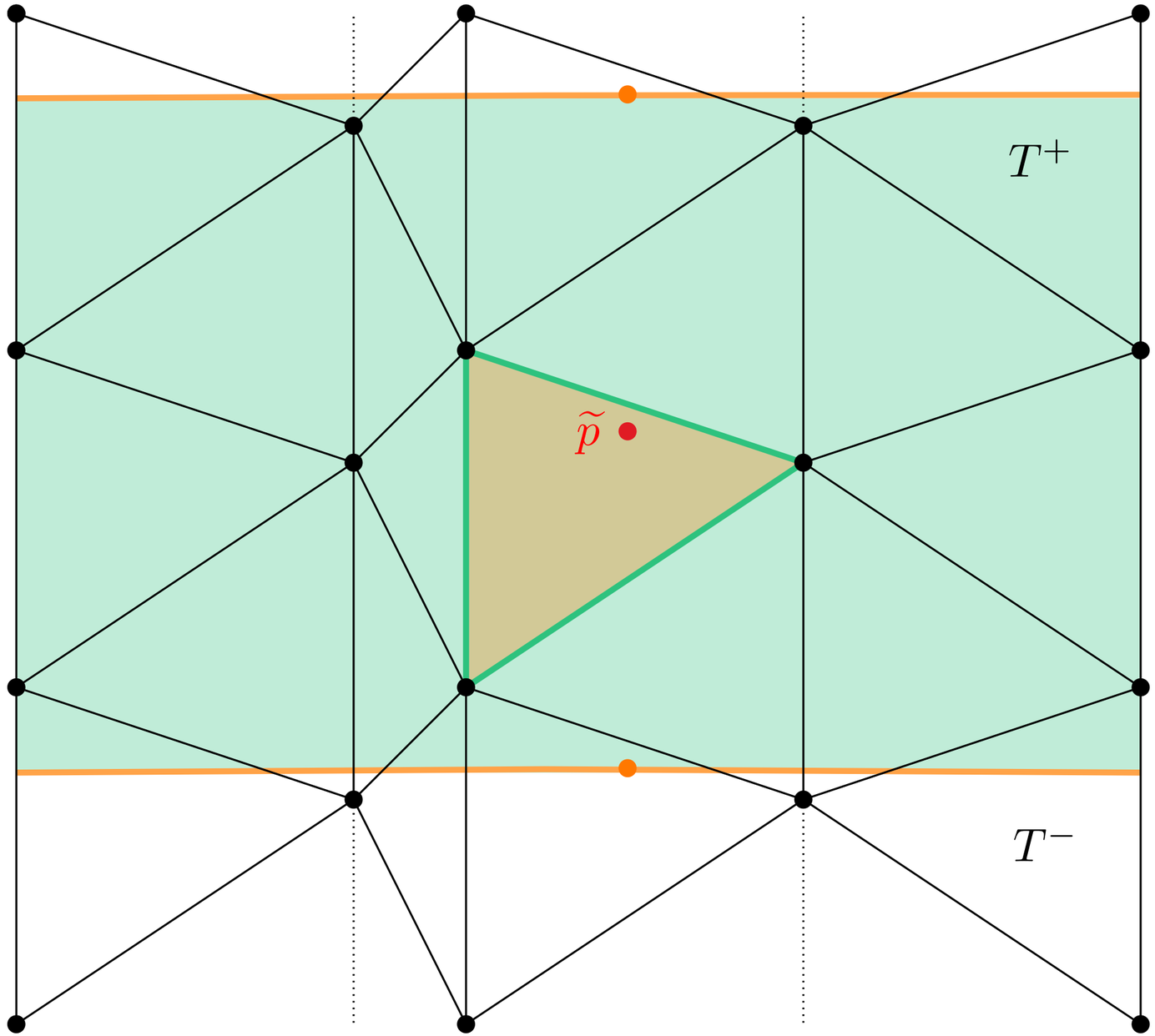
A connected component
of a collar's lift
(a "banana")



Prop : If p is in the 4ε-thin part,
then the conflict zone is included in the collar.

Restrict the conflict zone to triangles
having a vertex in the green region.

Insertion in the 4ε-thin part

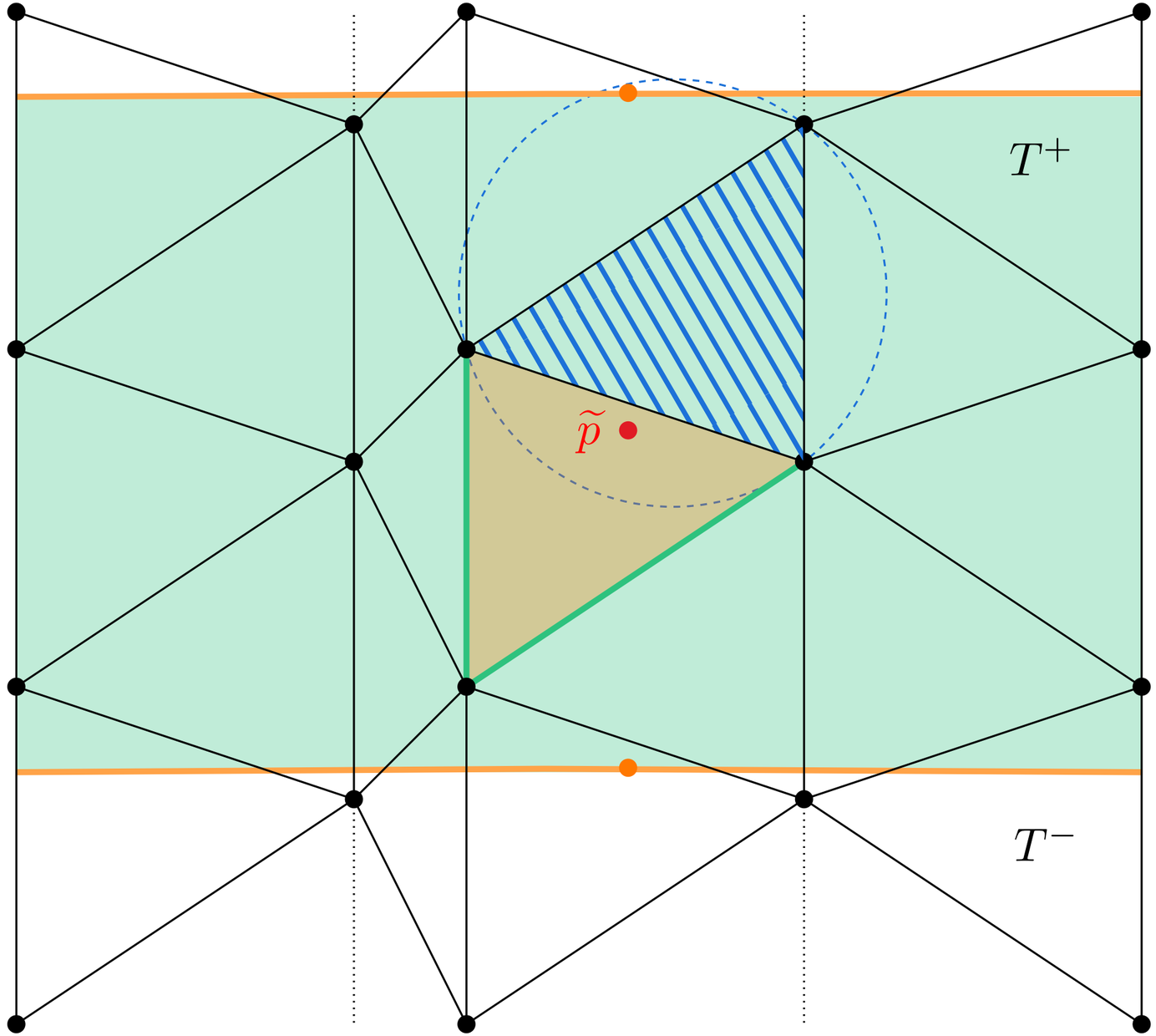


Processing in a banana

→ Find a triangle which contains $\widetilde{p}$.

→ Start a DFS to find conflict zone.

Insertion in the 4ε-thin part

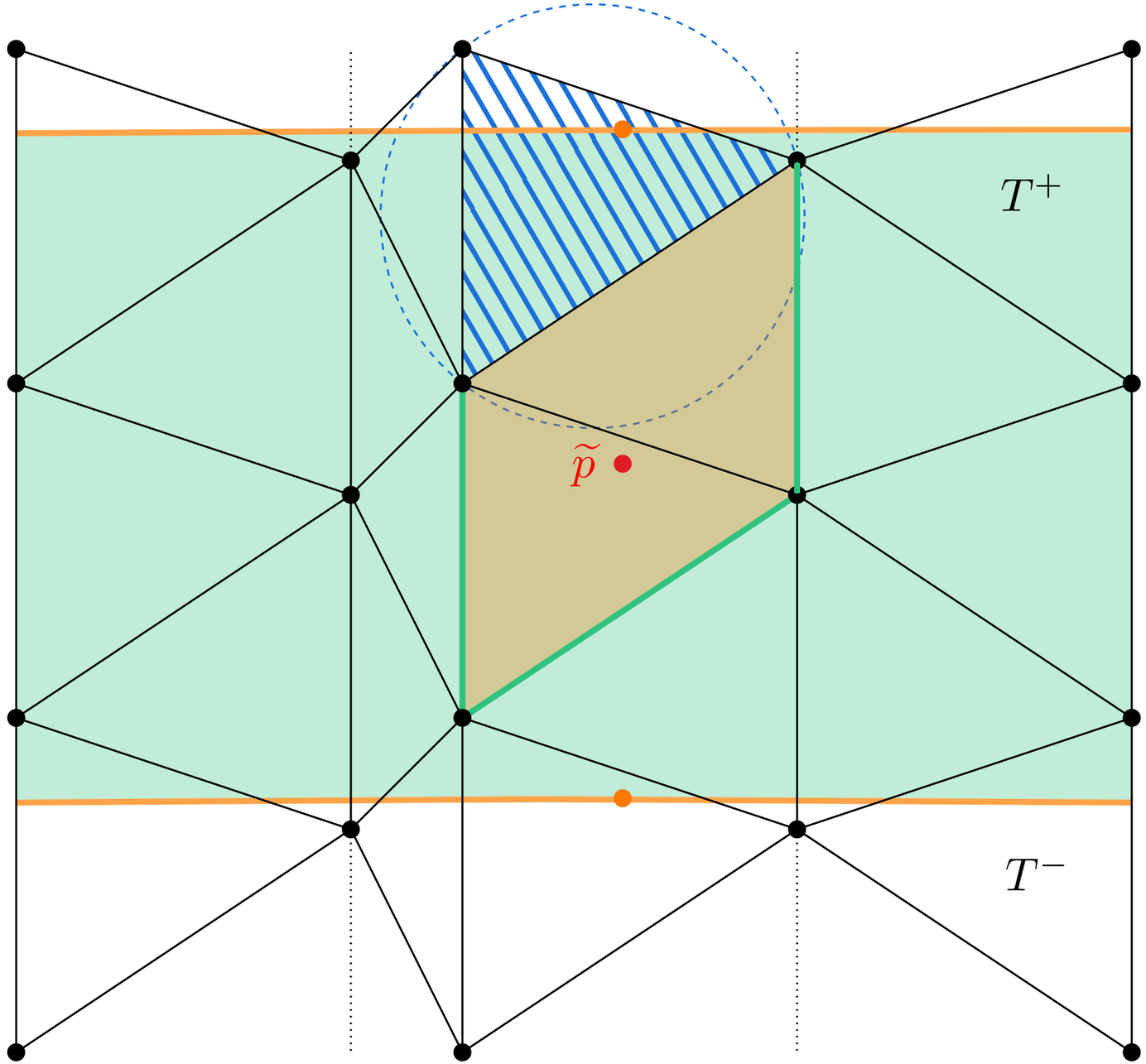


Processing in a banana

→ Conflict with $\widetilde{p}$ **and** the vertices are between the two lines.

→ Push the two other sides in the stack.

Insertion in the 4ε-thin part

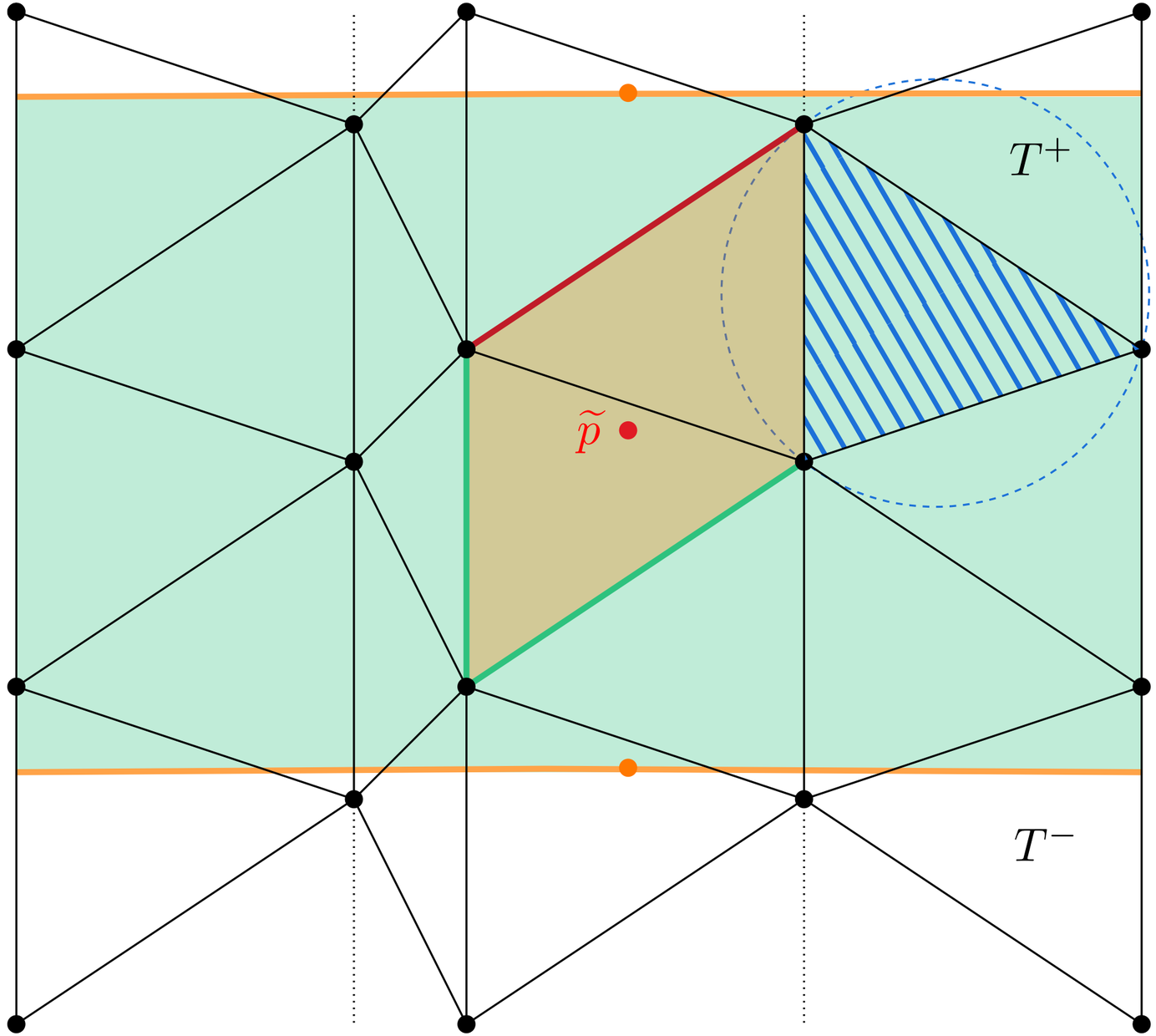


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

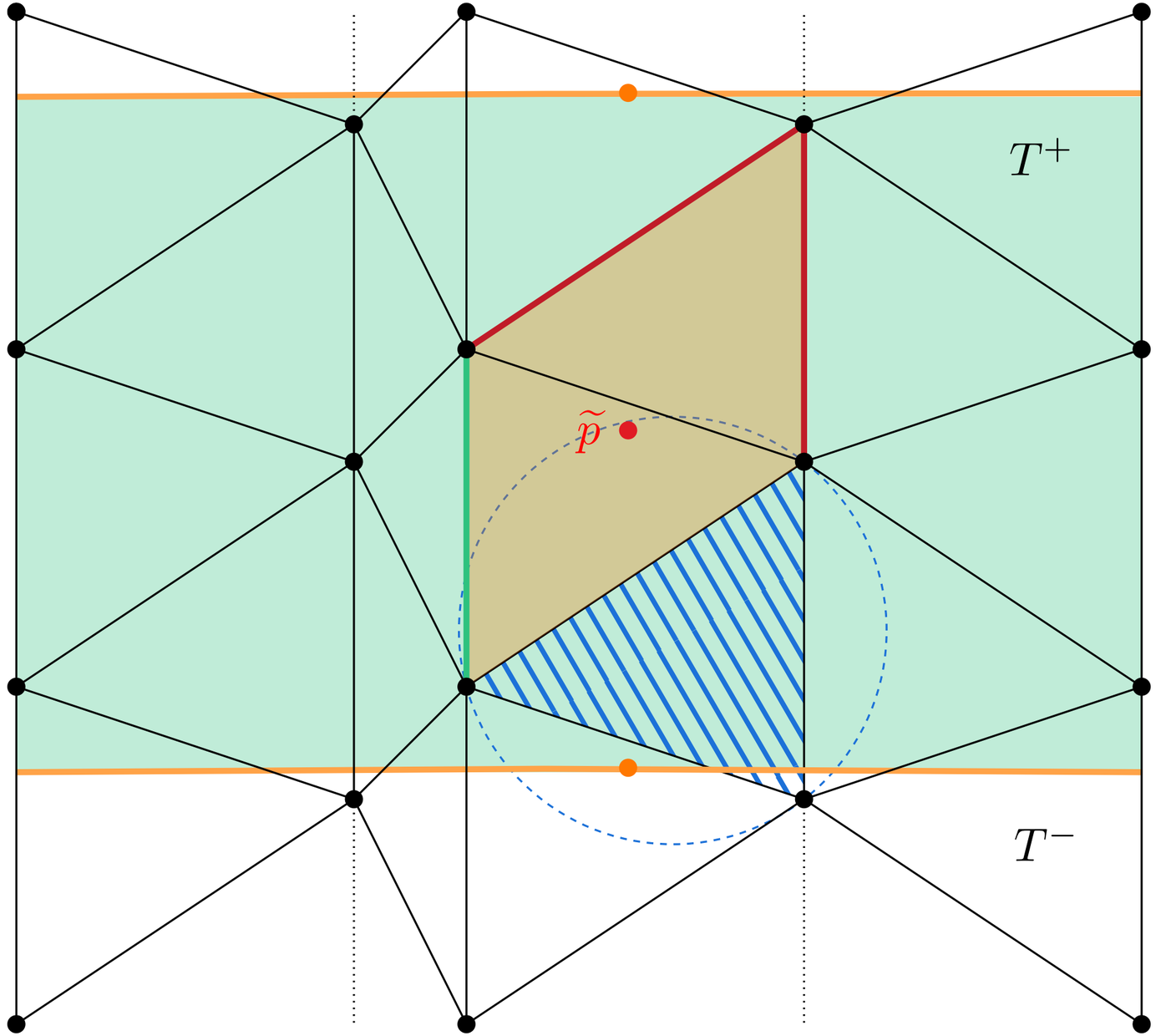


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

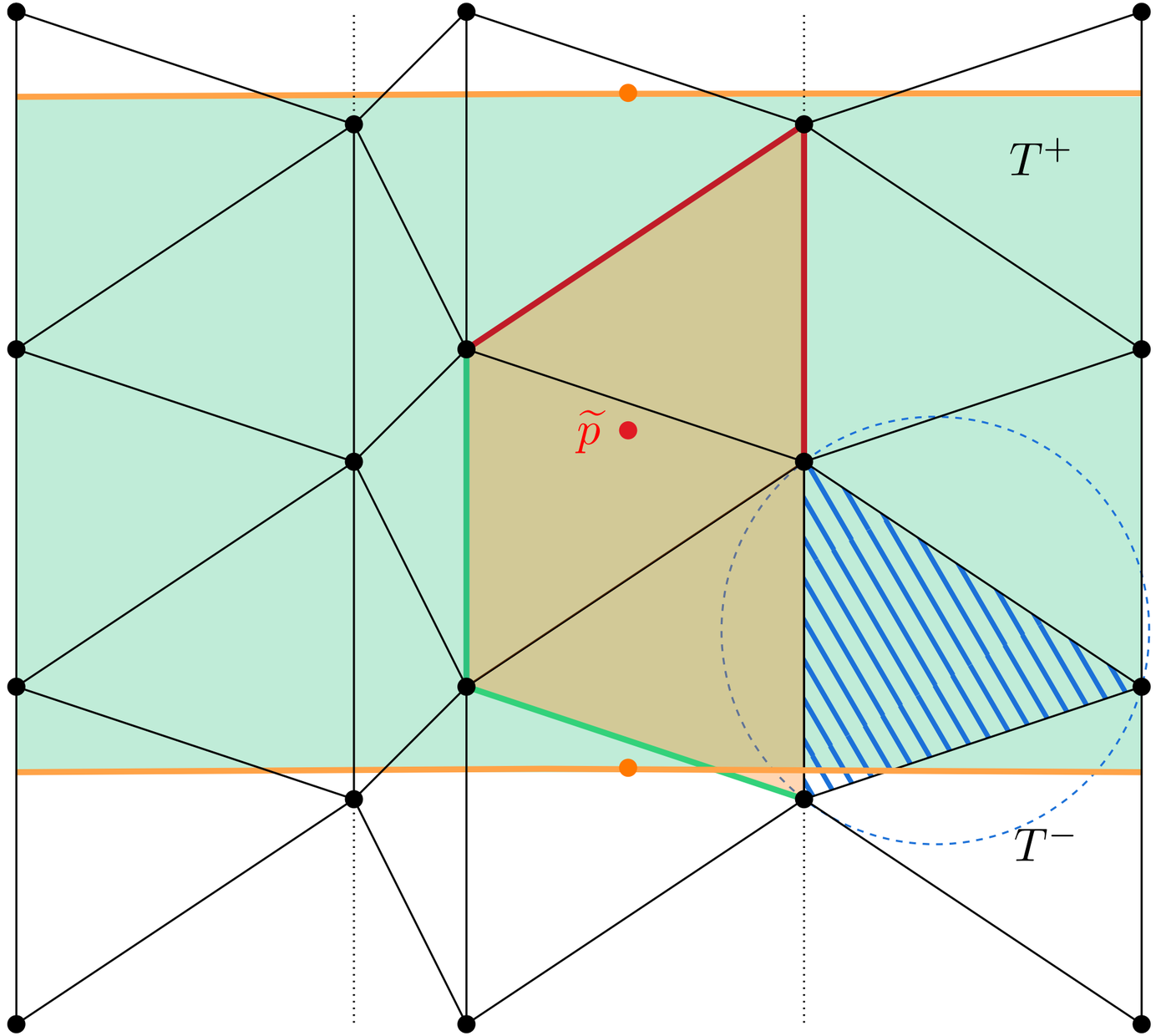


Processing in a banana

→ Conflict with $\widetilde{p}$ **and** has a vertex between the two lines.

→ Push the two other sides in the stack.

Insertion in the 4ε-thin part

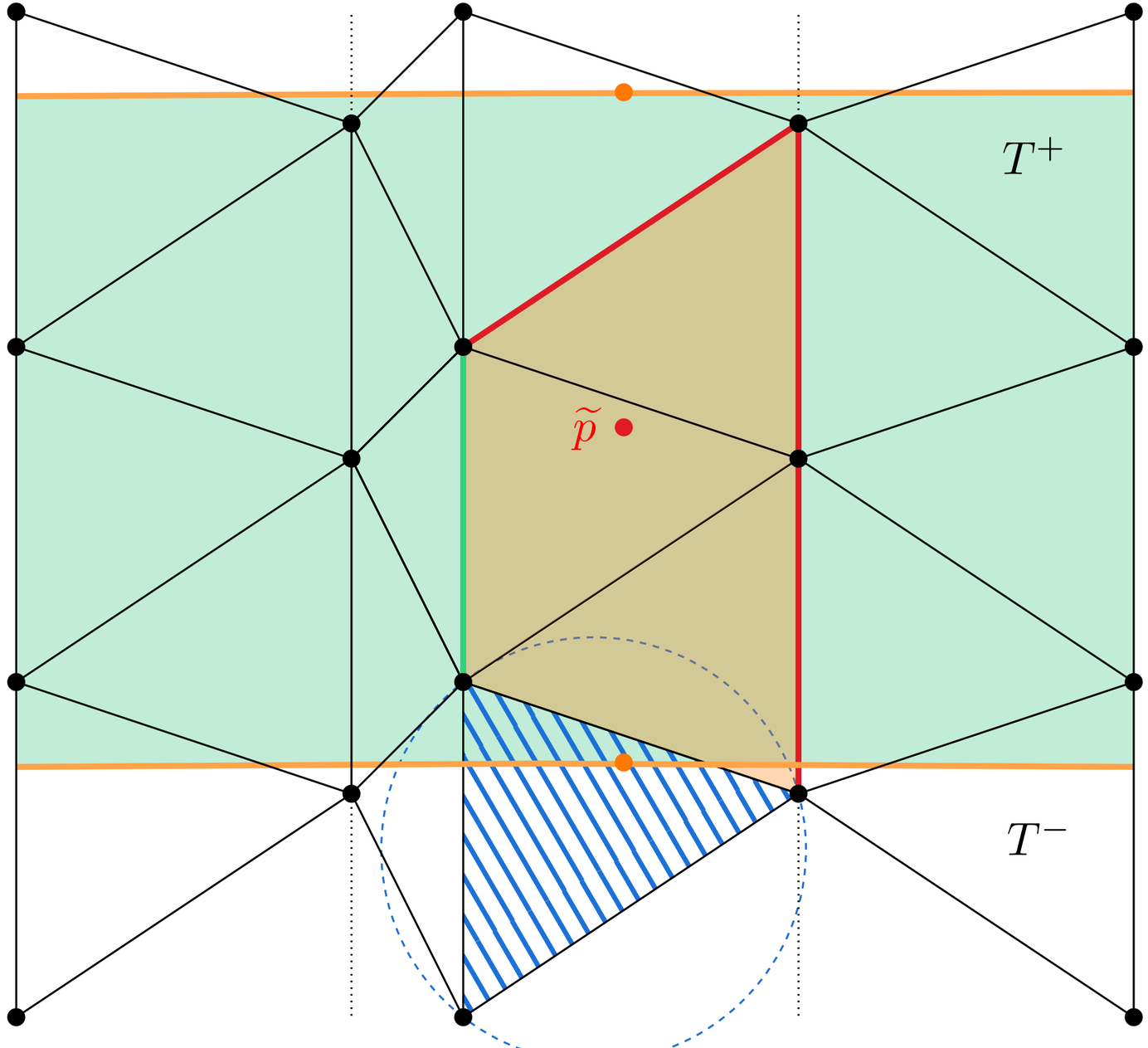


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

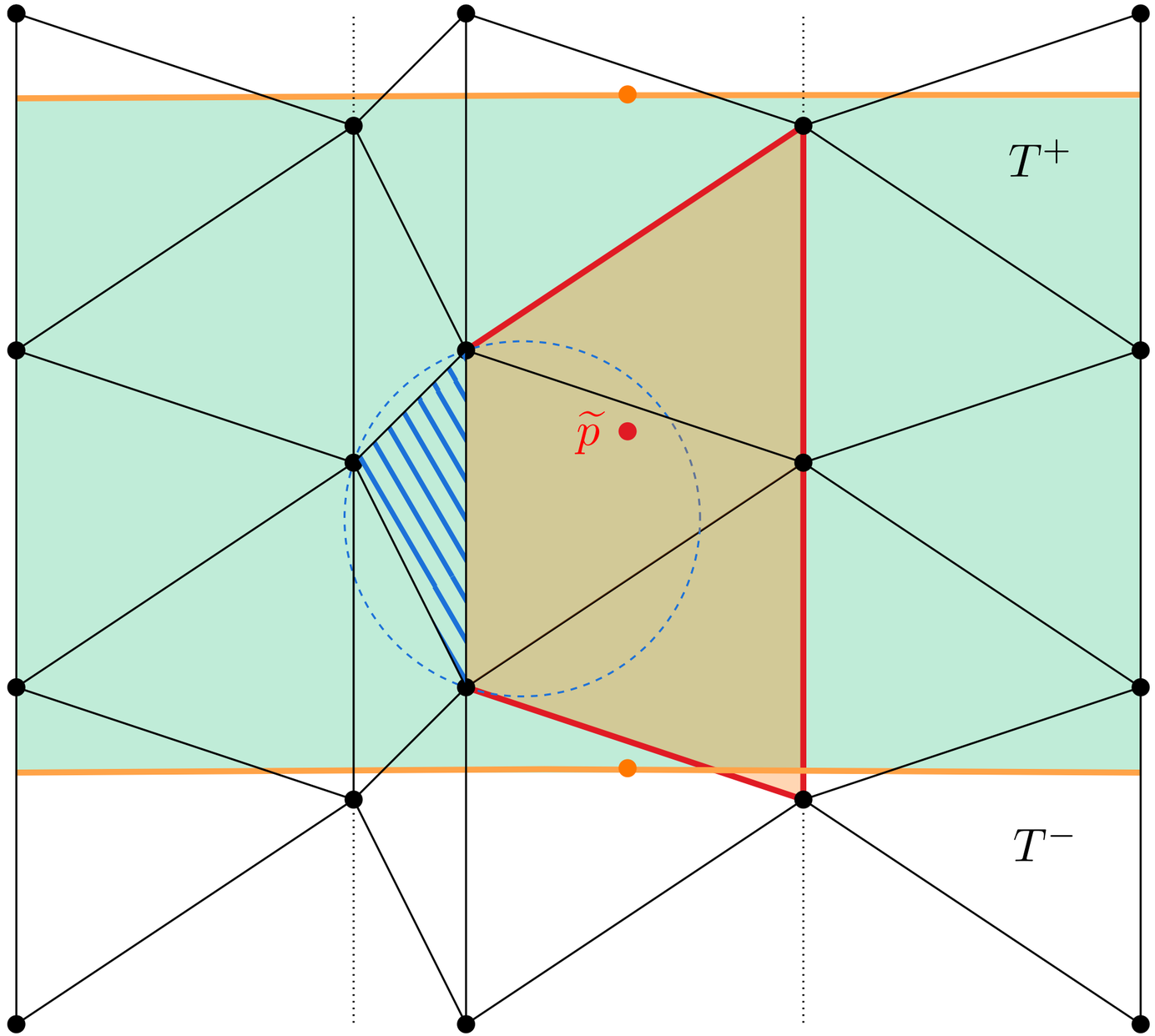


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

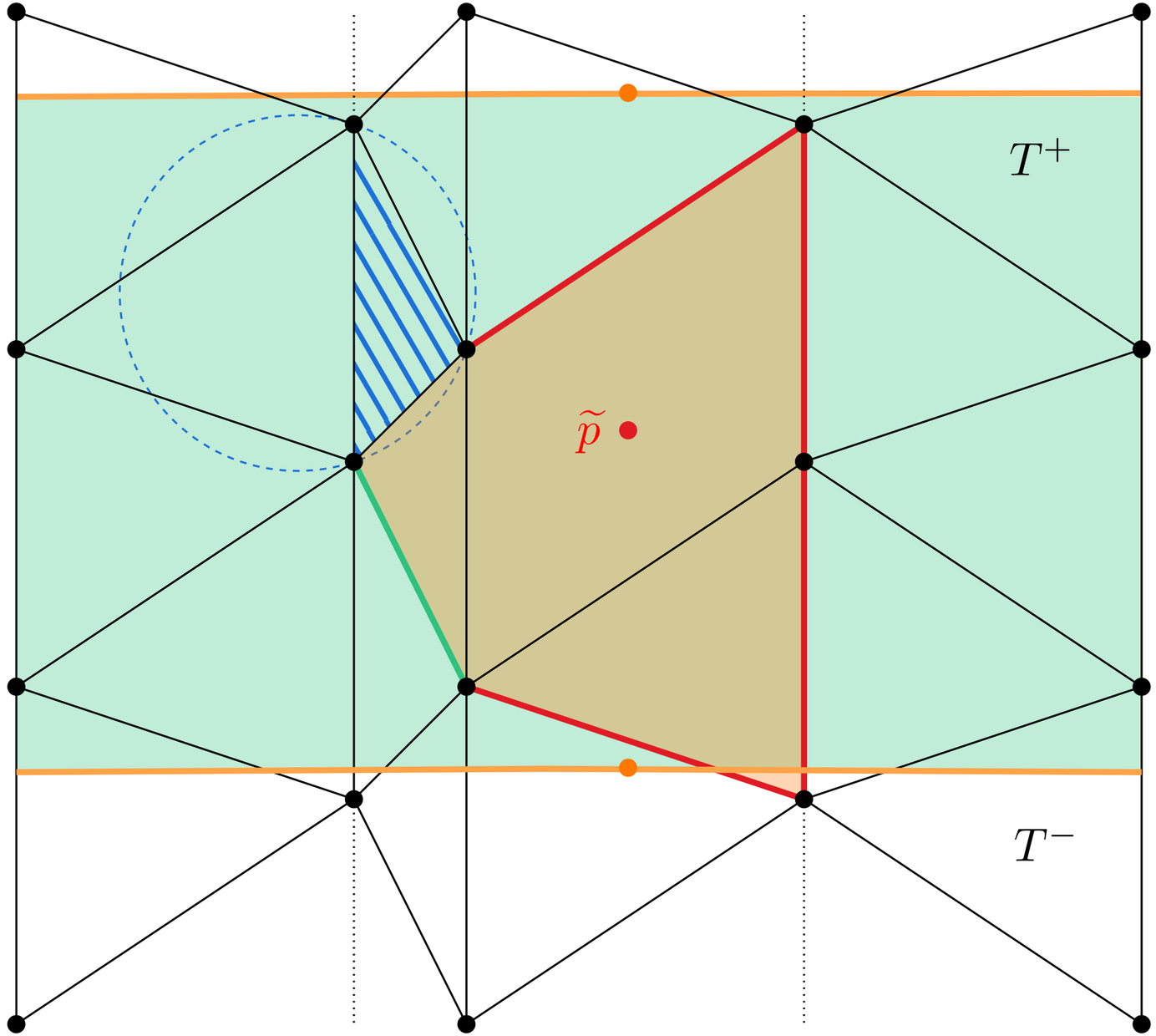


Processing in a banana

→ Conflict with $\widetilde{p}$ **and** has a vertex between the two lines.

→ Push the two other sides in the stack.

Insertion in the 4ε-thin part

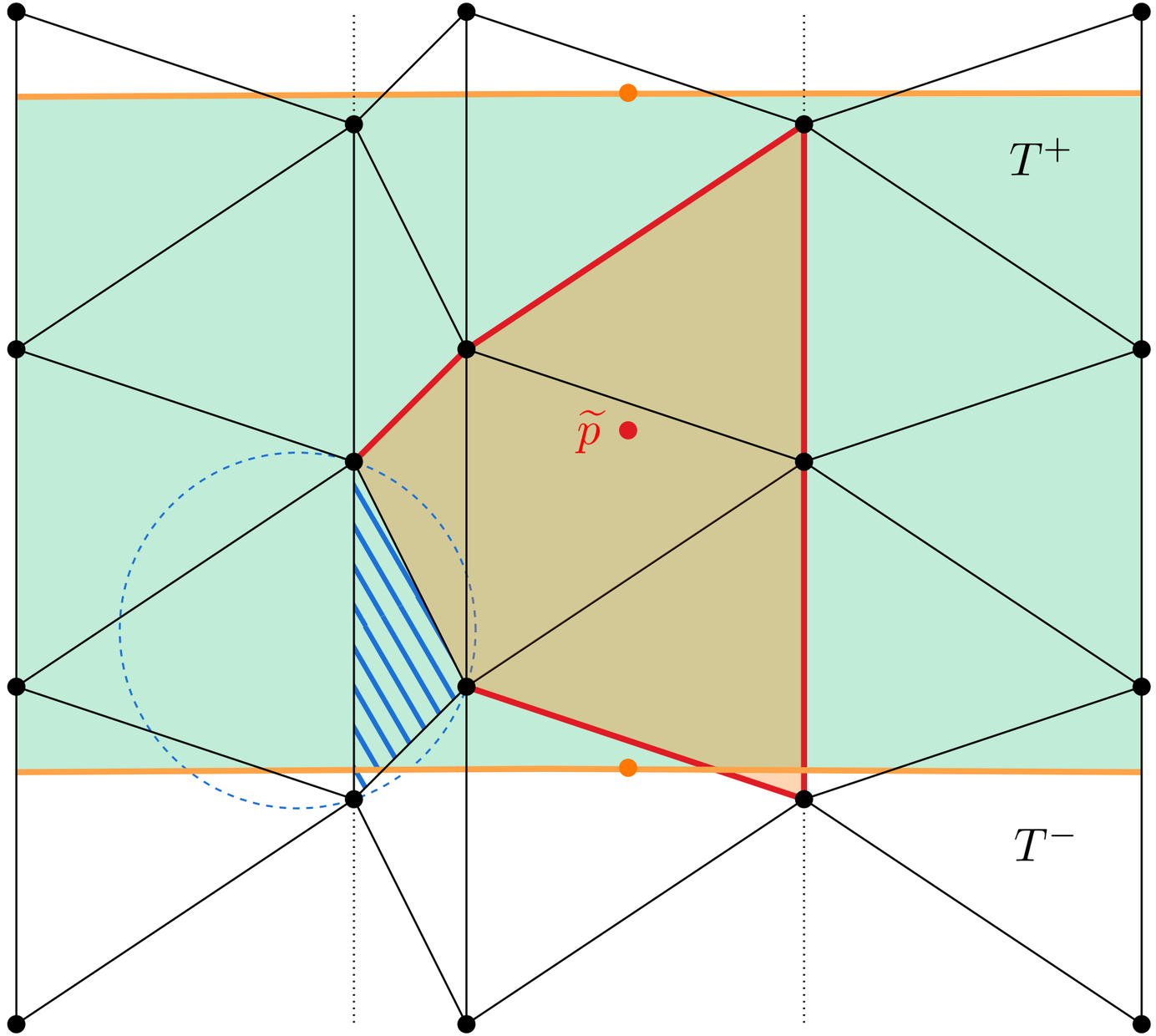


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

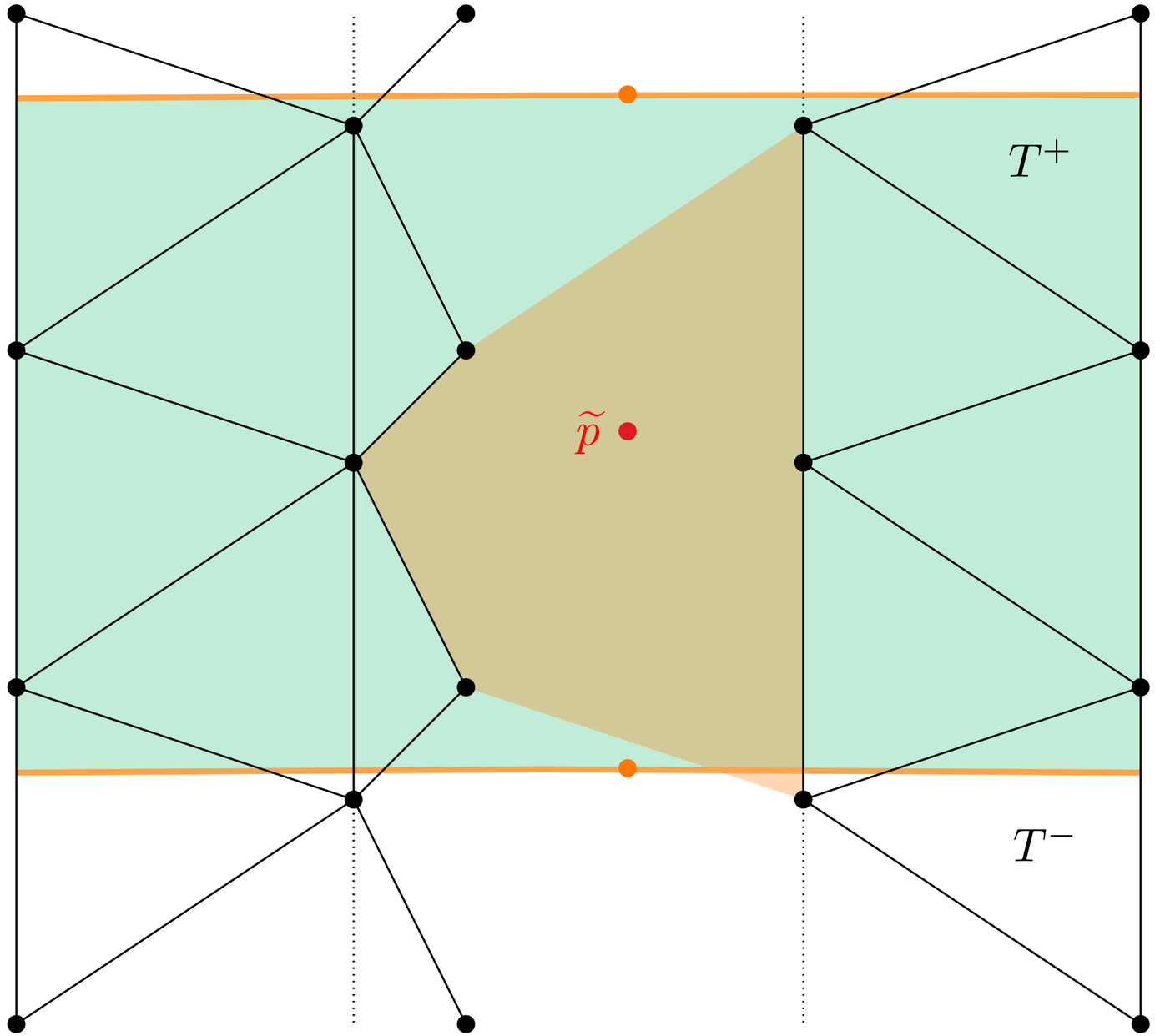


Processing in a banana

→ Not in conflict with $\widetilde{p}$.

→ The side in commun with the conflict zone belongs to the border.

Insertion in the 4ε-thin part

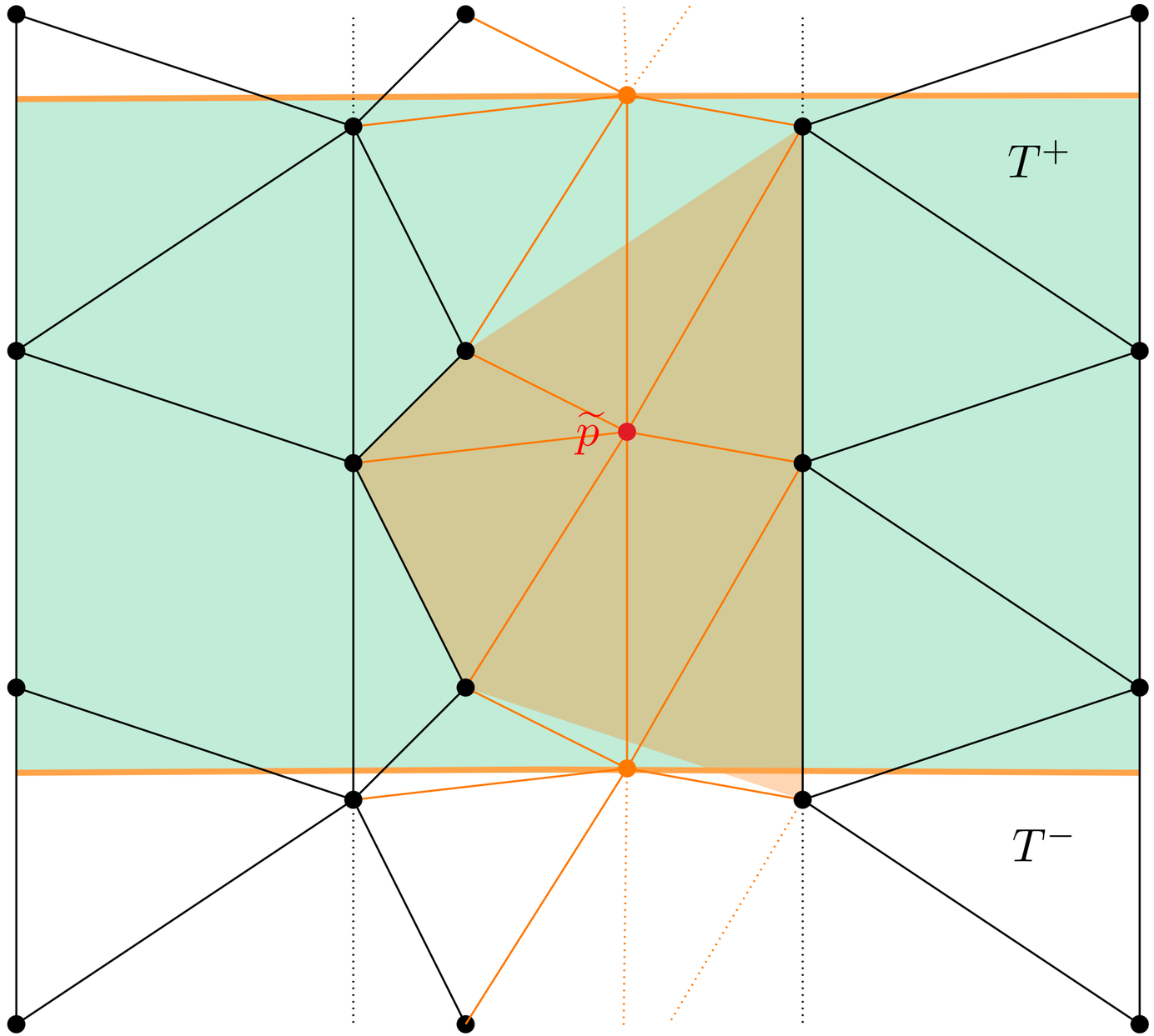


Processing in a banana

→ End of the BFS, we find the restricted conflict zone.

→ Remove all triangles (and their copies) of the restricted conflict zone.

Insertion in the 4ε-thin part



Processing in a banana

→ Connect the border's vertices to $\widetilde{p}$.

→ Check if new triangles are in conflict with two nearest copies of $\widetilde{p}$ and adapt the triangulation.

References

- About Delaunay triangulation and flip algorithm :
  - [Be08] M.de Berg et al. Computational Geometry: Algorithms and Applications
  - [La71]C.Lawson. “Transforming triangulations”
- About Delaunay triangulation on hyperbolic surfaces :
  - [DST24] V. Despré, J-M. Schlenker, and M. Teillaud. “Flipping Geometric Triangulations on Hyperbolic Surfaces”
  - [IT17] I. Iordanov and M. Teillaud : “Implementing Delaunay Triangulations of the Bolza Surface”
- The original Bowyer’s article :
  - [Bo81] A. Bowyer. “Computing Dirichlet tessellations”
- The surface initialization algorithm can be found in :
  - [DDLPT26](preprint) V. Delecroix, V. Despré, H. Parlier, C. Lanuel, and M. Teillaud. “Computing an ε-net of a closed hyperbolic surface”
- Collar’s lemma
  - [Bu10] P. Buser. Geometry and Spectra of Compact Riemann Surfaces