Topological Algorithms-II

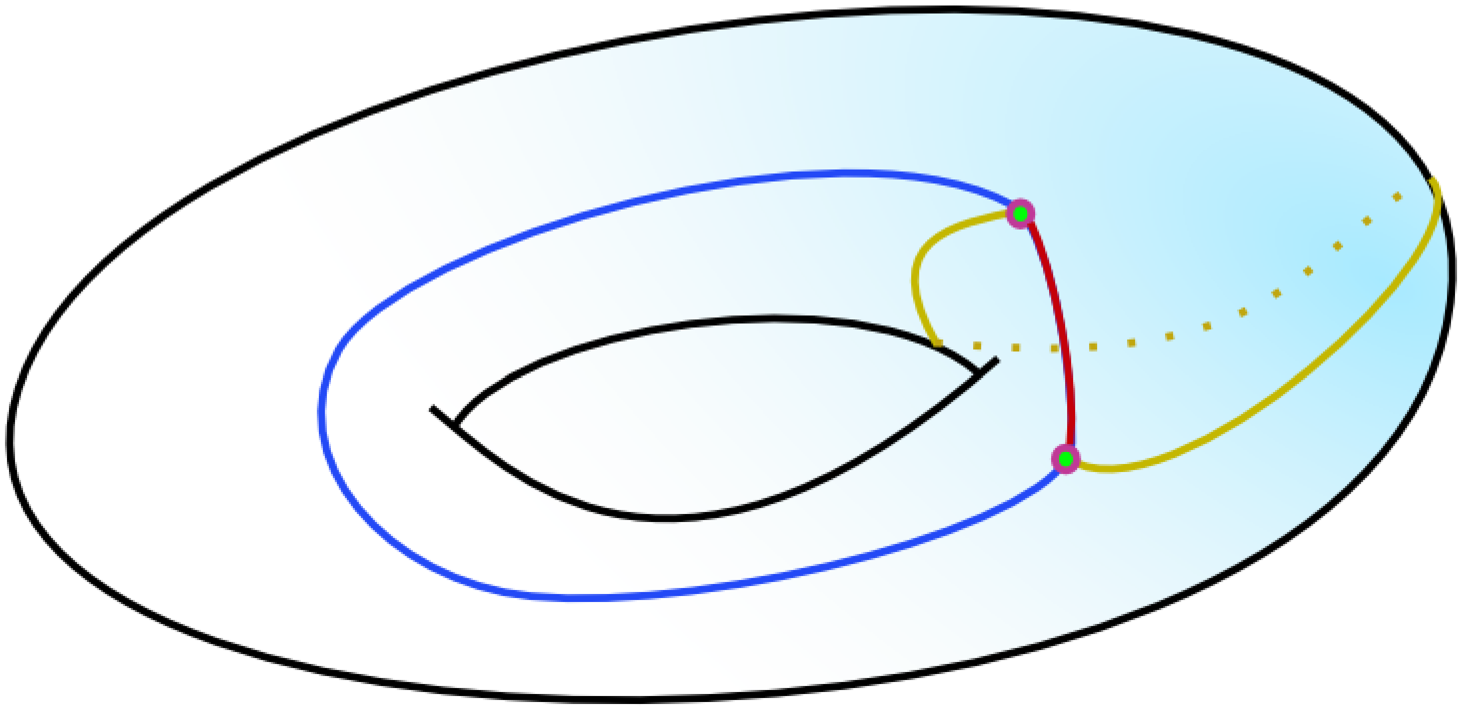
Mustafa Hajij

Cutting Graph

A cut graph G of a mesh M consists of a set of edges of M such that M/G is a simply connected mesh (topological disk).

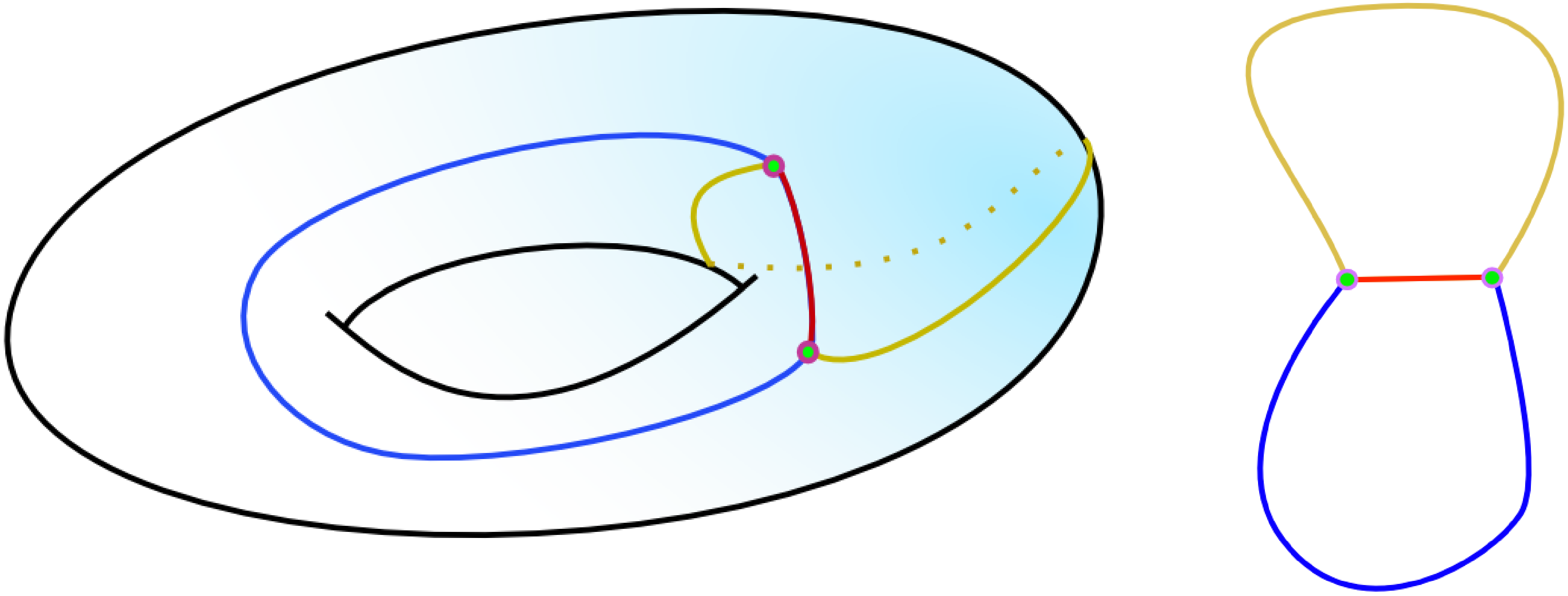
Cutting Graph

A cut graph G of a mesh M consists of a set of edges of M such that M/G is a simply connected mesh (topological disk).



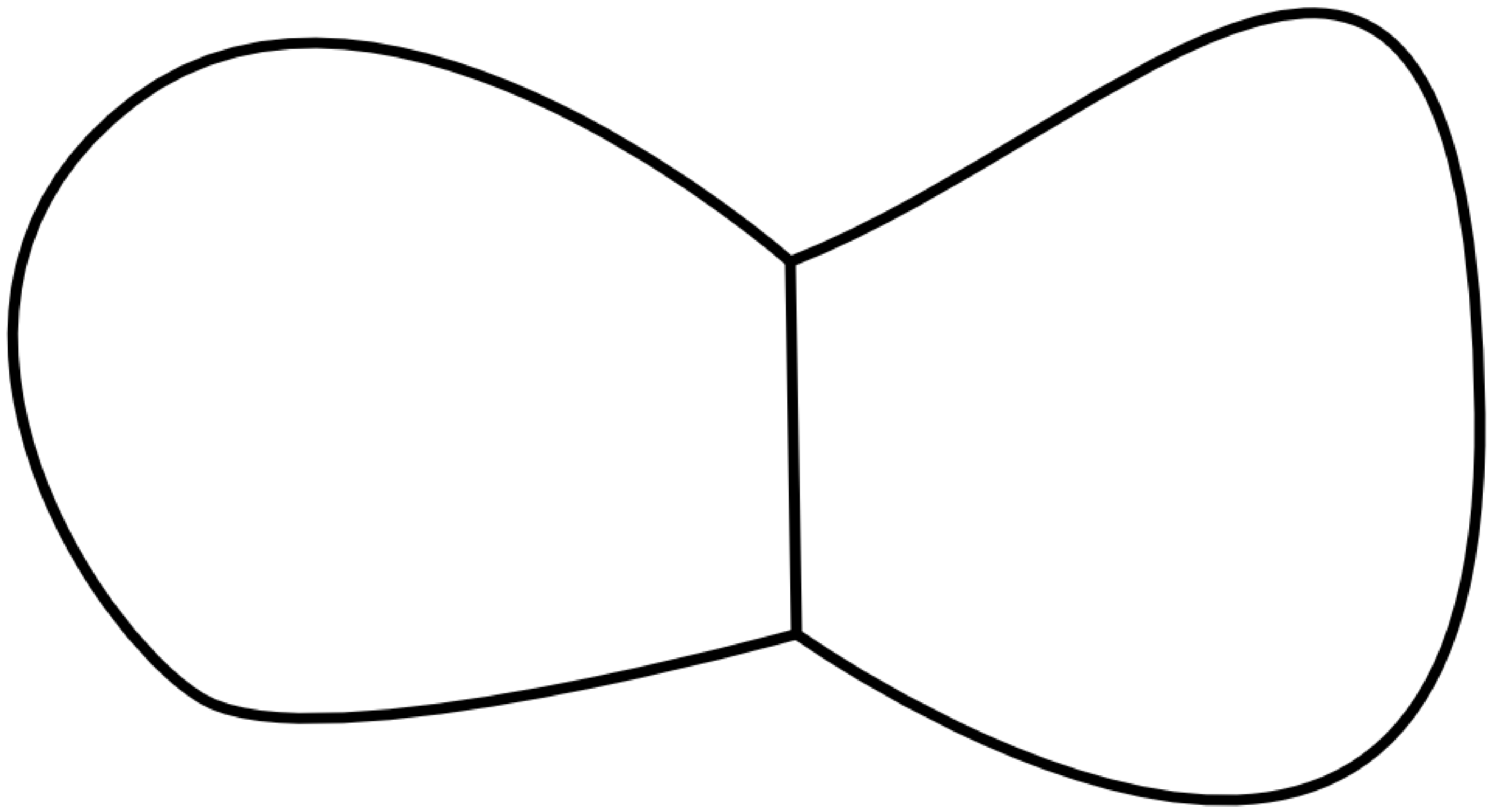
Cutting Graph

A cut graph G of a mesh M consists of a set of edges of M such that M/G is a simply connected mesh (topological disk).



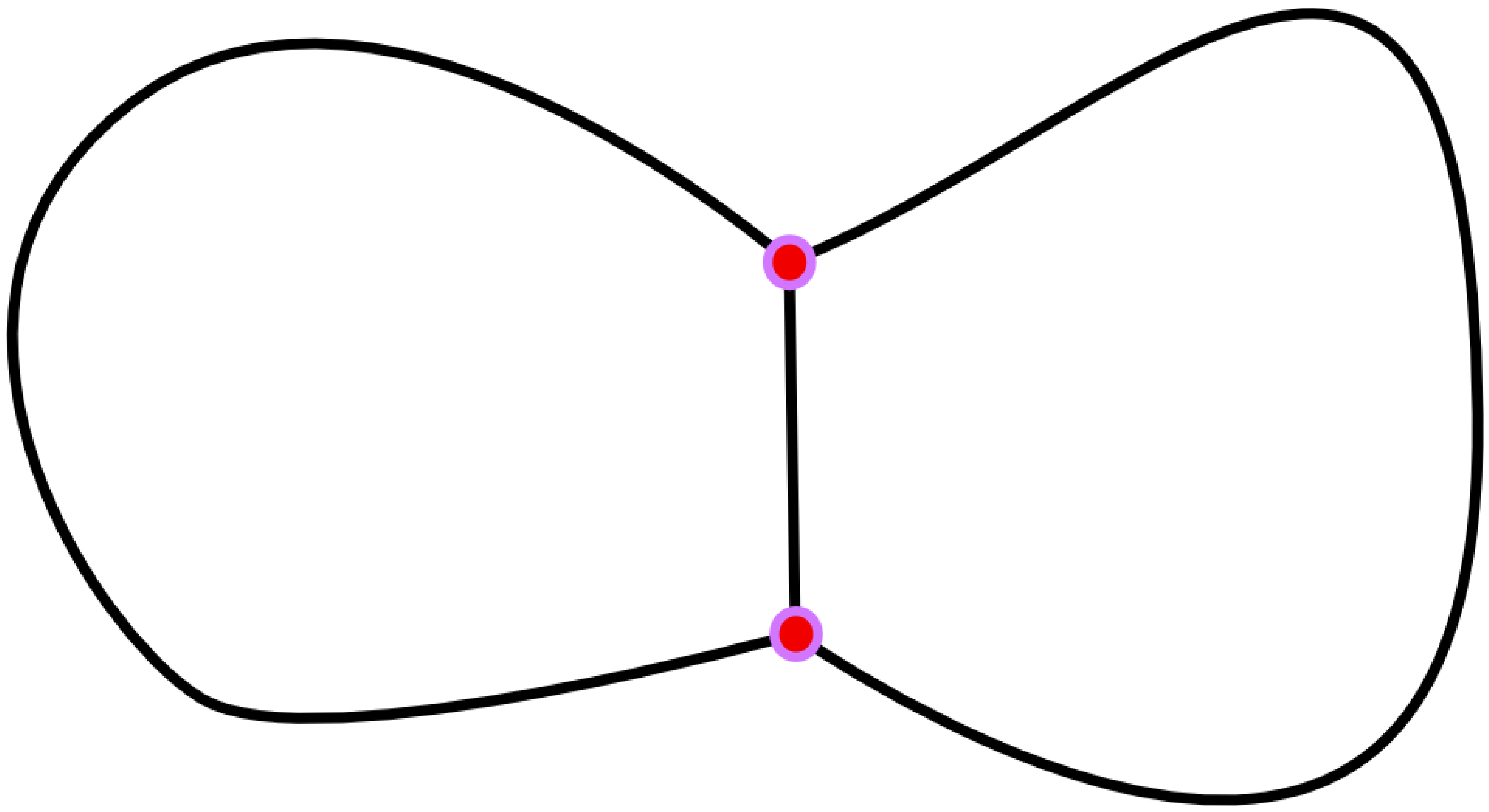
Cutting Graph

A cut graph G of a mesh M consists of a set of edges of M such that M/G is a simply connected mesh (topological disk).



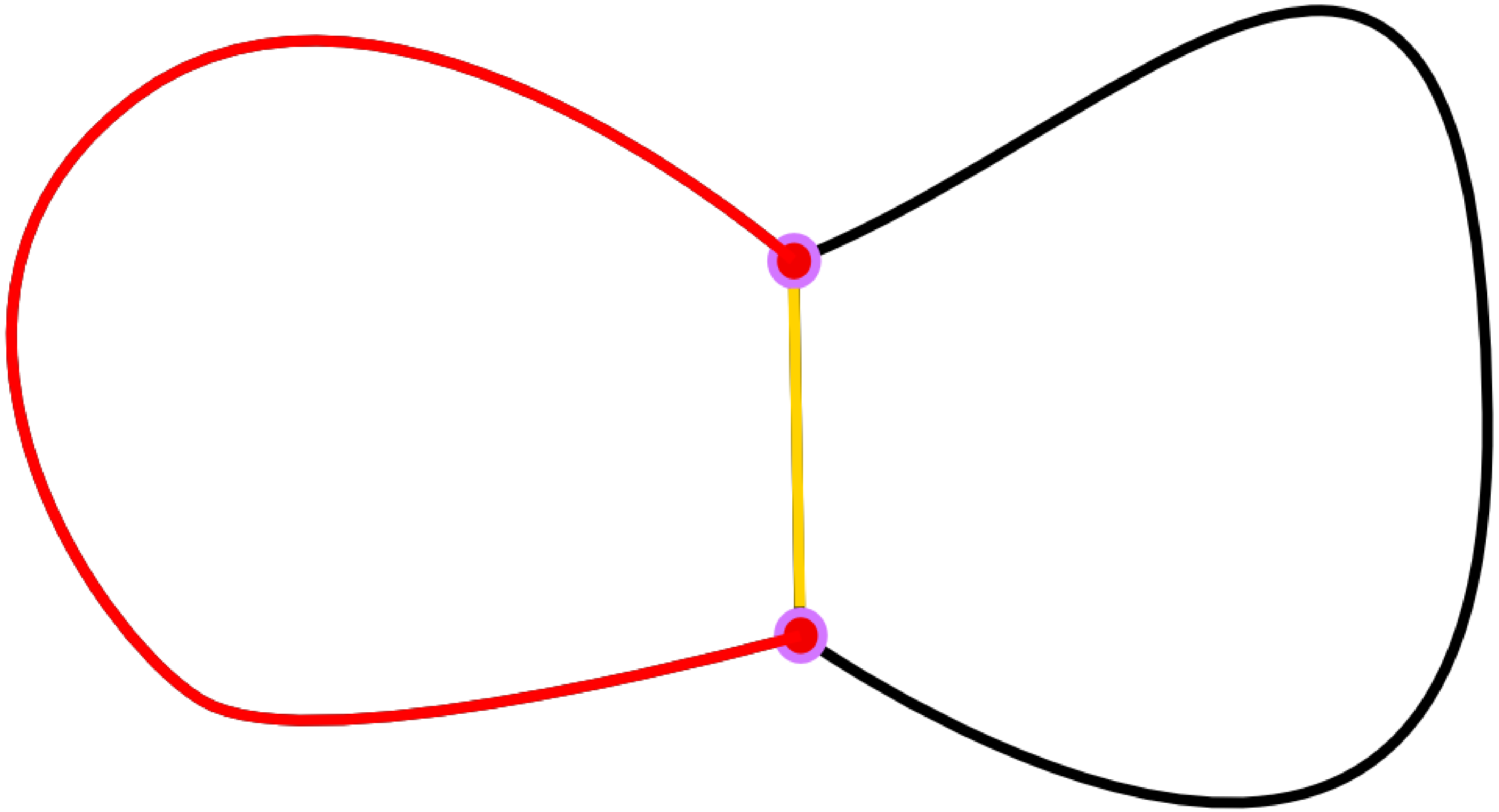
Cutting Graph

On a cutting graph G we locate the *Branching vertices*



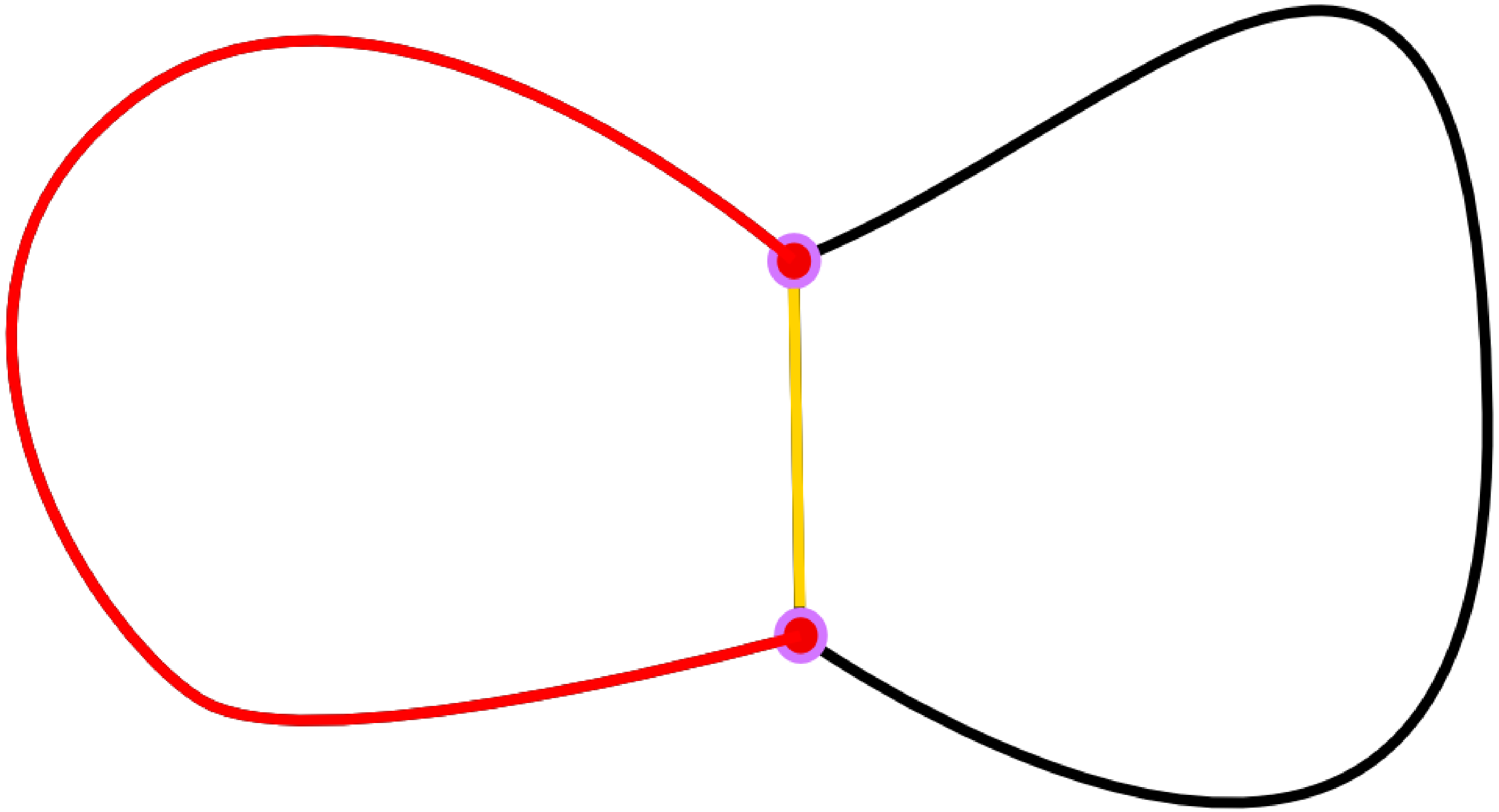
Cutting Graph

A *segment* in a cutting graph G is a set of consecutive edges in G, which connect two branching vertices.



Cutting Graph

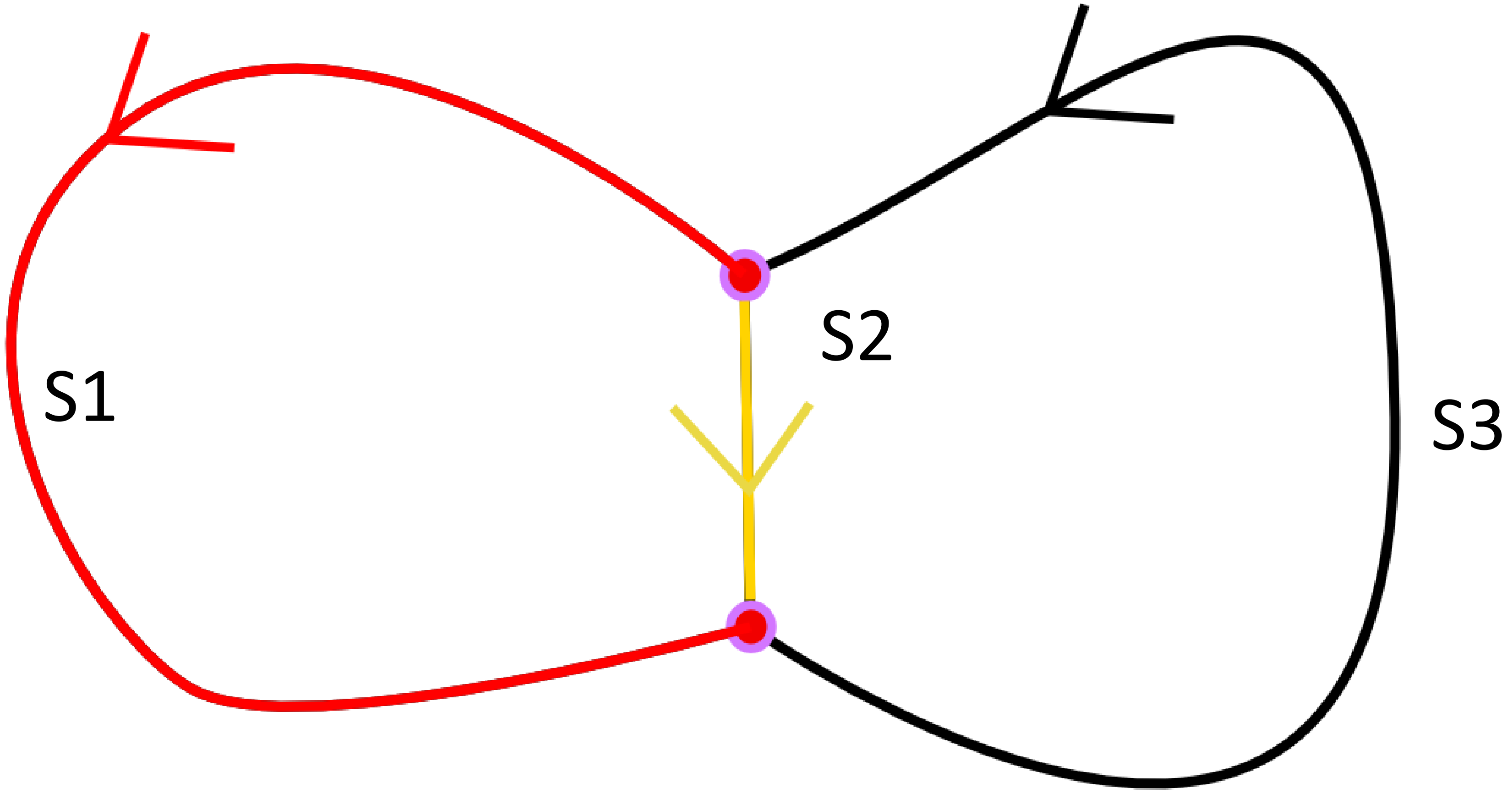
A *segment* in a cutting graph G is a set of consecutive edges in G, which connect two branching vertices.



Namely the branching vertices separate the cut graph into a collection of segments.

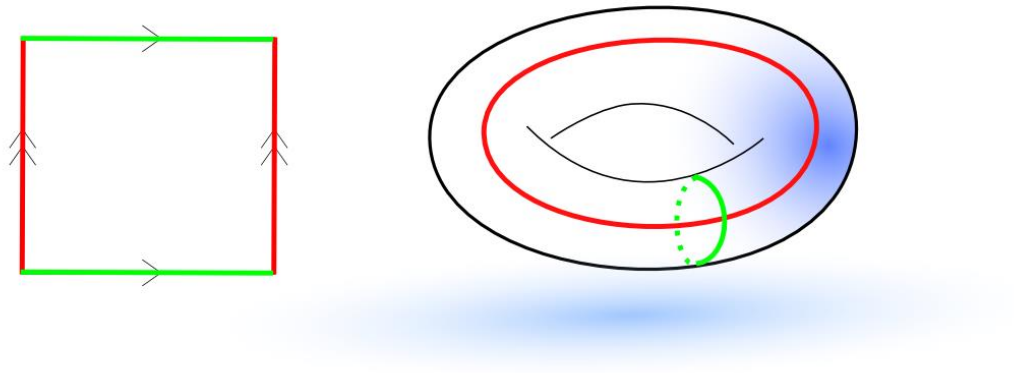
Cutting Graph

We give each segment an arbitrary orientation and denote the oriented segments by S={s_1,...,sn}

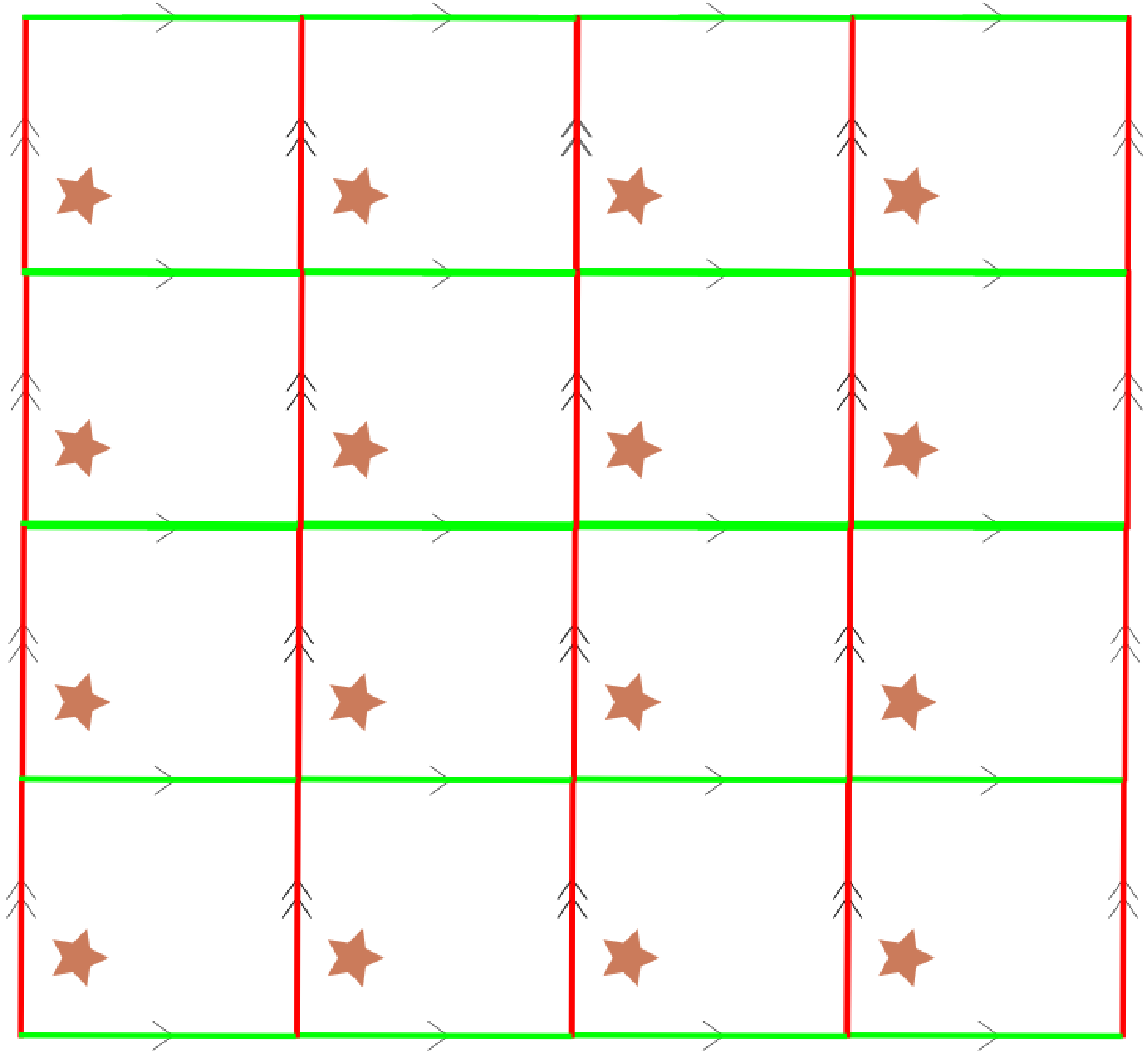


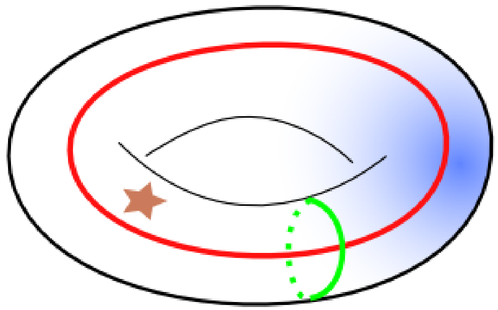
Computing the fundamental domain

A closed subset D of the universal cover $\tilde{M}$ of M is called a fundamental domain of $\tilde{M}$ if $\tilde{M}$ is the union of conjugates of D



The universal cover and the fundamental domain



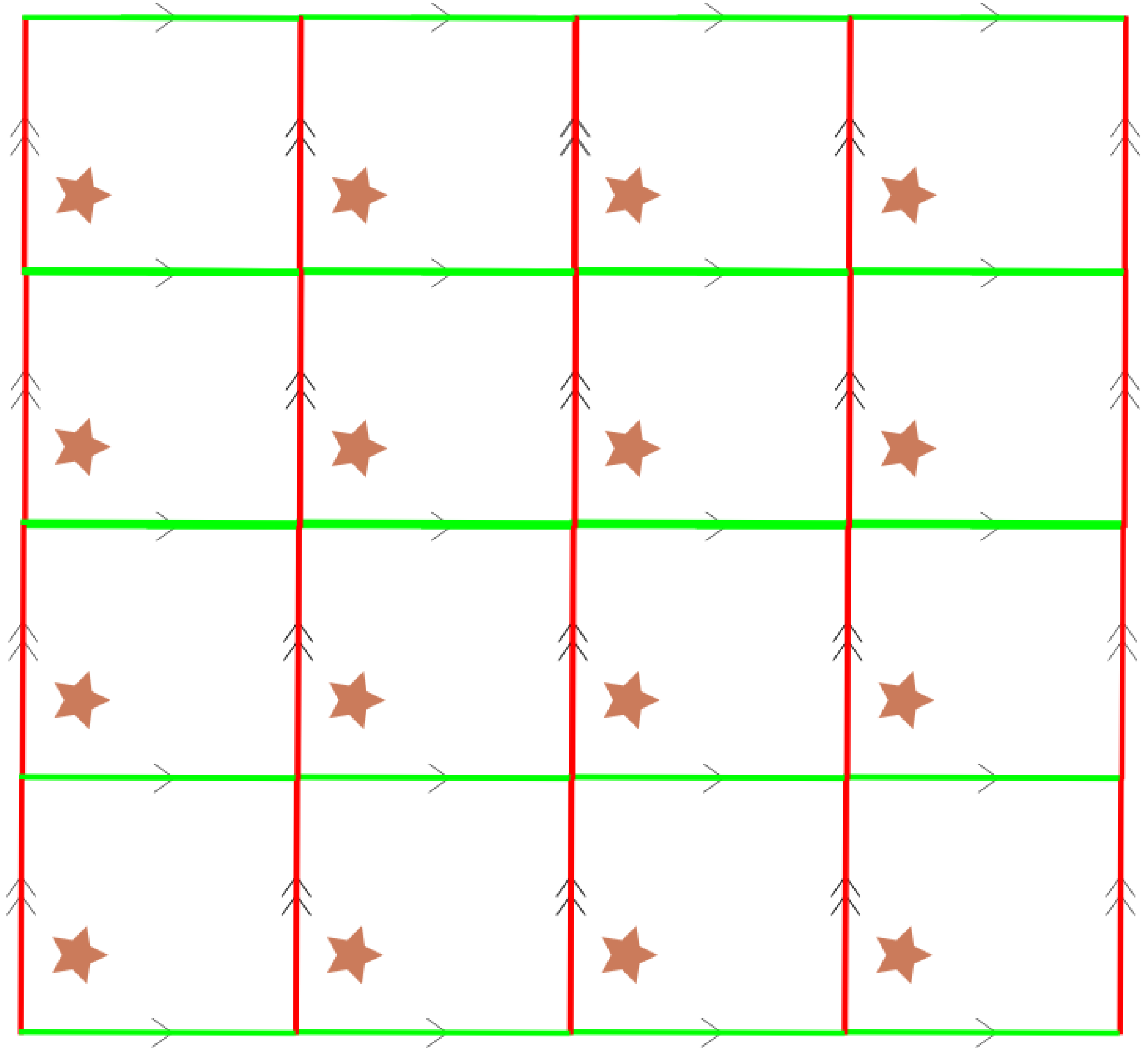


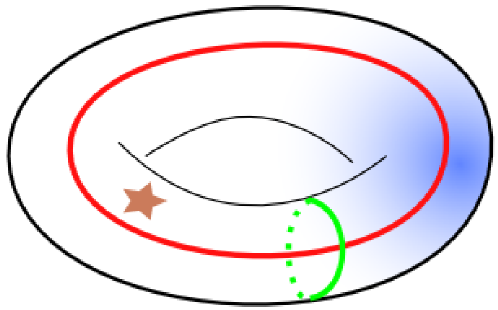
Computing the fundamental domain

Input : A mesh M
Output : A fundamental domain D of M

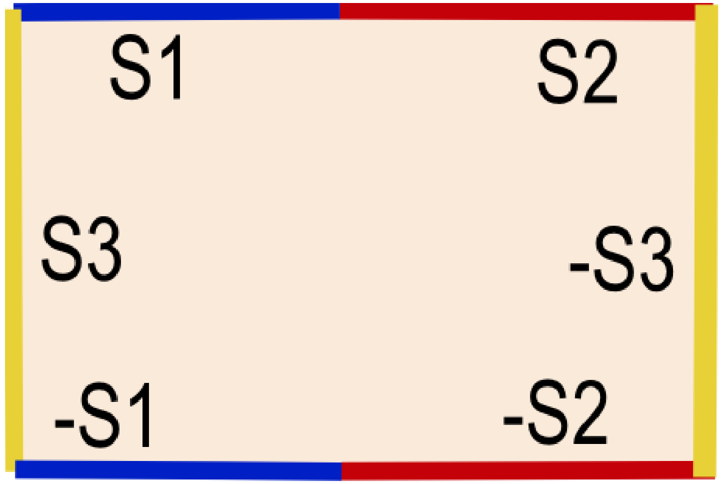
- Compute a cut graph G of M
- Slice M along G

The universal cover and the fundamental domain

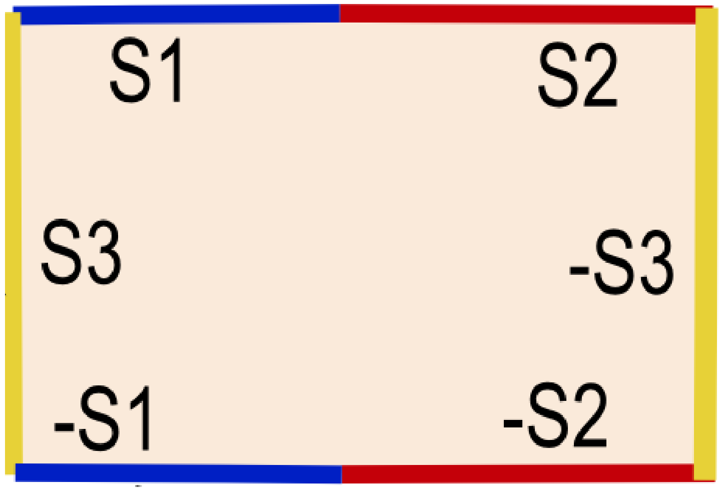


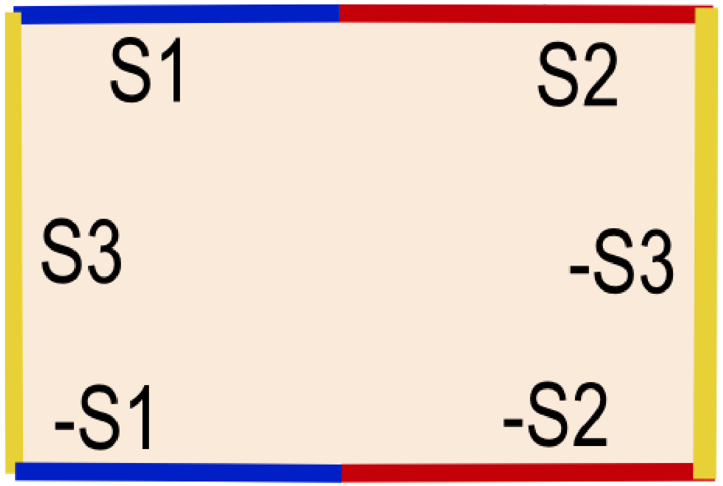


Constructing the universal cover

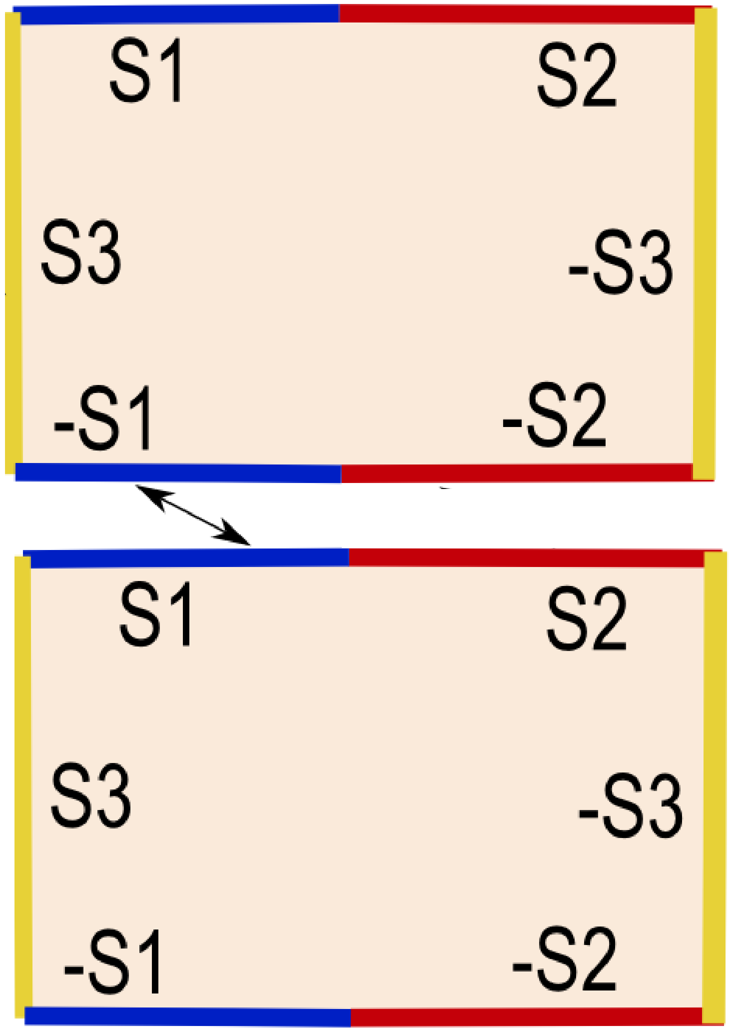


Constructing the universal cover

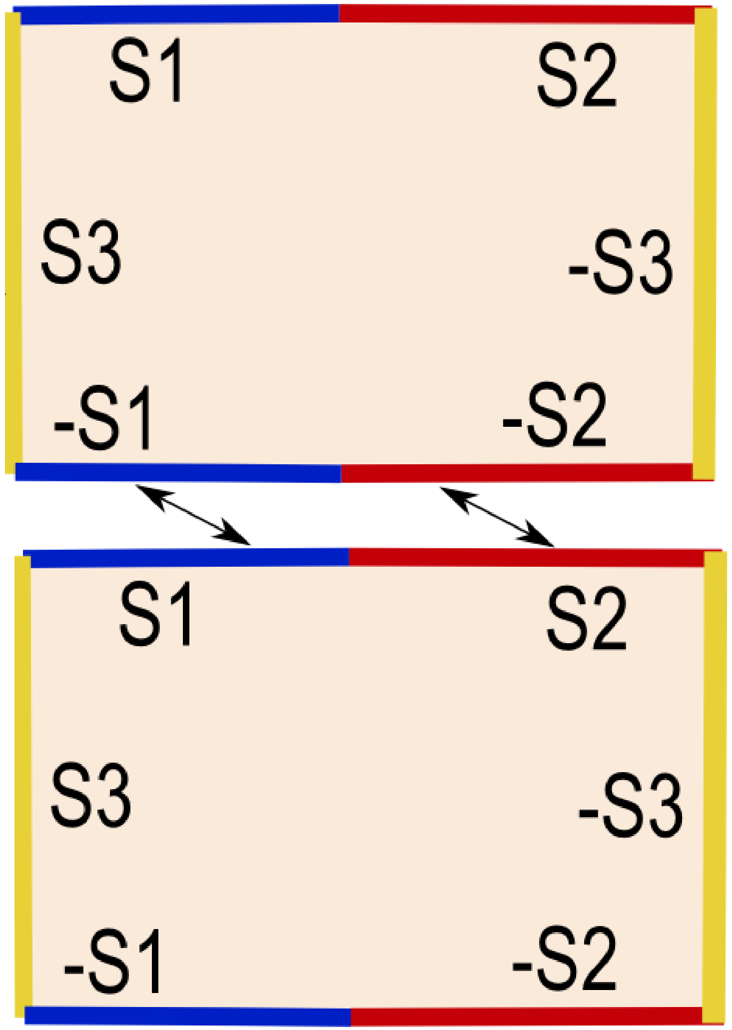




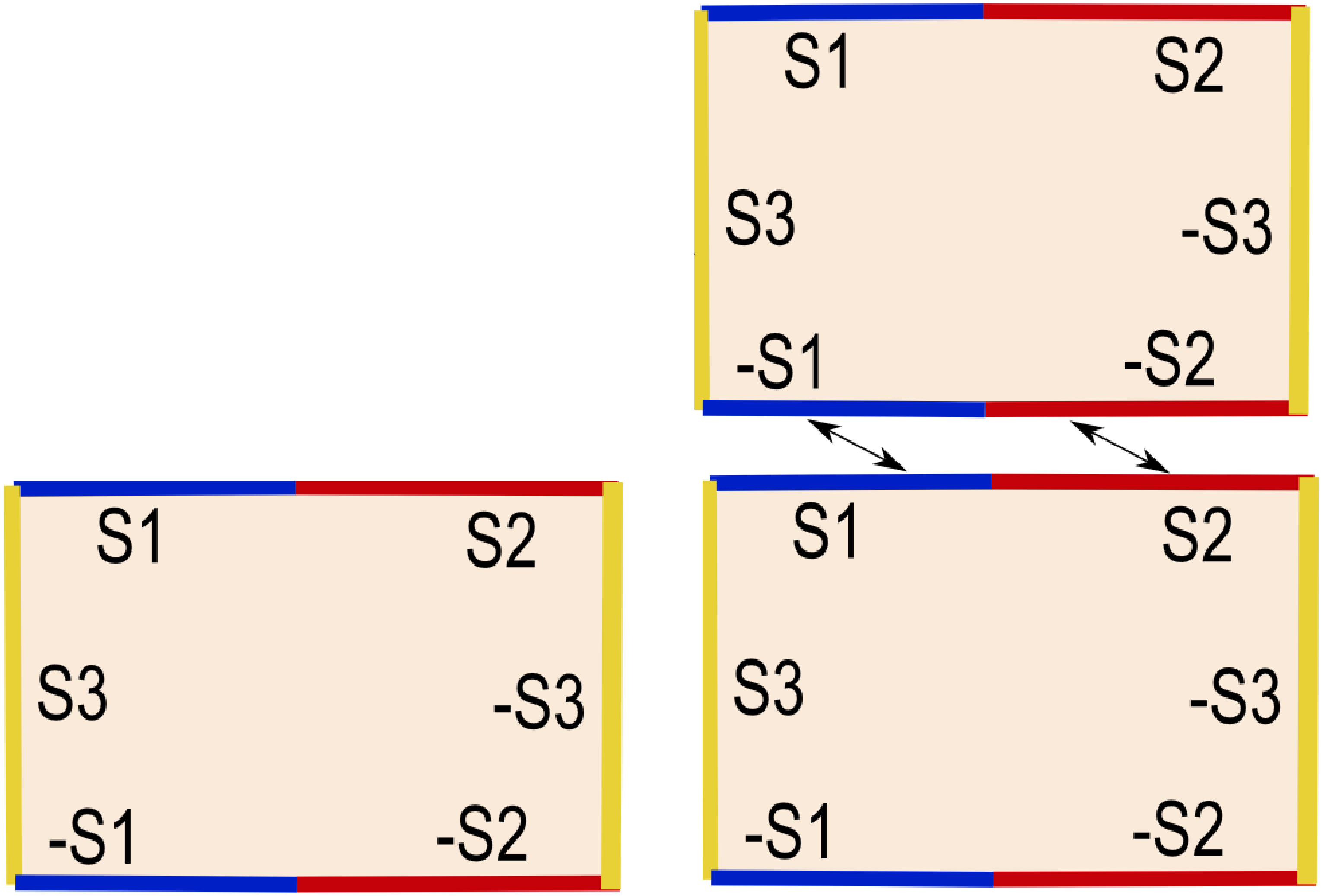
Constructing the universal cover



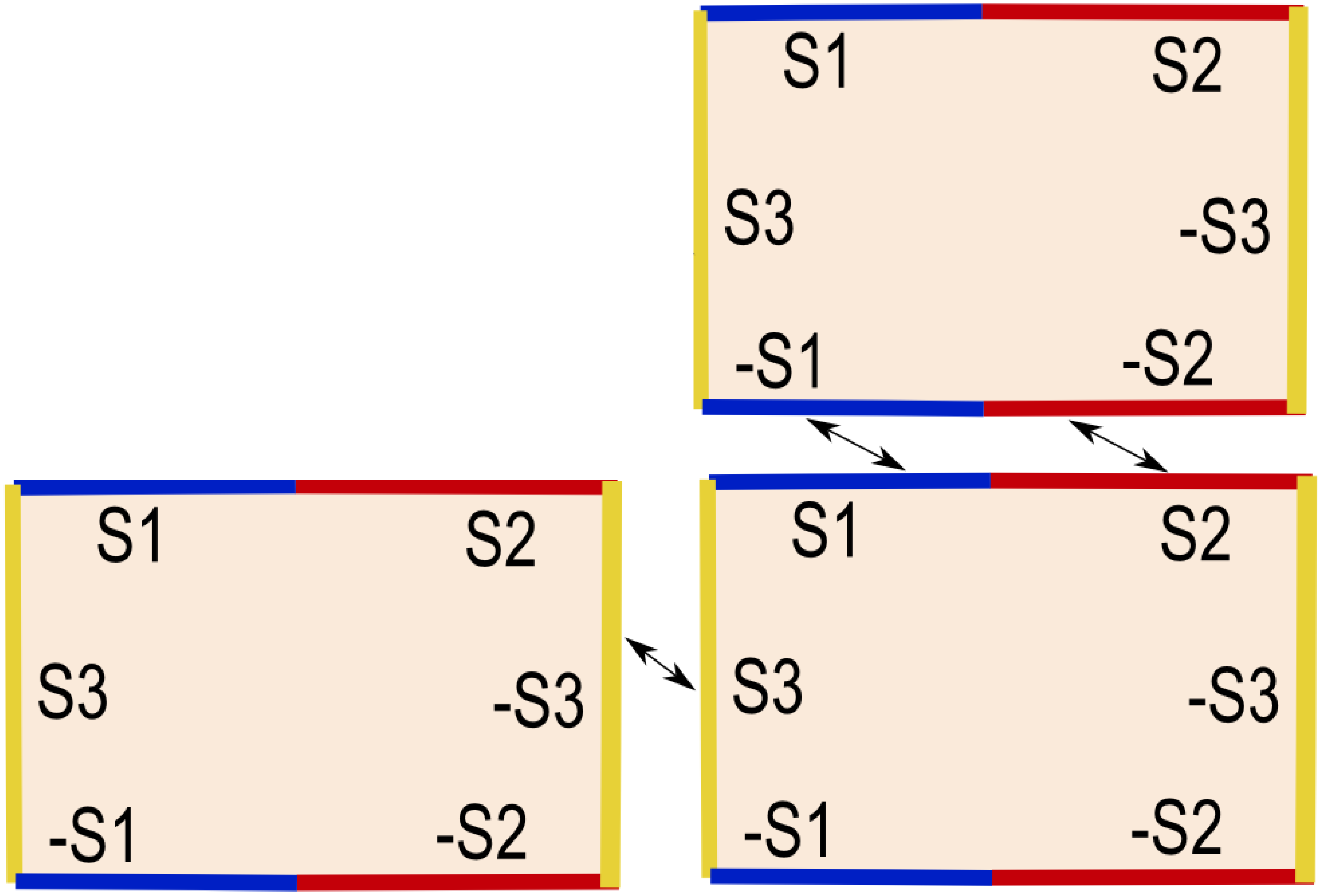
Constructing the universal cover



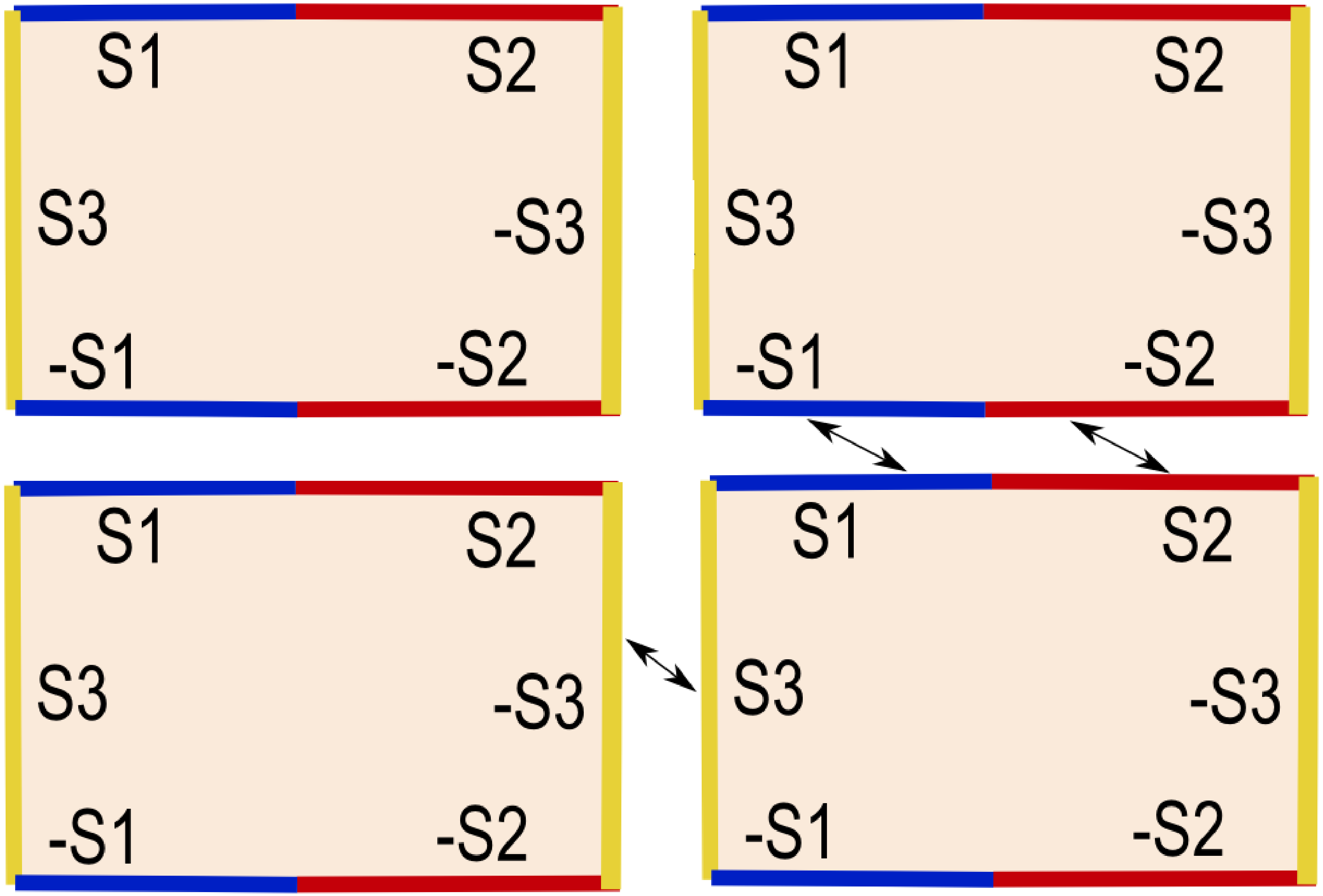
Constructing the universal cover



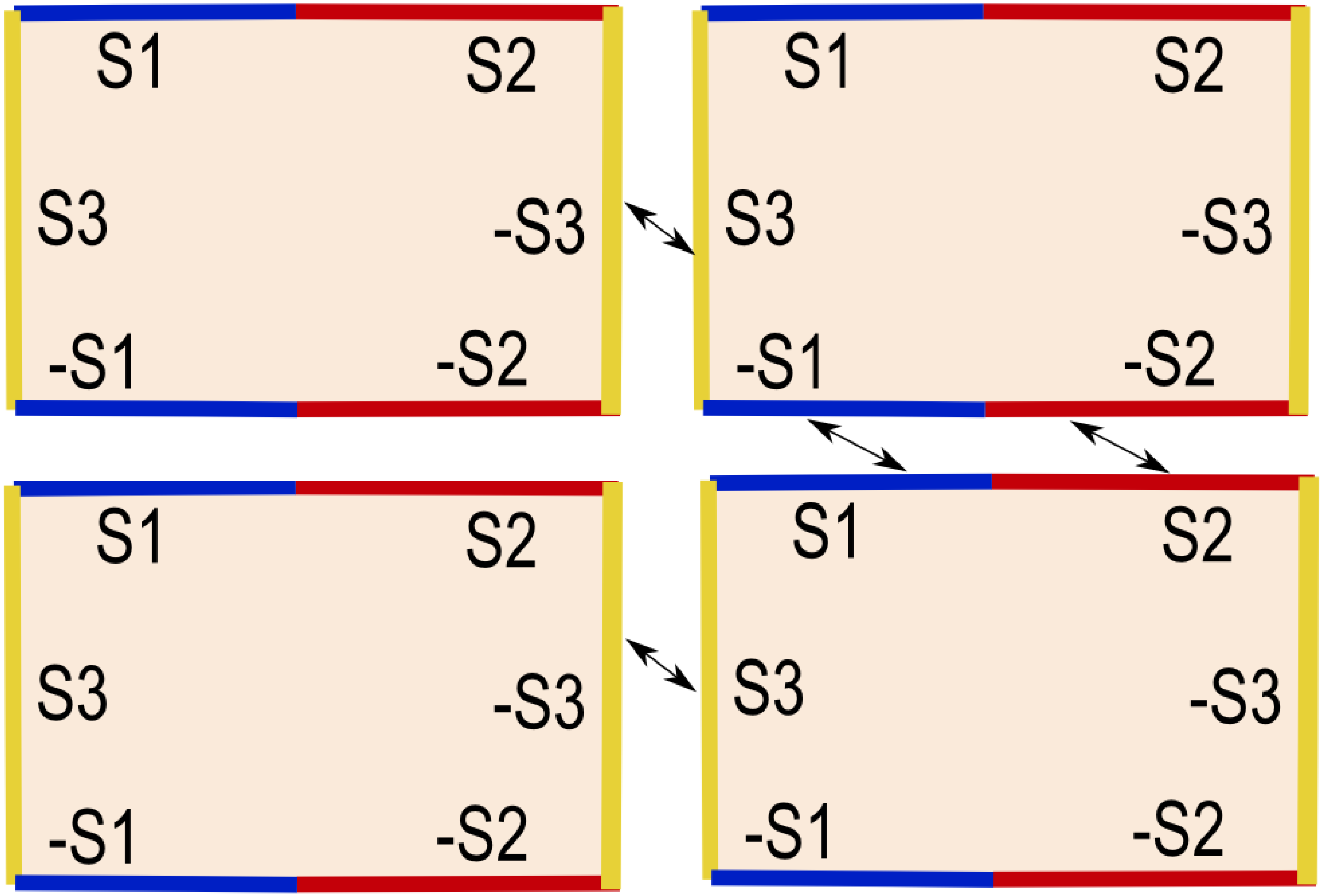
Constructing the universal cover



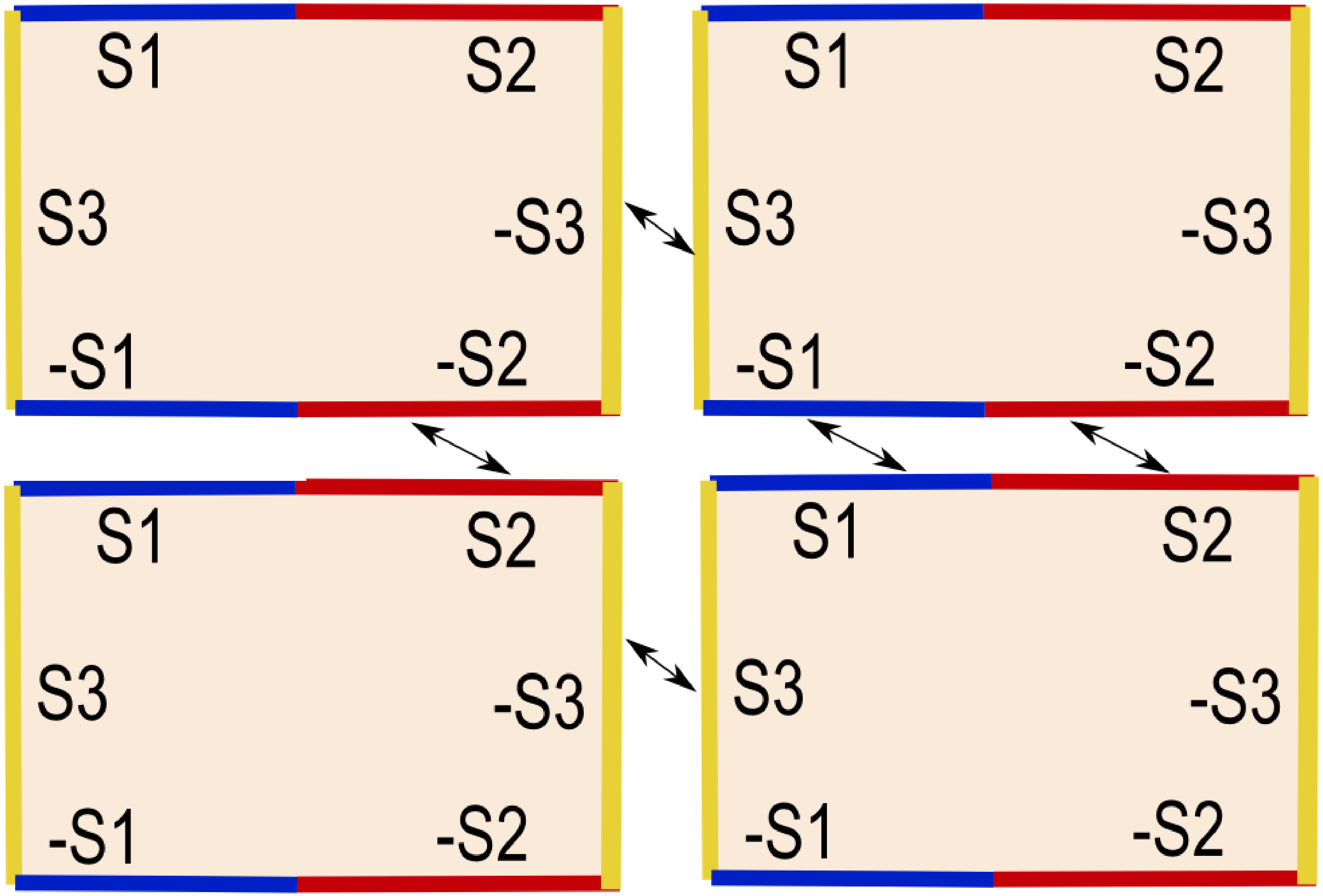
Constructing the universal cover



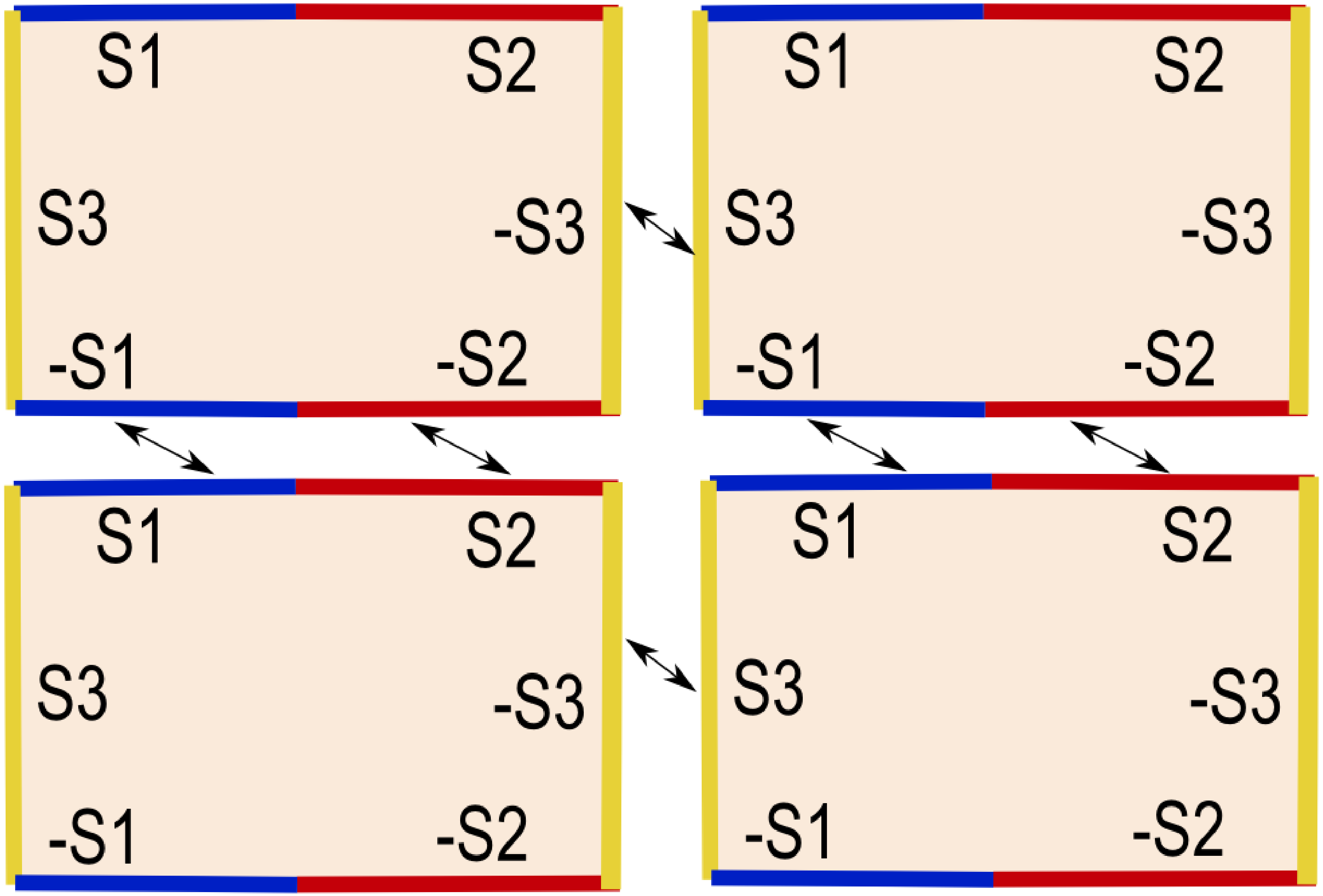
Constructing the universal cover



Constructing the universal cover



Constructing the universal cover



Constructing the universal cover

Input : A mesh Σ.

Output: A finite portion of the universal cover Σ^{-}.

Constructing the universal cover

Input : A mesh Σ.

Output: A finite portion of the universal cover $\bar{\Sigma}$.

1 Compute a cut graph G of Σ.

We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as {s1, s2, · · · , sn}.

Constructing the universal cover

Input : A mesh Σ.

Output: A finite portion of the universal cover $\bar{\Sigma}$.

1 Compute a cut graph G of Σ.

We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as {s1, s2, · · · , sn}.

2 Slice Σ along G to get a fundamental domain D, the boundary is composed of ±sk ’s.

Constructing the universal cover

Input : A mesh Σ.

Output: A finite portion of the universal cover $\bar{\Sigma}$.

1 Compute a cut graph G of Σ.

We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as {s1, s2, · · · , sn}.

2 Slice Σ along G to get a fundamental domain D, the boundary is composed of ±sk 's.

3 Initialize $\bar{\Sigma} \leftarrow D$, book keep $\partial\bar{\Sigma}$ using ±sk 's.

Constructing the universal cover

Input : A mesh Σ.
Output: A finite portion of the universal cover $\bar{\Sigma}$.
1 Compute a cut graph G of Σ.
We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as $\{s1, s2, \cdots, sn\}$.

2 Slice Σ along G to get a fundamental domain D, the boundary is composed of $\pm sk$'s.

3 Initialize $\bar{\Sigma} \leftarrow D$, book keep $\partial\bar{\Sigma}$ using $\pm sk$'s.

4 Glue a copy of D to current $\bar{\Sigma}$ along only one segment $sk \in \partial\bar{\Sigma}$, $-sk \in \partial D$, $\bar{\Sigma} \leftarrow \bar{\Sigma} \cup_{sk} D$.

Constructing the universal cover

Input : A mesh Σ.

Output: A finite portion of the universal cover $\bar{\Sigma}$.

1 Compute a cut graph G of Σ.

We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as $\{s1, s2, \cdots, sn\}$.

2 Slice Σ along G to get a fundamental domain D, the boundary is composed of $\pm sk$ ’s.

3 Initialize $\bar{\Sigma} \leftarrow D$, book keep $\partial\bar{\Sigma}$ using $\pm sk$ ’s.

4 Glue a copy of D to current $\bar{\Sigma}$ along only one segment $sk \in \partial\bar{\Sigma}$, $-sk \in \partial D$, $\bar{\Sigma} \leftarrow \bar{\Sigma} \cup_{sk} D$.

5 Update $\partial\bar{\Sigma}$, if $\pm si$ are adjacent in $\partial\bar{\Sigma}$, glue the boundary of $\bar{\Sigma}$ along si . Repeat this step until no adjacent $\pm si$ in the boundary.

Constructing the universal cover

Input : A mesh Σ.
Output: A finite portion of the universal cover Σ¯.
1 Compute a cut graph G of Σ.
We call a vertex on G with valence greater than 2 a *Branching vertex*. The *Branching vertices* divide G to segments, assign an orientation to each segment, labeled as {s1, s2, · · · , sn}.

2 Slice Σ along G to get a fundamental domain D, the boundary is composed of ±sk 's.

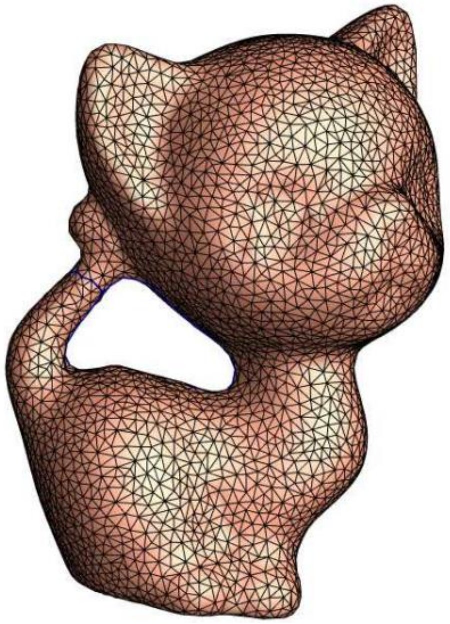
3 Initialize Σ¯ ← D, book keep ∂Σ¯ using ±sk 's.

4 Glue a copy of D to current Σ¯ along only one segment sk ∈ ∂Σ¯, −sk ∈ ∂D, Σ¯ ← Σ¯ $\cup_{sk}$ D.

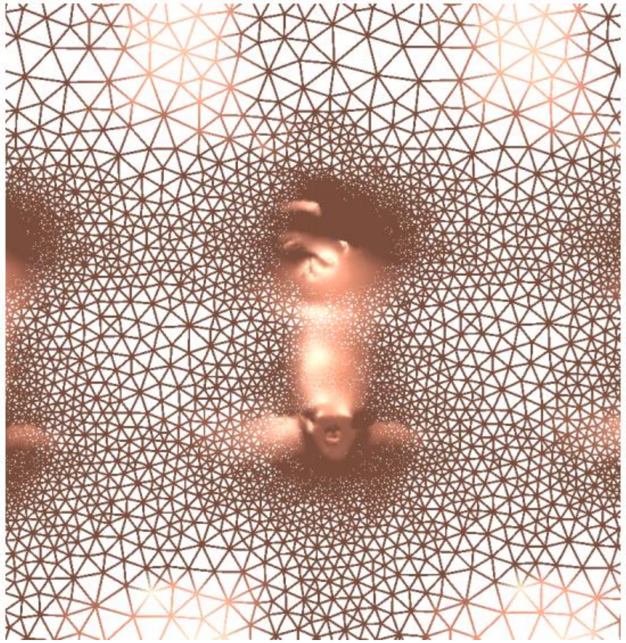
5 Update ∂Σ¯, if ±si are adjacent in ∂Σ¯, glue the boundary of Σ¯ along si . Repeat this step until no adjacent ±si in the boundary.

6 Repeat gluing the copies of D until Σ¯ is large enough.

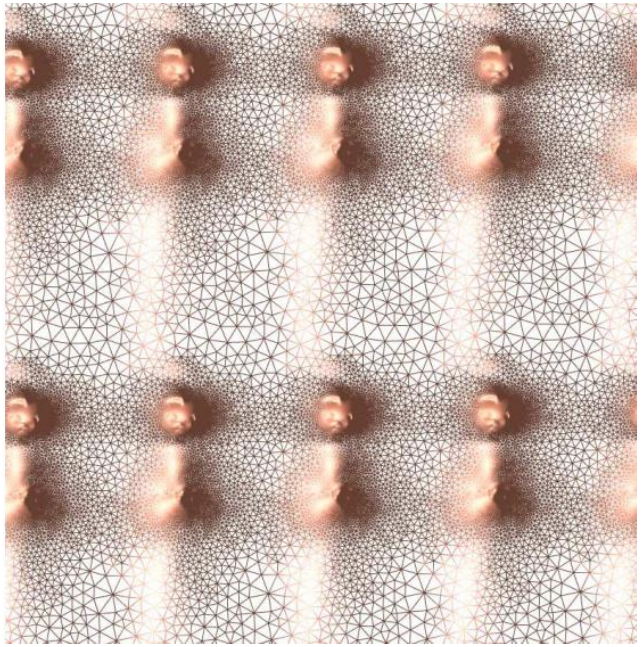
Constructing the universal cover



Input mesh



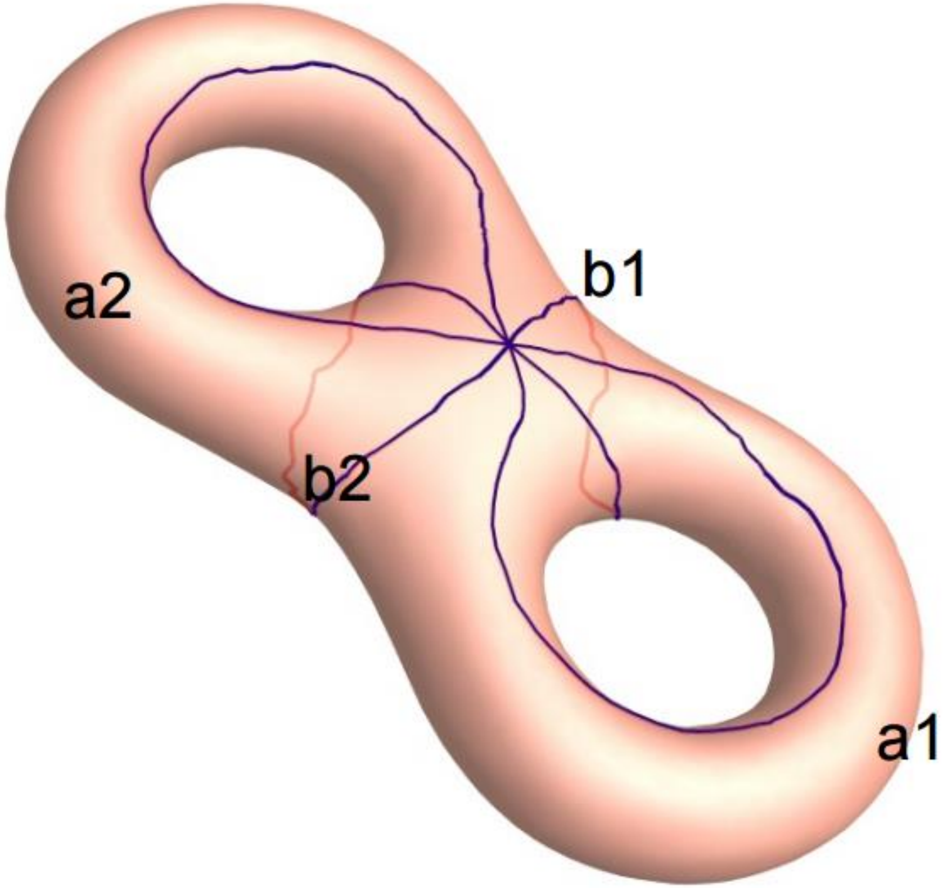
Fundamental domain



Finite portion of the universal cover

Image :David Gu

Constructing the universal cover



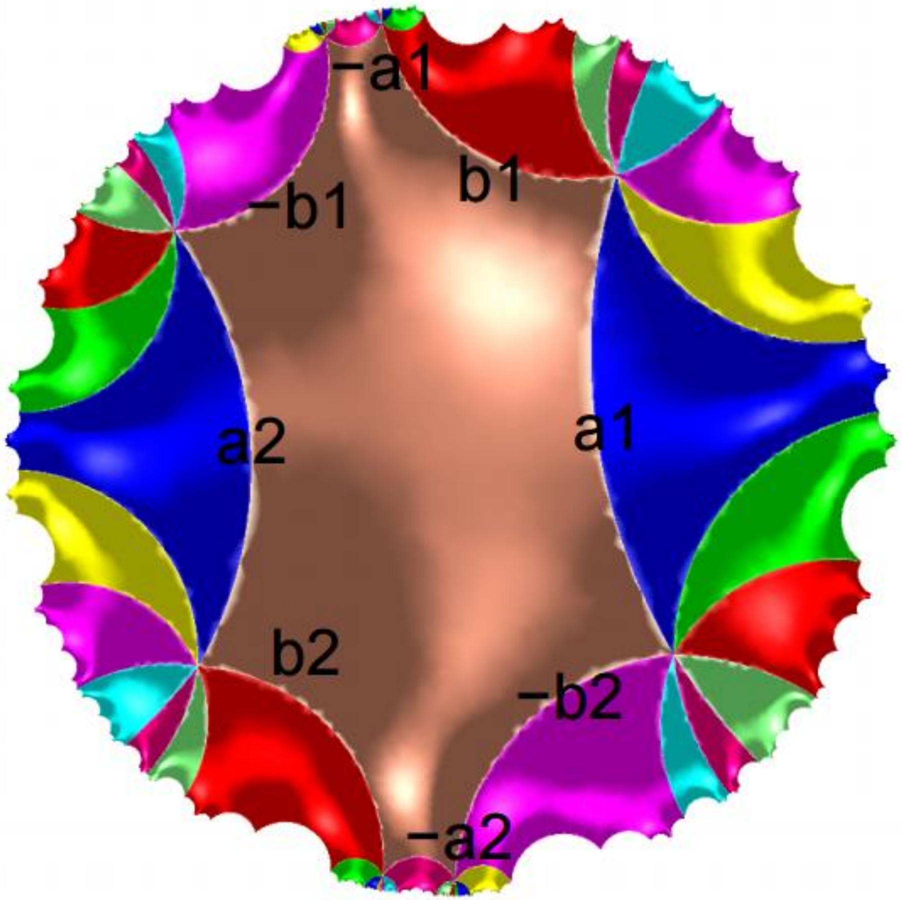


Image :David Gu

Curve lifting

A curve in the original surface can be lifted to the universal covering space

Curve lifting

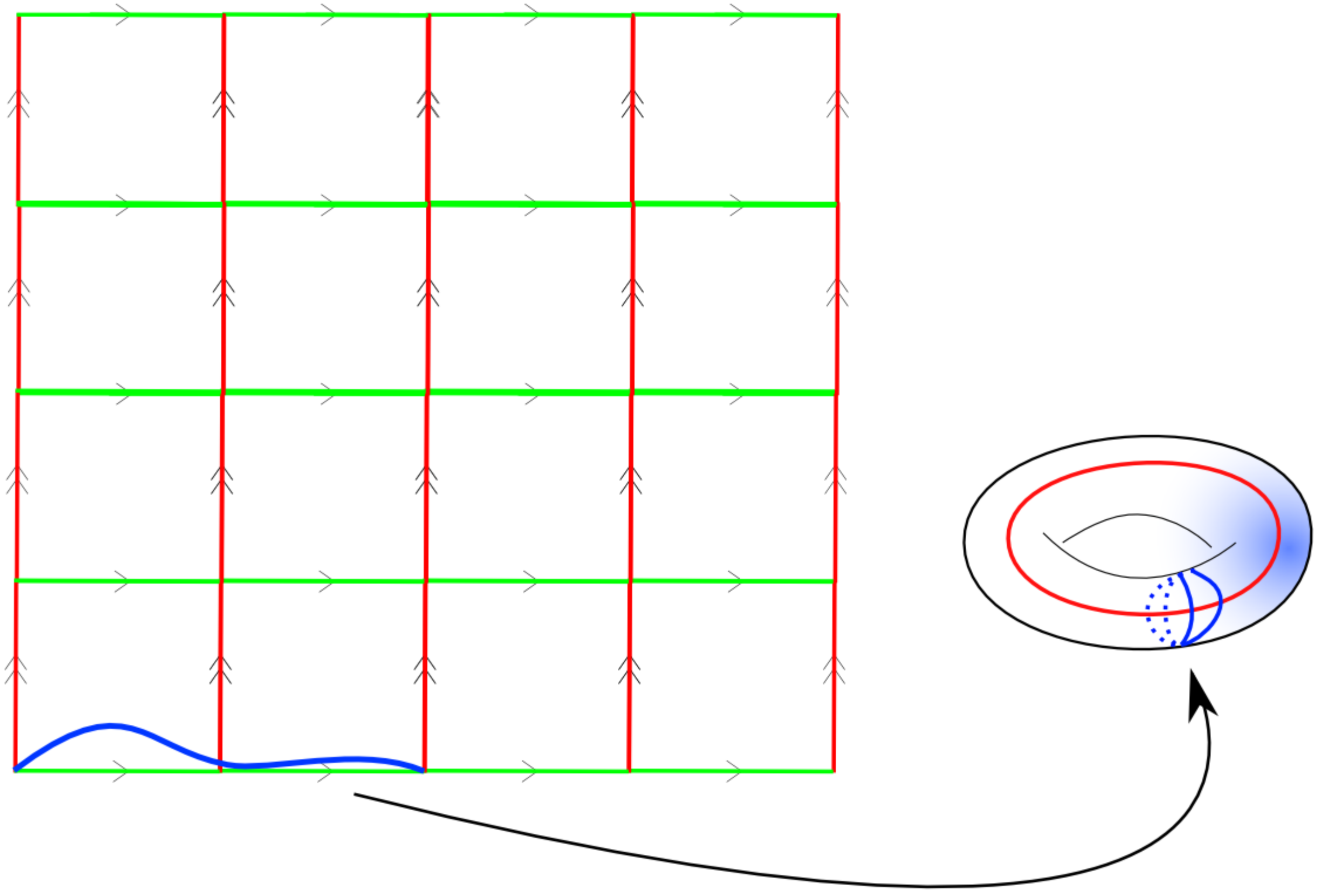
A curve in the original surface can be lifted to the universal covering space

A closed non-trivial loop can be lifted to an open path in the universal covering space

Curve lifting

A curve in the original surface can be lifted to the universal covering space

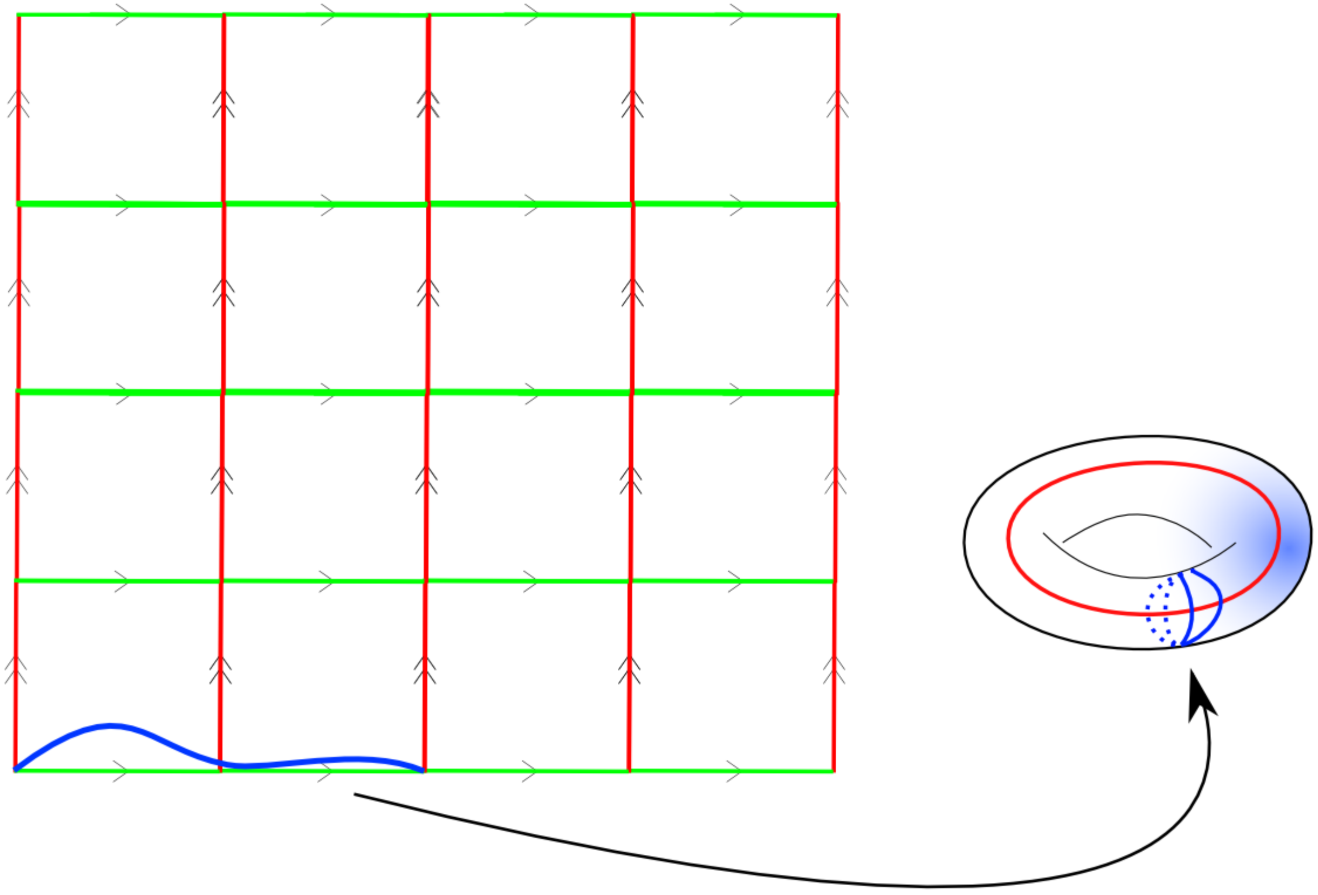
A closed non-trivial loop can be lifted to an open path in the universal covering space



Curve lifting

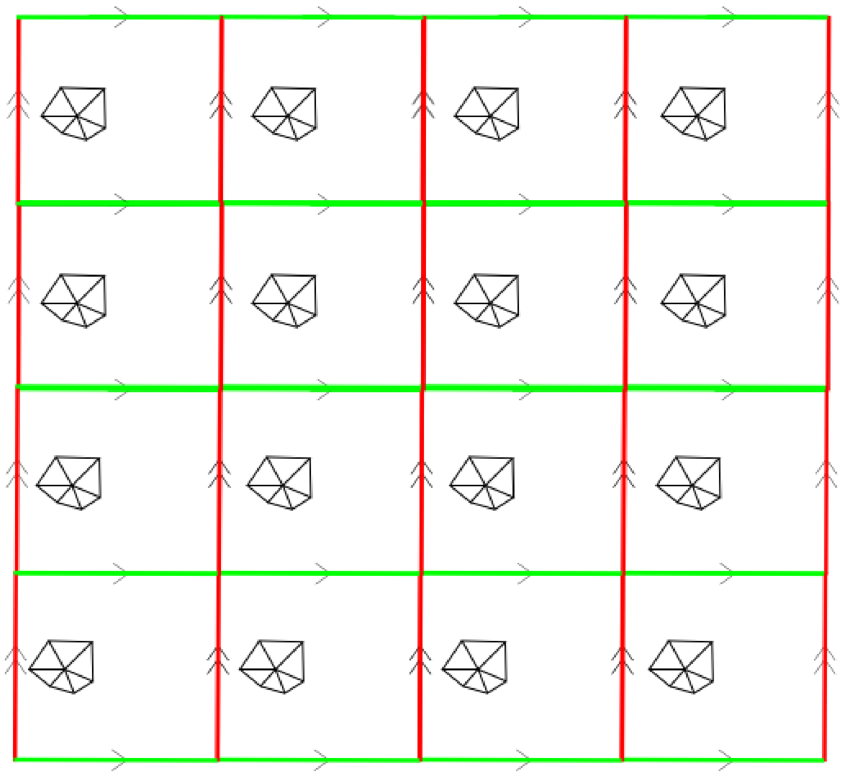
A curve in the original surface can be lifted to the universal covering space

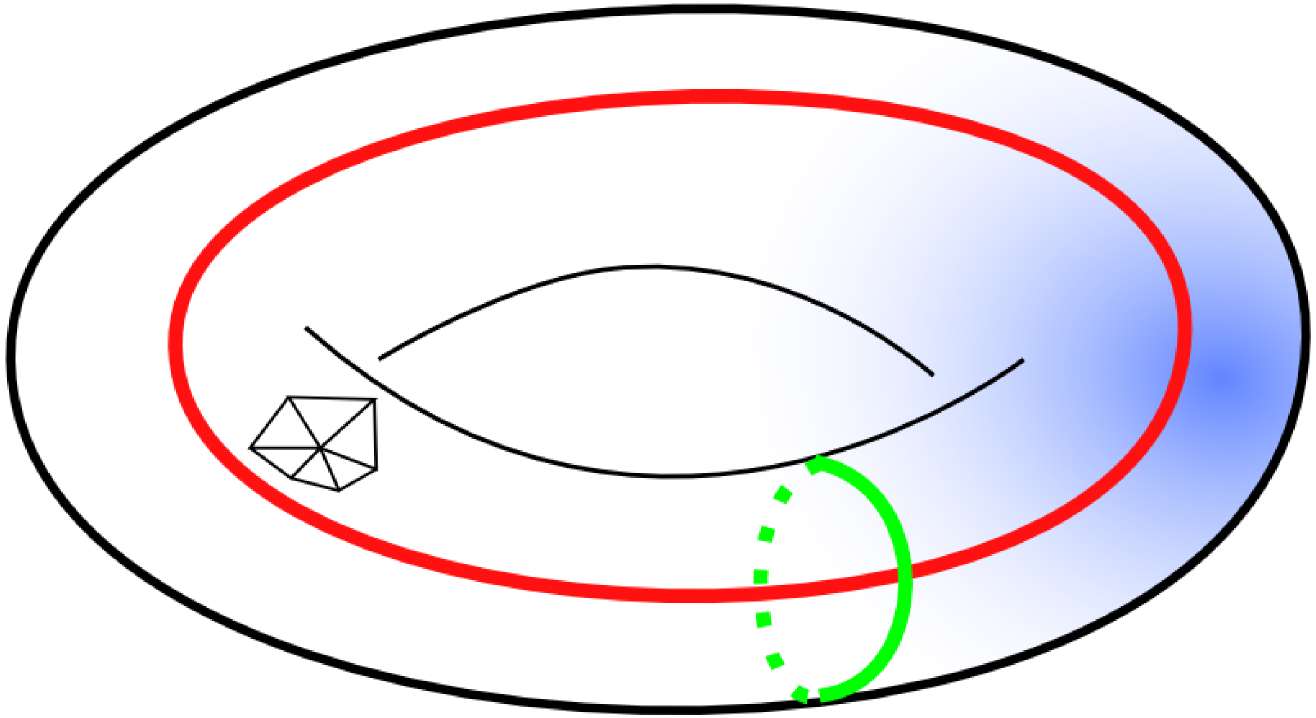
A closed non-trivial loop can be lifted to an open path in the universal covering space



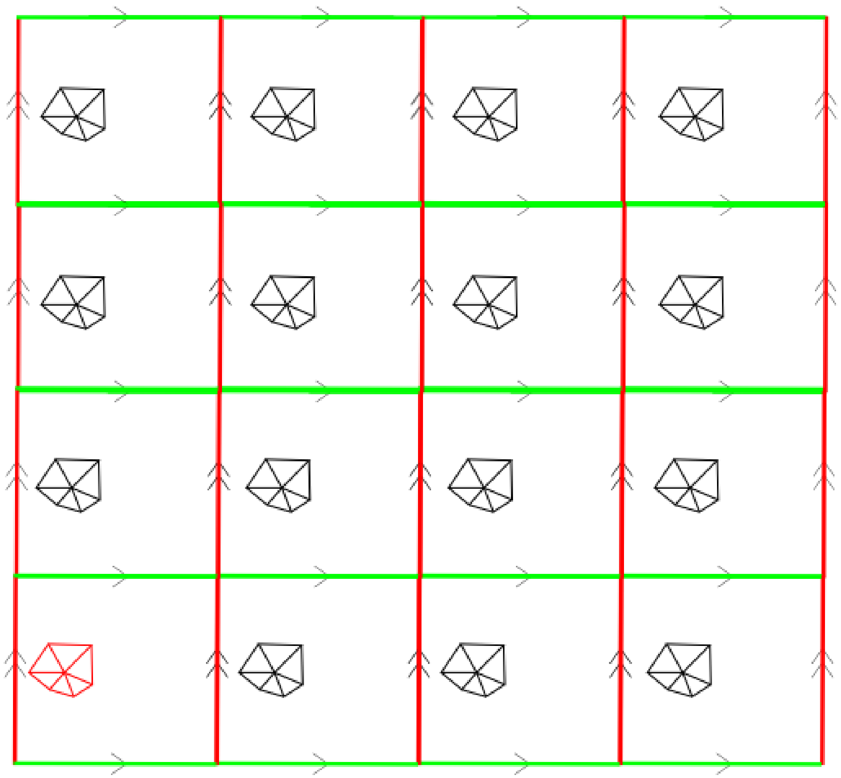
Key idea : many topological problems can be solved on the universal cover easier than on the original surface.

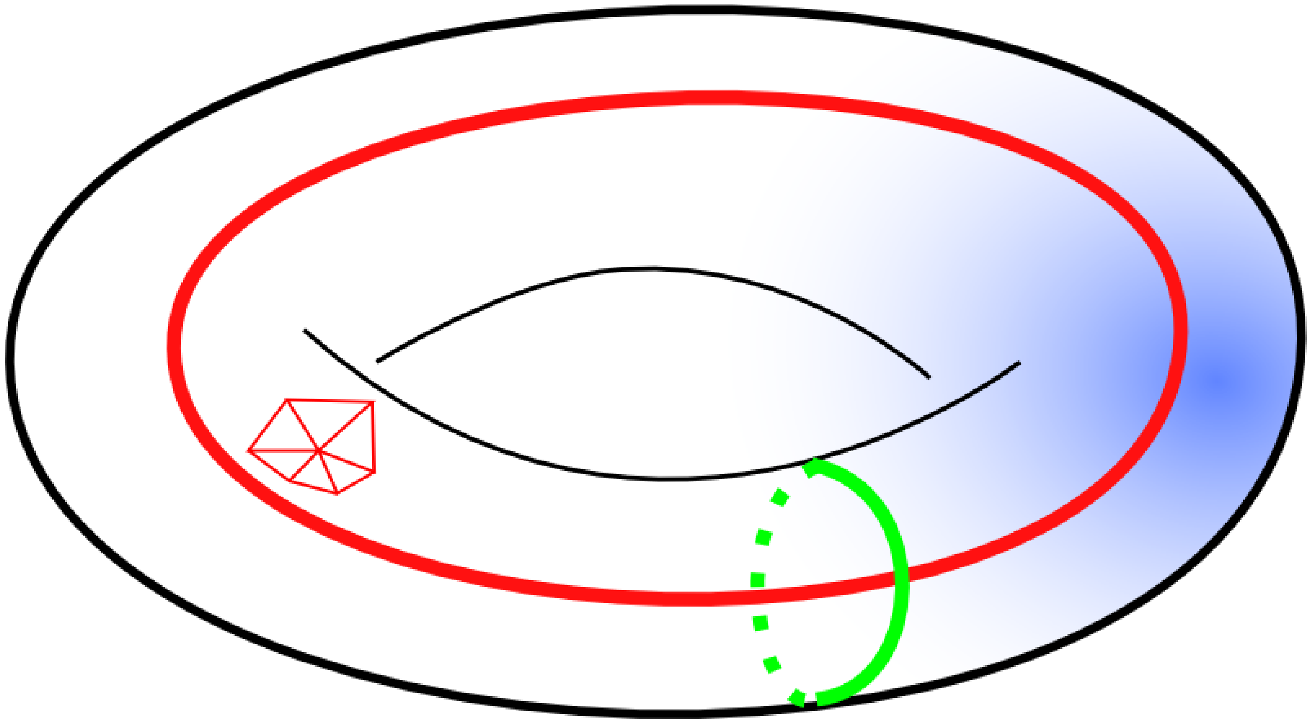
Curve lifting



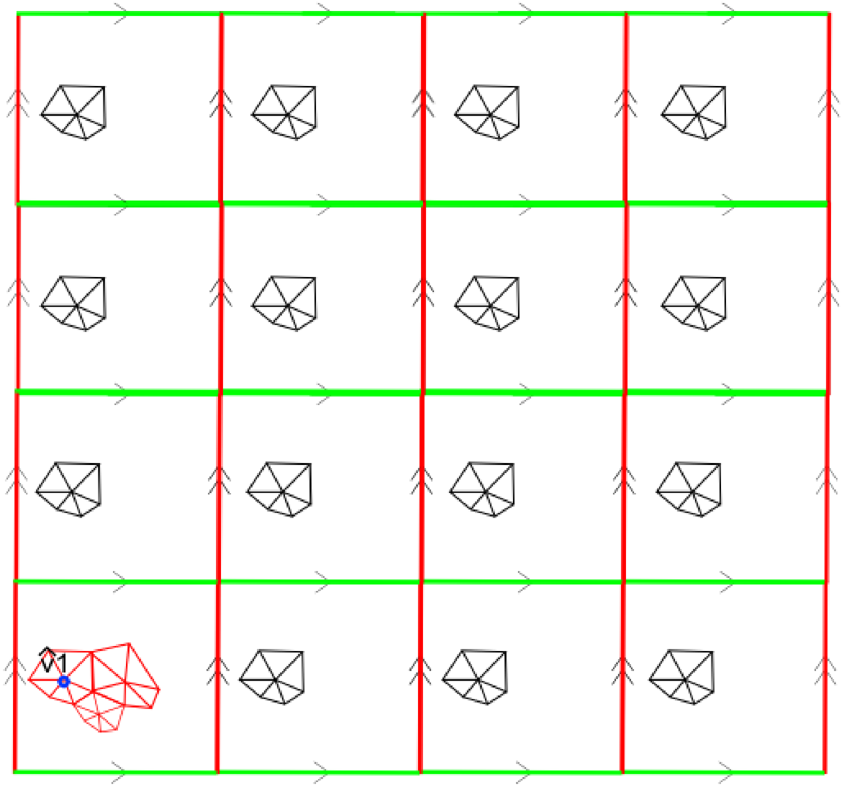


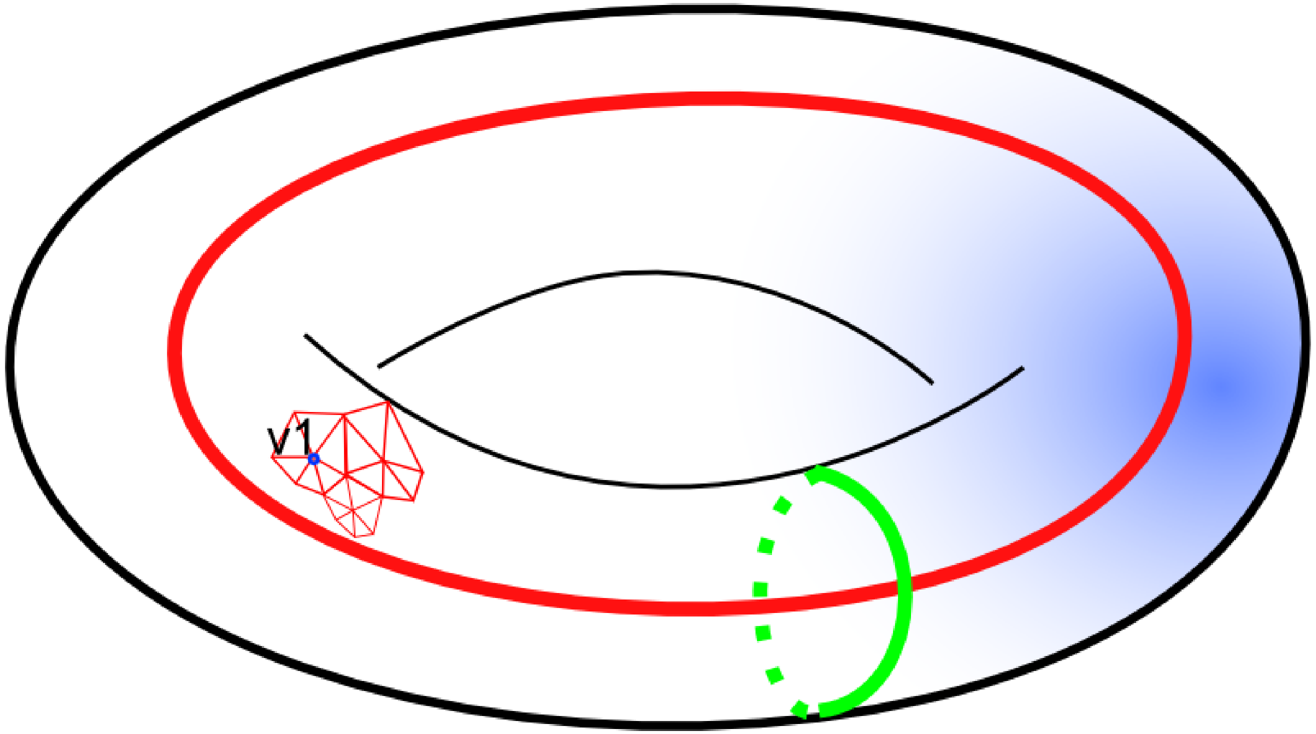
Curve lifting



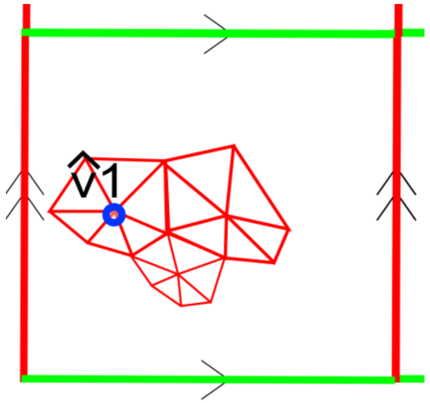


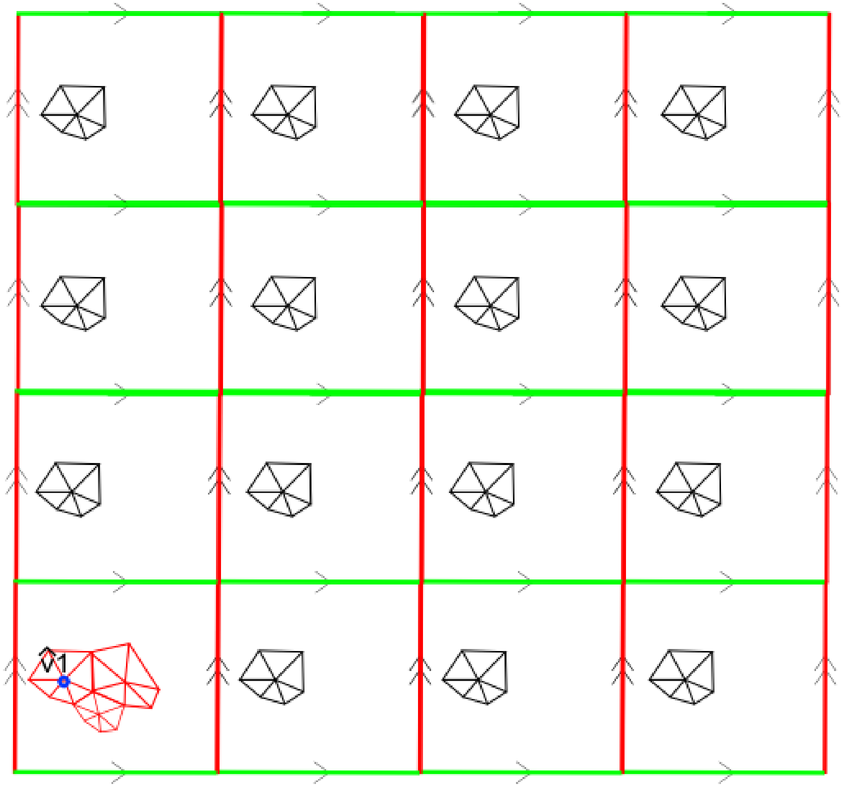
Curve lifting

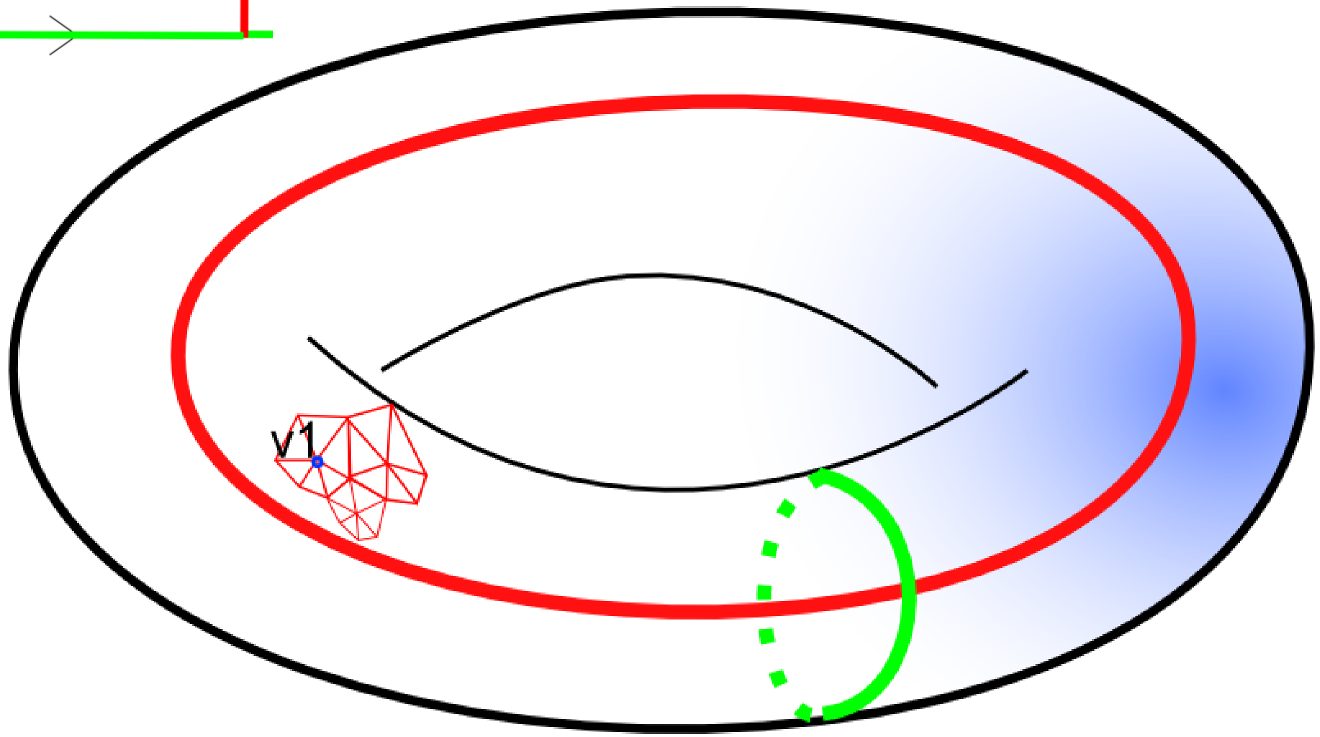




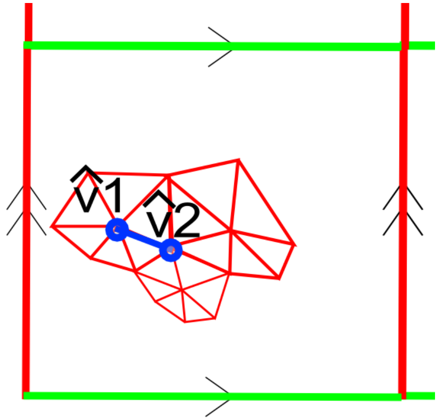
Curve lifting

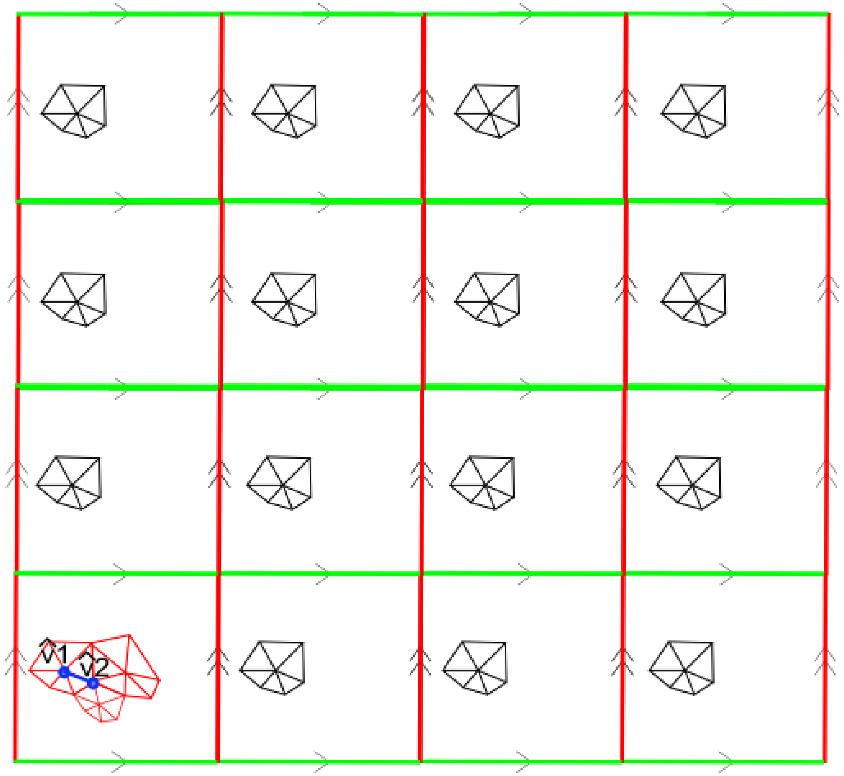


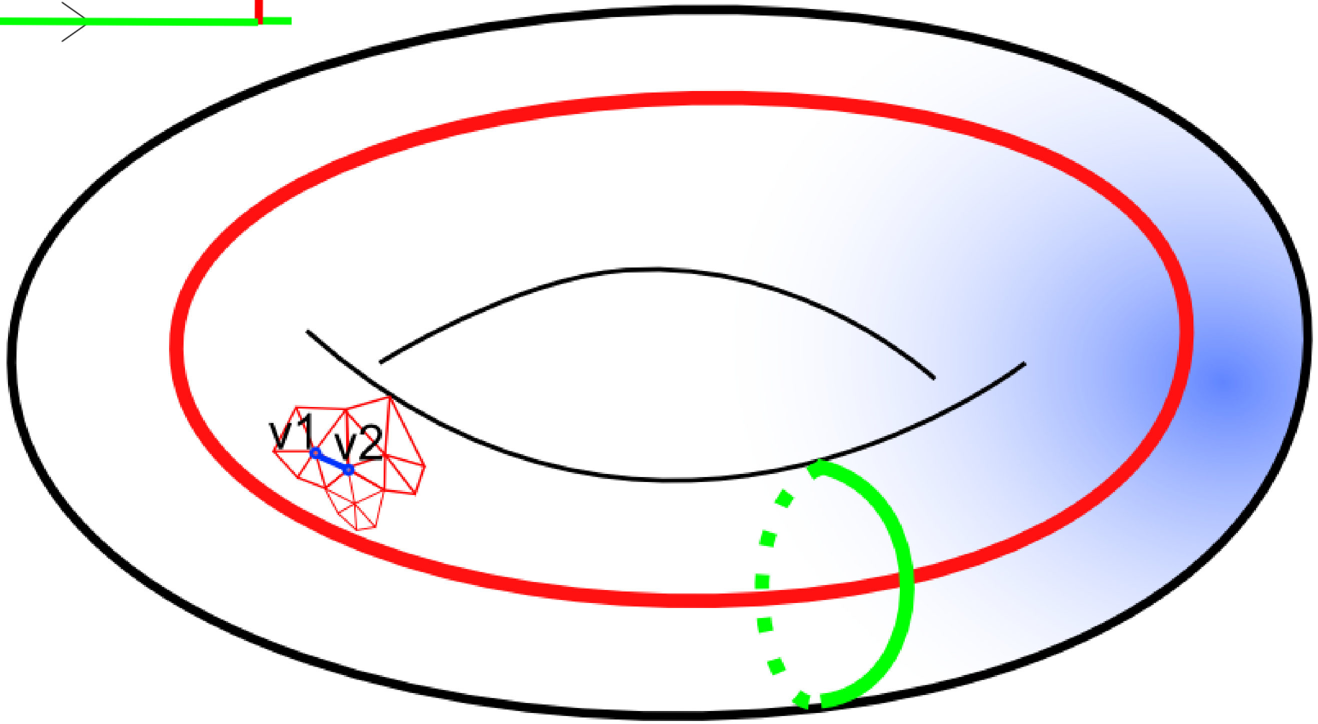




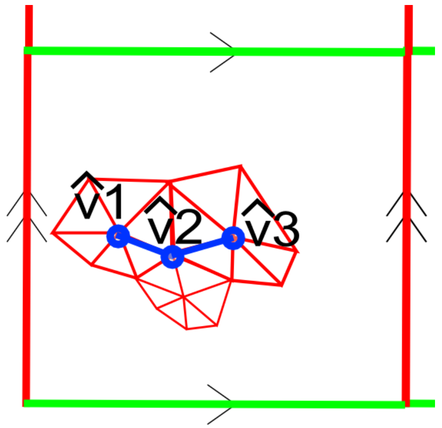
Curve lifting

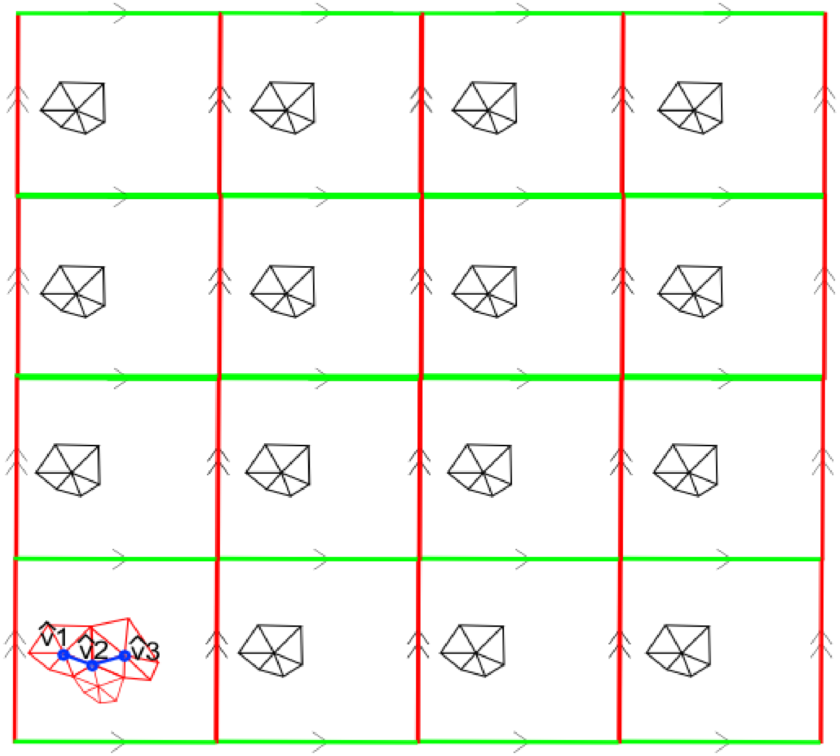


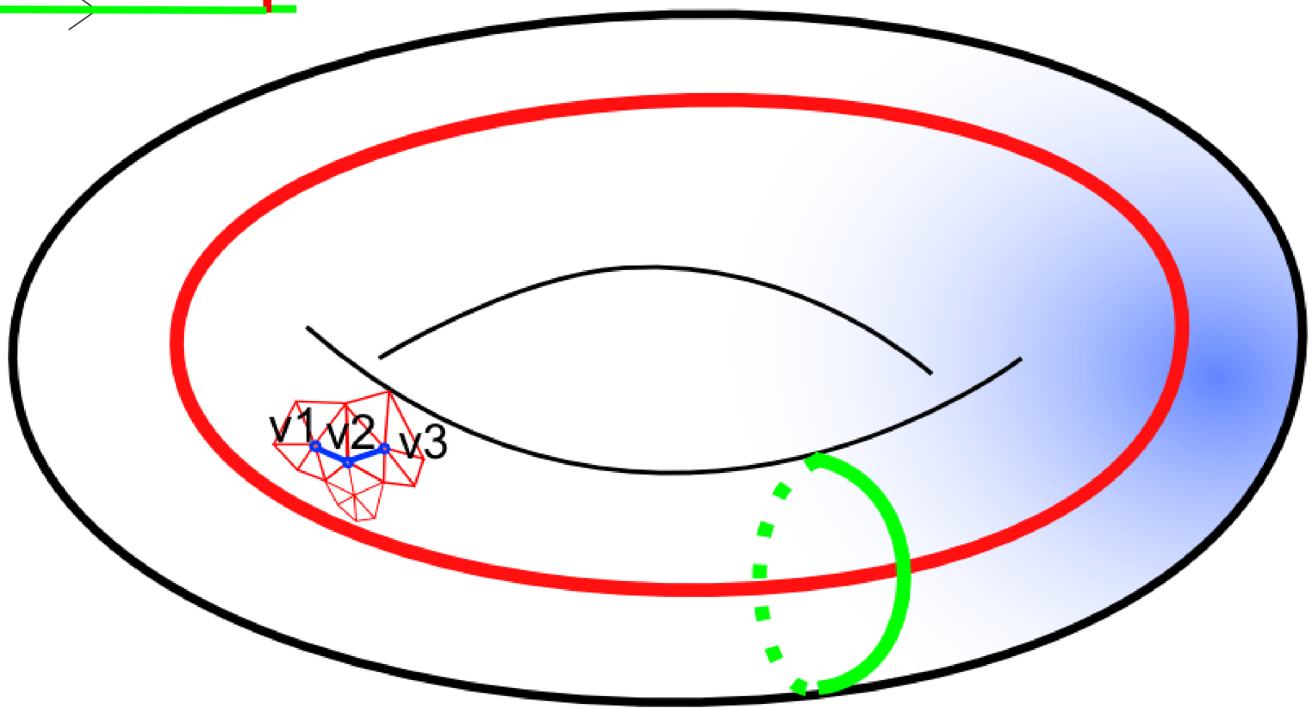




Curve lifting







Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

4-Denote the one ring neighborhoods of $\hat{v}_1$ and v_1 by $N(\hat{v}_1)$ and $N(v_1)$ respectively. The map π:$N(\hat{v}_1) \to N(v_1)$ is bijection.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

4-Denote the one ring neighborhoods of $\hat{v}_1$ and v_1 by $N(\hat{v}_1)$ and $N(v_1)$ respectively. The map π:$N(\hat{v}_1) \to N(v_1)$ is bijection.

5- Hence we can uniquely local the pre-image of v_2 in $N(\hat{v}_1)$, $\hat{v}_2$.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

4-Denote the one ring neighborhoods of $\hat{v}_1$ and v_1 by $N(\hat{v}_1)$ and $N(v_1)$ respectively. The map π:$N(\hat{v}_1) \to N(v_1)$ is bijection.

5- Hence we can uniquely local the pre-image of v_2 in $N(\hat{v}_1)$, $\hat{v}_2$.

6-Then we can uniquely local the pre-image of v_3 in $N(\hat{v}_2)$, $\hat{v}_3$.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

4-Denote the one ring neighborhoods of $\hat{v}_1$ and v_1 by $N(\hat{v}_1)$ and $N(v_1)$ respectively. The map π:$N(\hat{v}_1) \to N(v_1)$ is bijection.

5- Hence we can uniquely local the pre-image of v_2 in $N(\hat{v}_1)$, $\hat{v}_2$.

6-Then we can uniquely local the pre-image of v_3 in $N(\hat{v}_2)$, $\hat{v}_3$.

7-We continue this process step by step.

Curve lifting

1-Suppose that γ is a closed loop in M represented as a list of consecutive half-edges : $\{[v_1, v_2], ...[v_{n-1}, v_n], [v_n, v_1]\}$.

2-Denote by $\tilde{M}$ to the universal cover of M and by $\pi : \tilde{M} \to M$ is the projection. Locally this projection is one-to-one.

3-Locale a pre-image $\hat{v}_1 \in \tilde{M}$ such that $\pi(\hat{v}_1) = v_1$

4-Denote the one ring neighborhoods of $\hat{v}_1$ and v_1 by $N(\hat{v}_1)$ and $N(v_1)$ respectively. The map π:$N(\hat{v}_1) \to N(v_1)$ is bijection.

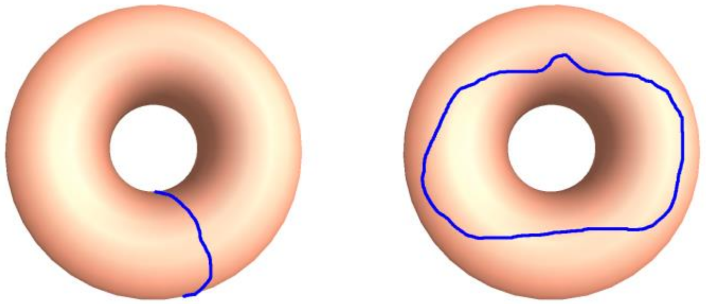
5- Hence we can uniquely local the pre-image of v_2 in $N(\hat{v}_1)$, $\hat{v}_2$.

6-Then we can uniquely local the pre-image of v_3 in $N(\hat{v}_2)$, $\hat{v}_3$.

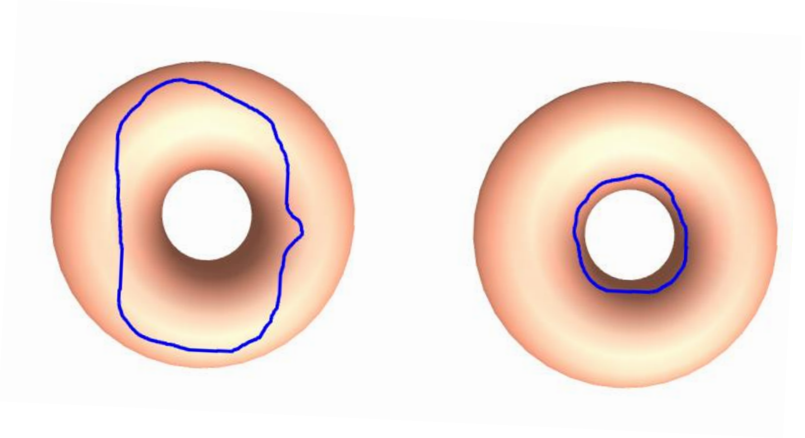
7-We continue this process step by step.

8-At the k-th step we can uniquely local the pre-image of v_k in $N(\hat{v}_k)$, $\hat{v}_k$, until we reach v_1 again.

Homotopy detection



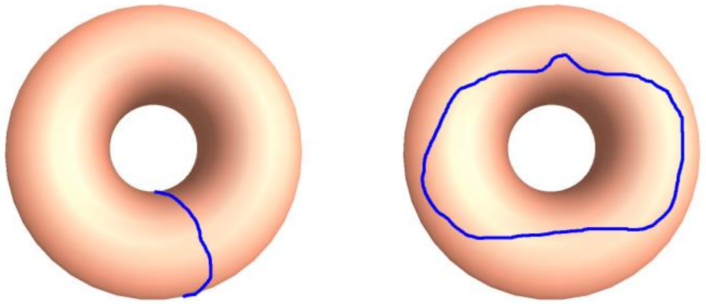
Non-homotopic curves



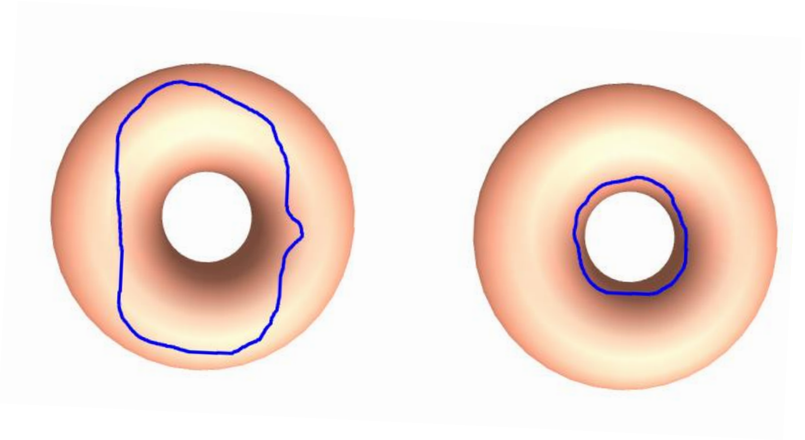
Homotopic curves

Image :David Gu

Homotopy detection



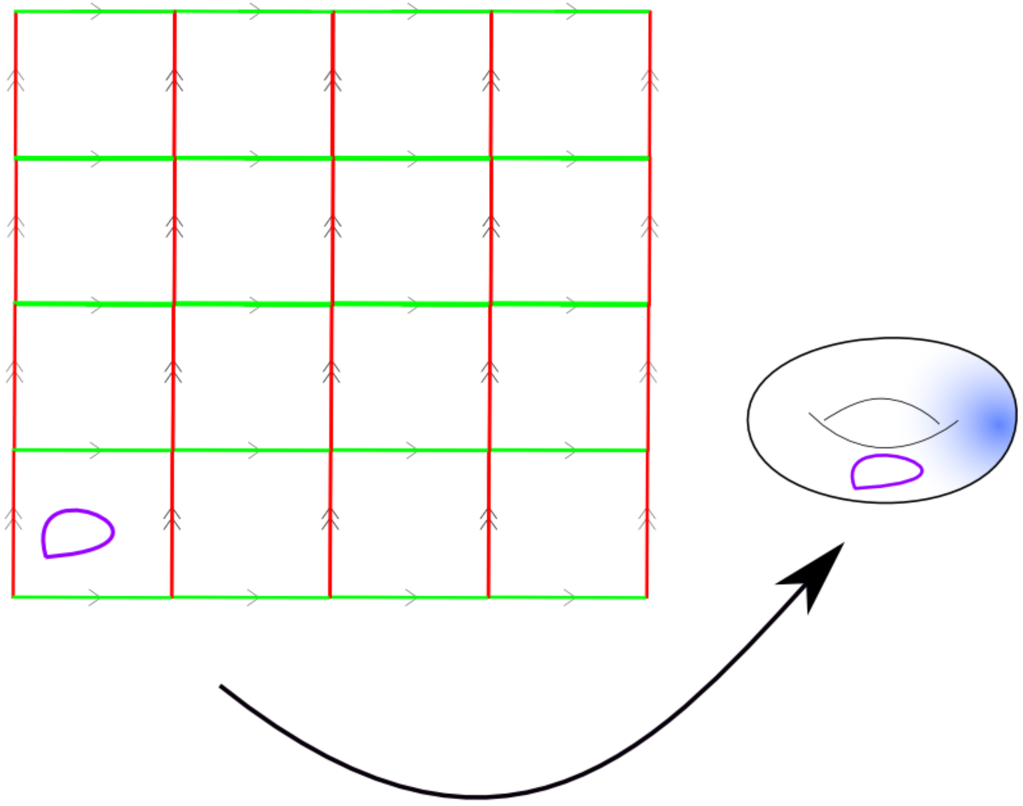
Non-homotopic curves



Homotopic curves

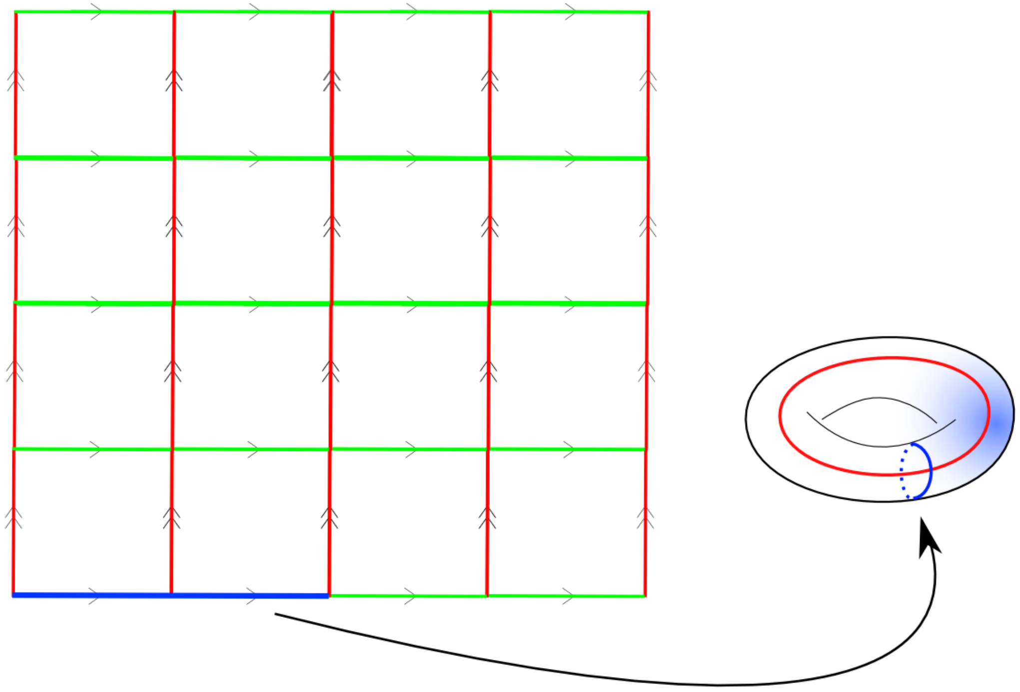
Image :David Gu

Homotopy detection



Homotopically trivial loops are lifted to closed loops in the covering space.

Homotopy detection



Homotopically non-trivial loops are lifted to open curves in the covering space.

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

3-use the Dijkstra's algorithm to compute a shortest path connecting v_1 and w_1 :$\gamma = \{v_1, d_1, ..., d_k, w_1\}$

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

3-use the Dijkstra's algorithm to compute a shortest path connecting v_1 and w_1 :$\gamma = \{v_1, d_1, ..., d_k, w_1\}$
4-Construct $\Gamma = \gamma_1 \circ \gamma \circ \gamma_2^{-1} \circ \gamma^{-1}$

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

3-use the Dijkstra's algorithm to compute a shortest path connecting v_1 and w_1 :$\gamma = \{v_1, d_1, ..., d_k, w_1\}$
4-Construct $\Gamma = \gamma_1 \circ \gamma \circ \gamma_2^{-1} \circ \gamma^{-1}$

$$\Gamma = \{v_1, ..., v_n, v_1, d_1, ..., d_k, w_1, w_m, ..., w_1, d_k, ..., d_1, w_1\}$$

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

3-use the Dijkstra's algorithm to compute a shortest path connecting v_1 and w_1 : $\gamma = \{v_1, d_1, ..., d_k, w_1\}$
4-Construct $\Gamma = \gamma_1 \circ \gamma \circ \gamma_2^{-1} \circ \gamma^{-1}$

$$\Gamma = \{v_1, ..., v_n, v_1, d_1, ..., d_k, w_1, w_m, ..., w_1, d_k, ..., d_1, w_1\}$$

5-Construct a finite portion of the universal cover.

Homotopy detection

Input : A mesh M, two chains γ_1 and γ_2
Output: Whether γ_1 is homotopic to γ_2
1- represent γ_1 by a circular list of vertices

$$\{v_1, ..., v_n\}$$

2- represent γ_2 by a circular list of vertices

$$\{w_1, ..., w_m\}$$

3-use the Dijkstra's algorithm to compute a shortest path connecting v_1 and w_1 :$\gamma = \{v_1, d_1, ..., d_k, w_1\}$
4-Construct $\Gamma = \gamma_1 \circ \gamma \circ \gamma_2^{-1} \circ \gamma^{-1}$

$$\Gamma = \{v_1, ..., v_n, v_1, d_1, ..., d_k, w_1, w_m, ..., w_1, d_k, ..., d_1, w_1\}$$

5-Construct a finite portion of the universal cover.
6-Left Γ to the universal cover and obtain the curve $\bar{\Gamma}$. If $\bar{\Gamma}$ is a loop then γ_1 is homotopic to γ_2. Otherwise, they are not homotopic.

References

Algorithms presented here can be found in :

D. Gu and S Yau, Computational conformal geometry. Somerville, Mass, USA: International Press, 2008.