Computing Shortest Non-Trivial Cycles on Orientable Surfaces of Bounded Genus in Almost Linear Time

Martin Kutz

Max-Planck Institut für Informatik
Saarbrücken, Germany

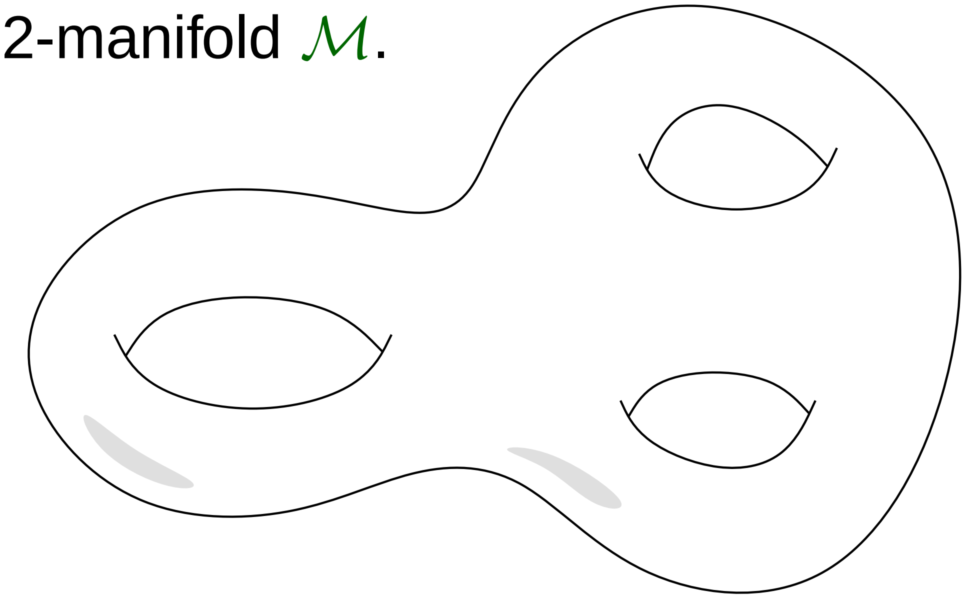
Computing Shortest Non-Trivial Cycles on Orientable Surfaces of Bounded Genus in Almost Linear Time

Martin Kutz

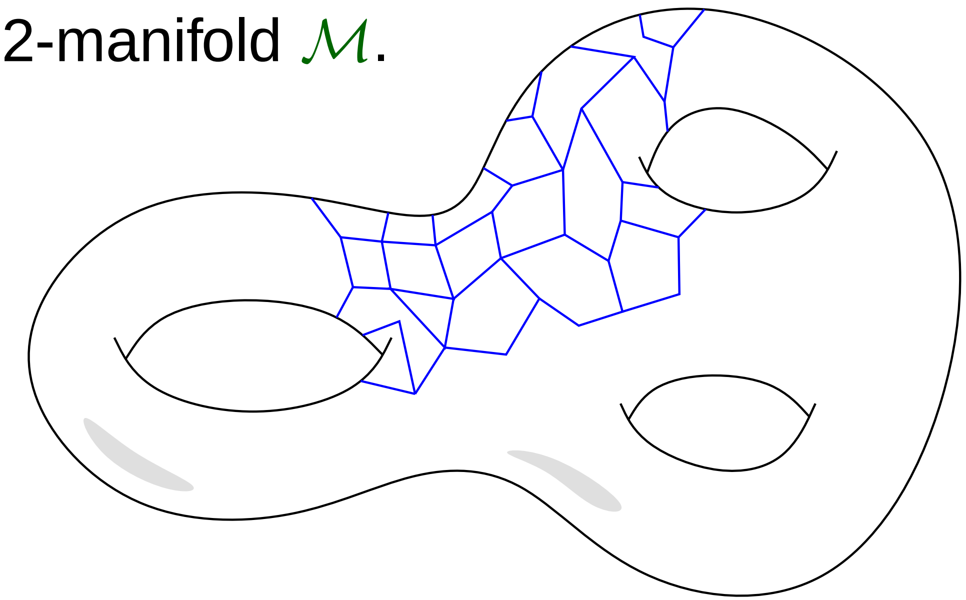
Max-Planck Institut für Informatik
Saarbrücken, Germany

Combinatorial Surfaces

A *surface*: connected, orientable 2-manifold $\mathcal{M}$.



Combinatorial Surfaces



A *surface*: connected, orientable 2-manifold $\mathcal{M}$.

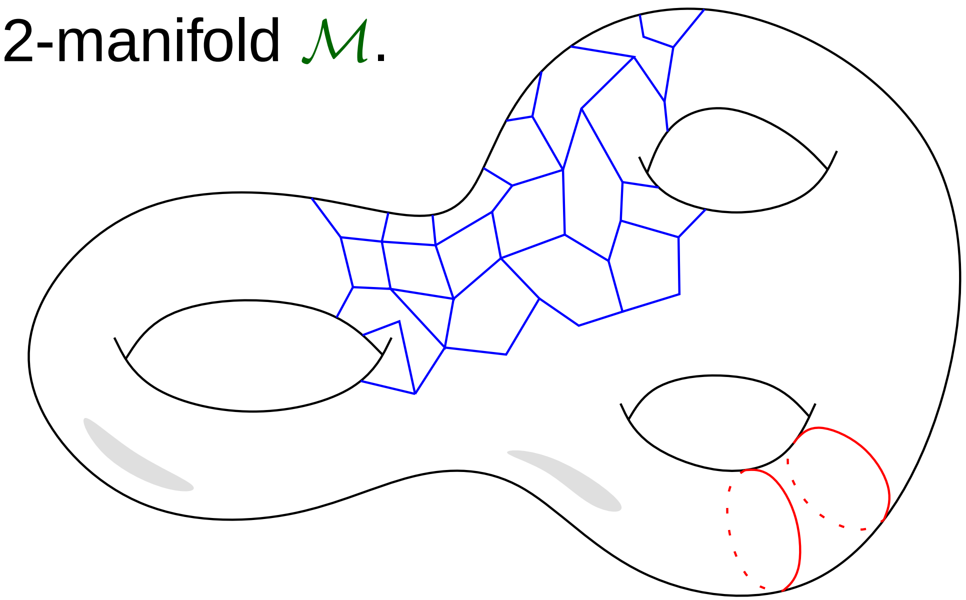
Combinatorial representation:

a graph with local encoding
of embedding

(e.g., edge-face incidences,
or cyclic ordering of edges around each vertex)

Combinatorial Surfaces

A *surface*: connected, orientable 2-manifold $\mathcal{M}$.



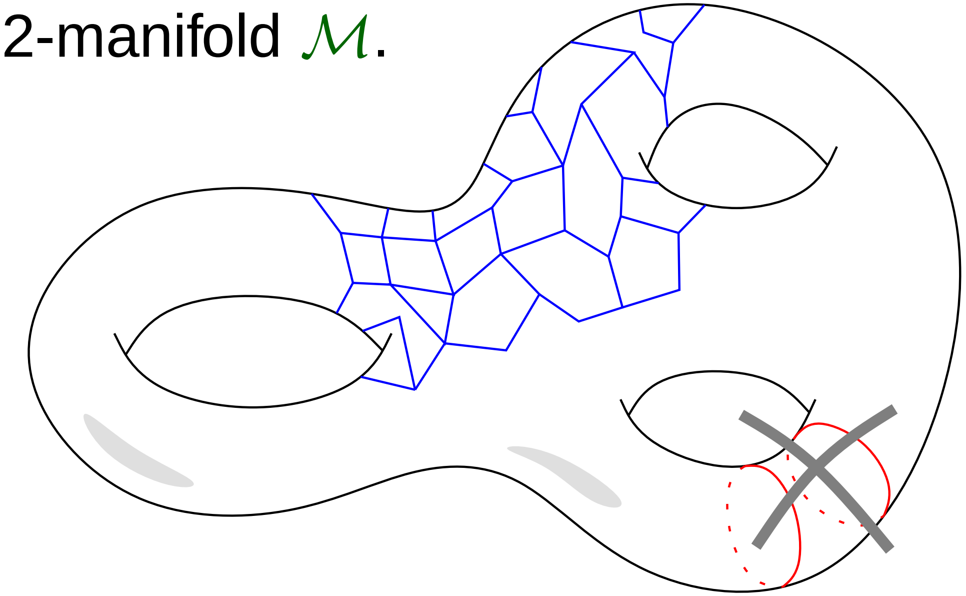
Combinatorial representation:

a graph with local encoding of embedding

(e.g., edge-face incidences,
or cyclic ordering of edges around each vertex)

Combinatorial Surfaces

A *surface*: connected, orientable 2-manifold $\mathcal{M}$.



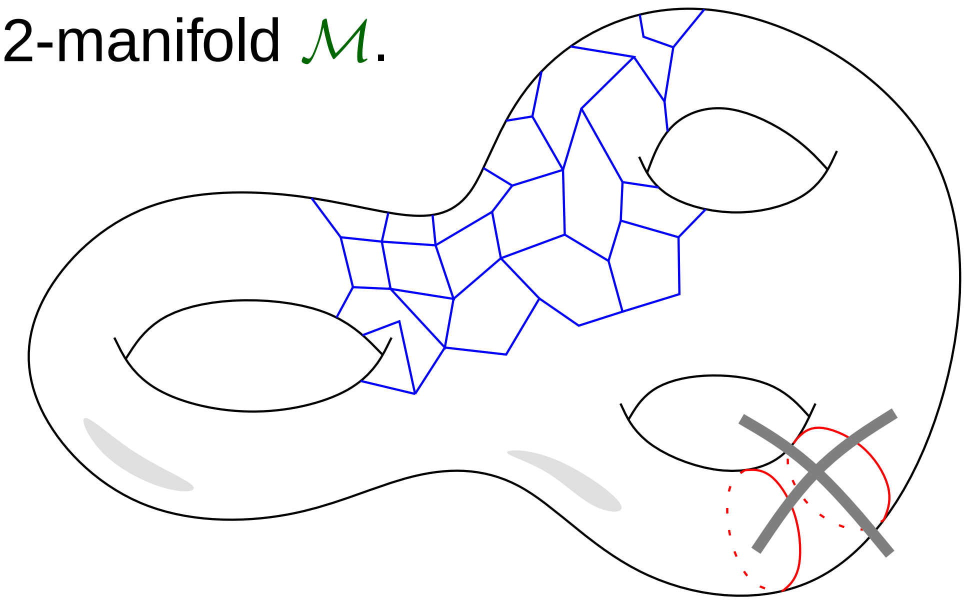
Combinatorial representation:

a graph with local encoding of embedding

(e.g., edge-face incidences, or cyclic ordering of edges around each vertex)

Combinatorial Surfaces

A *surface*: connected, orientable 2-manifold $\mathcal{M}$.



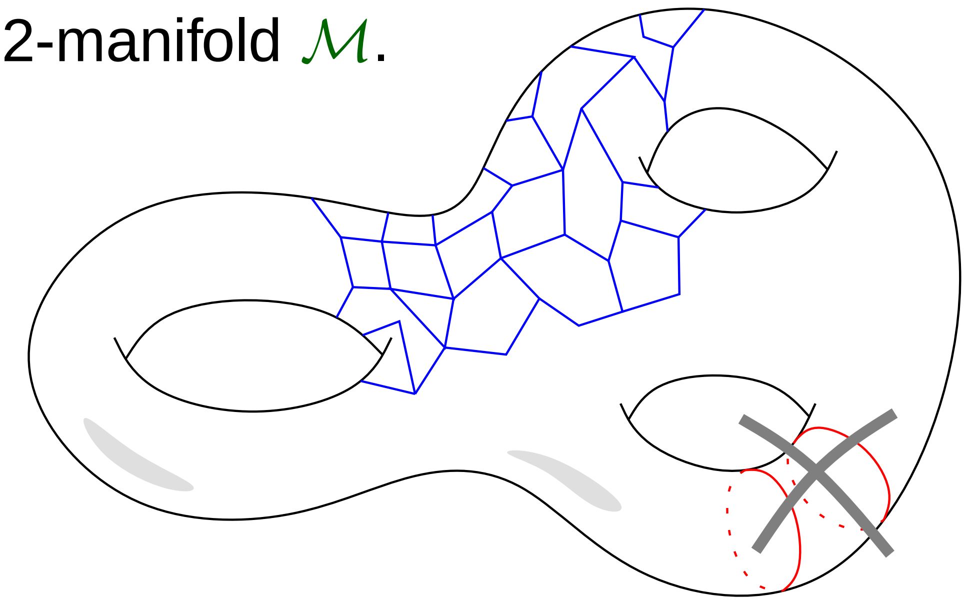
Combinatorial representation:

a graph with local encoding
of embedding

(e.g., edge-face incidences,
or cyclic ordering of edges around each vertex)

Only combinatorial information — *no geometry*.

Combinatorial Surfaces



A *surface*: connected, orientable 2-manifold $\mathcal{M}$.

Combinatorial representation:

a graph with local encoding of embedding

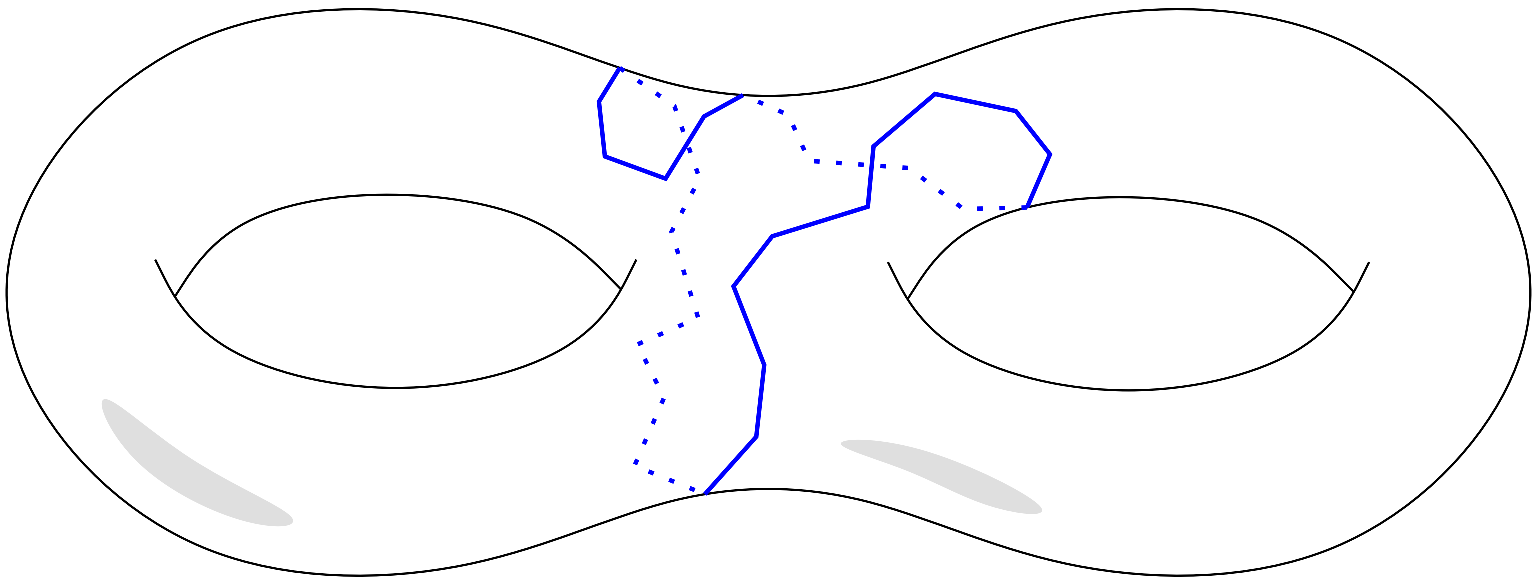
(e.g., edge-face incidences, or cyclic ordering of edges around each vertex)

Only combinatorial information — *no geometry*.

Edges may have weights.

Non-Trivial Cycles

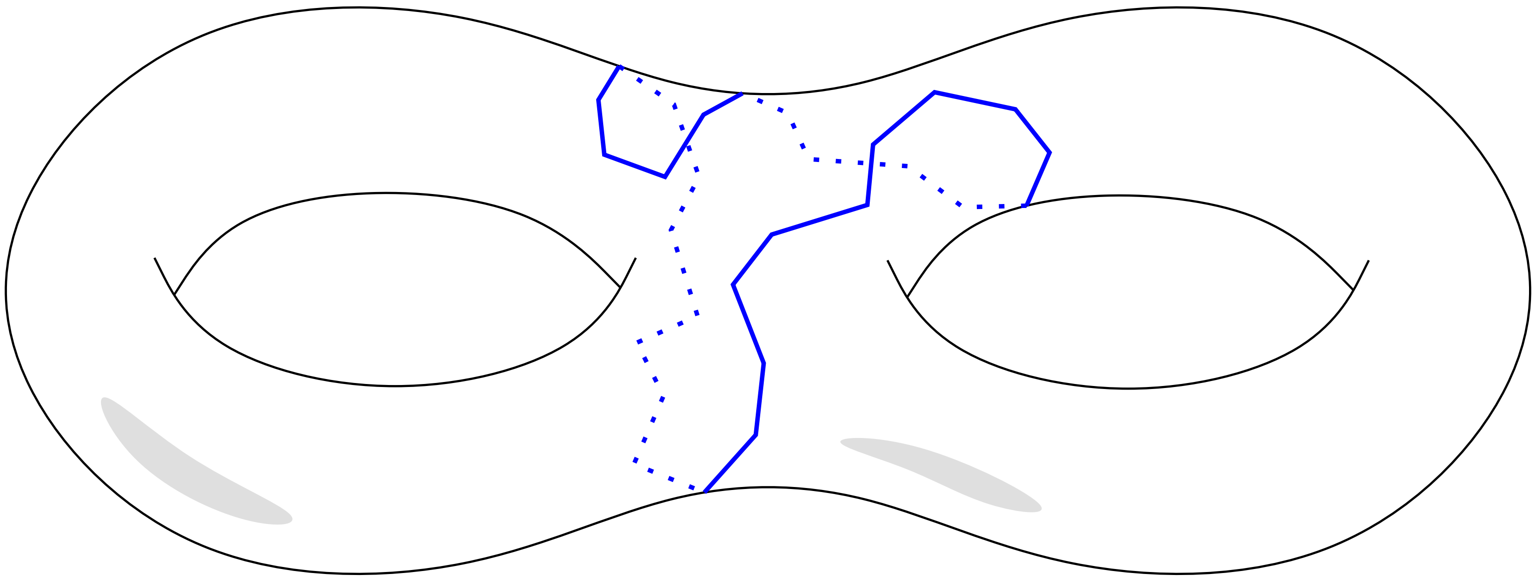
A *cycle* on a surface $\mathcal{M}$ is a closed walk on the defining graph.



Non-Trivial Cycles

A *cycle* on a surface $\mathcal{M}$ is a closed walk on the defining graph.

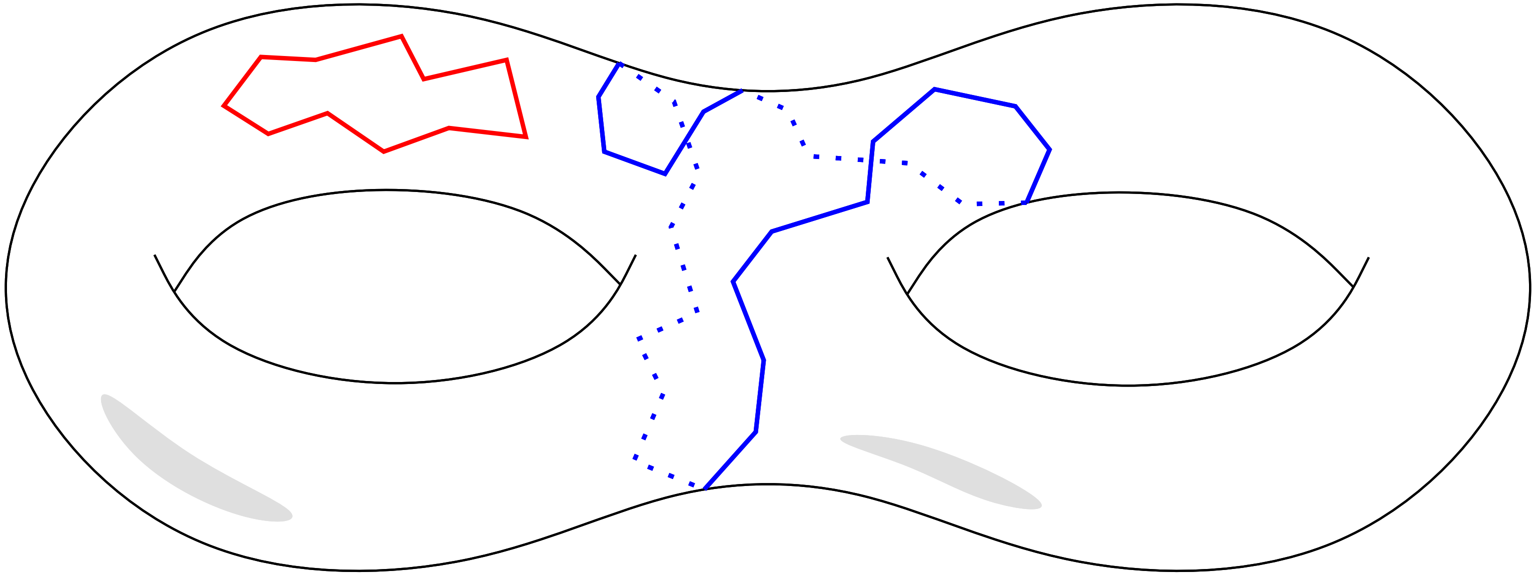
Want to compute short cycles that are topologically interesting.



Non-Trivial Cycles

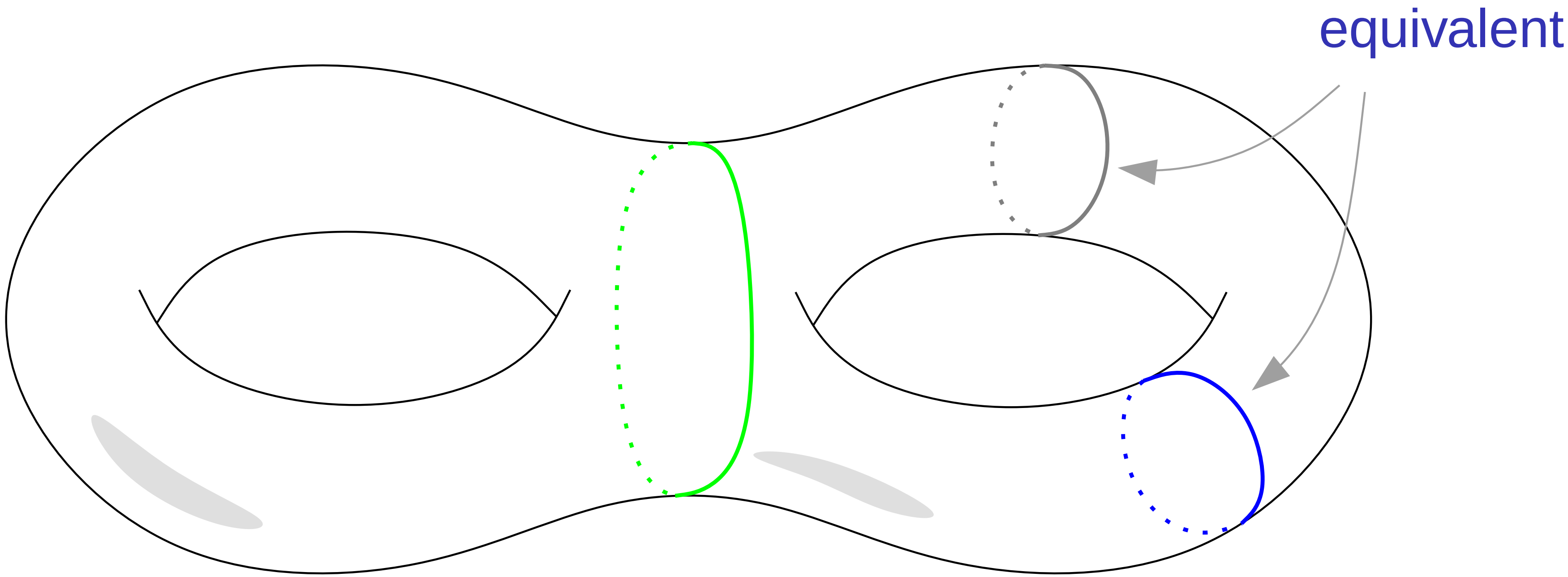
A *cycle* on a surface $\mathcal{M}$ is a closed walk on the defining graph.

Want to compute short cycles that are topologically interesting.



Non-Trivial Cycles

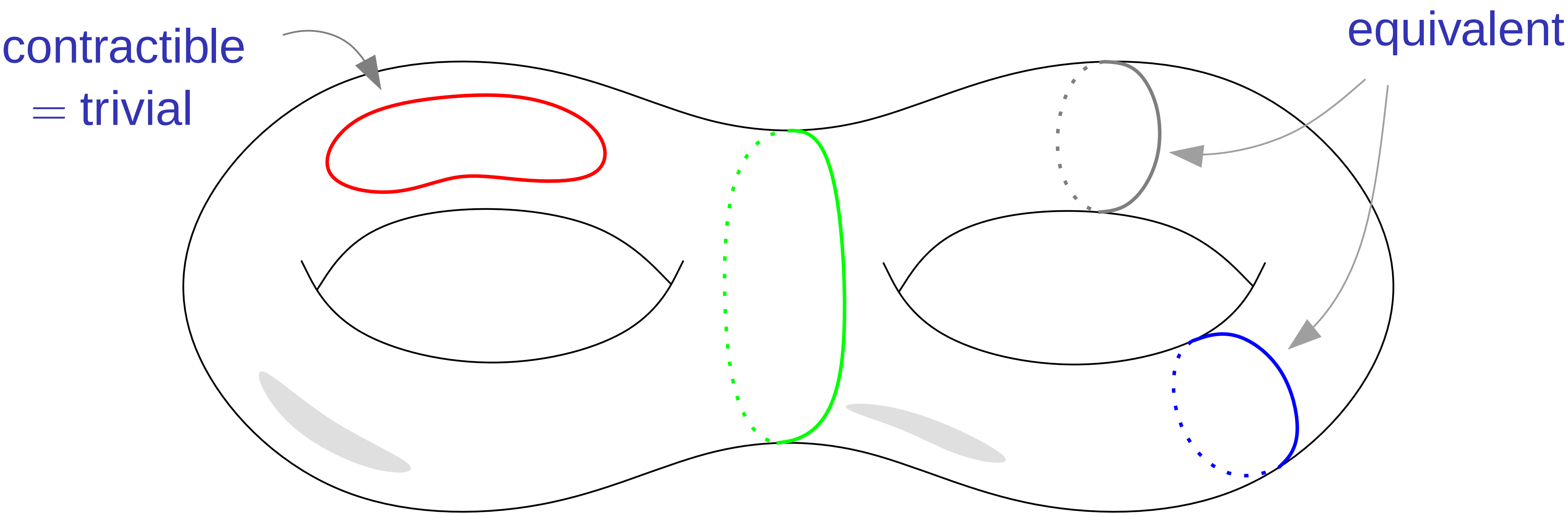
Two cycles on a surface $\mathcal{M}$ are *equivalent*
if one can be continuously transformed into the other.



Non-Trivial Cycles

Two cycles on a surface $\mathcal{M}$ are *equivalent*
if one can be continuously transformed into the other.

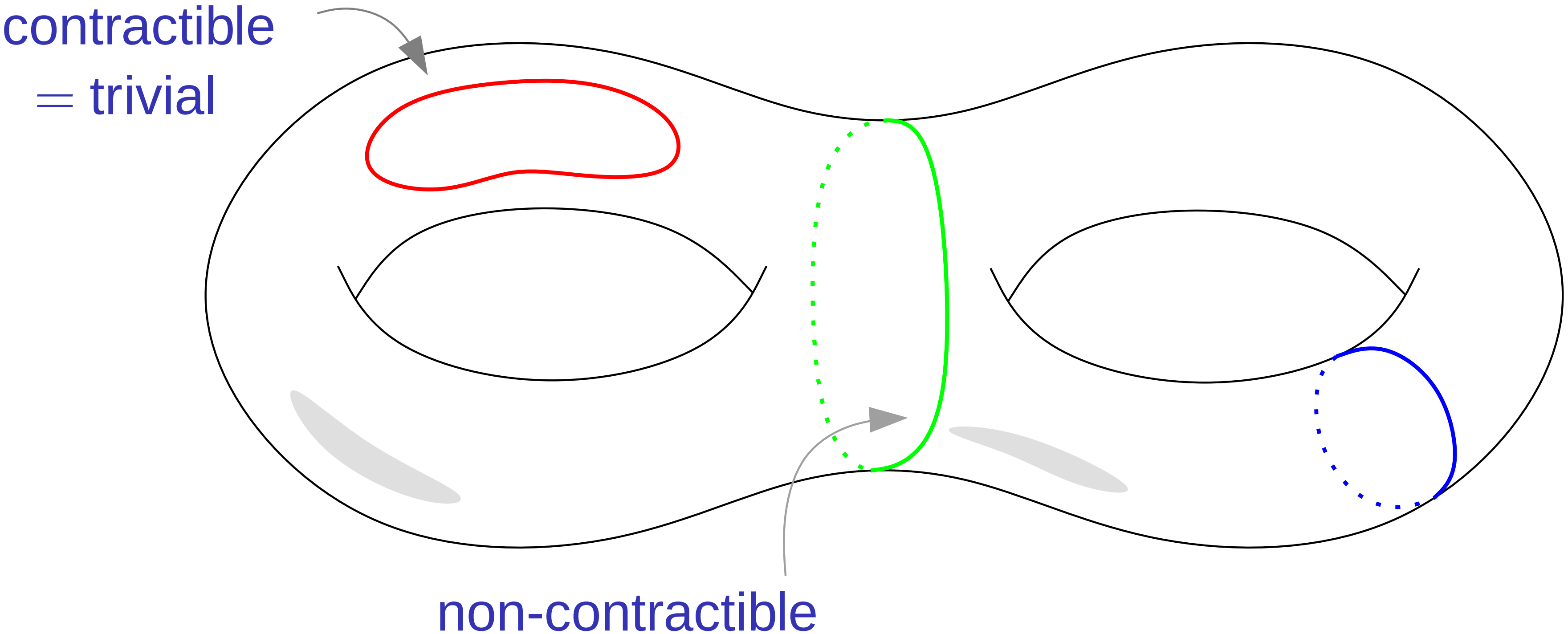
Cycle γ *contractible* (*trivial*): γ equivalent to a point.



Non-Trivial Cycles

Two cycles on a surface $\mathcal{M}$ are *equivalent* if one can be continuously transformed into the other.

Cycle γ *contractible* (*trivial*): γ equivalent to a point.

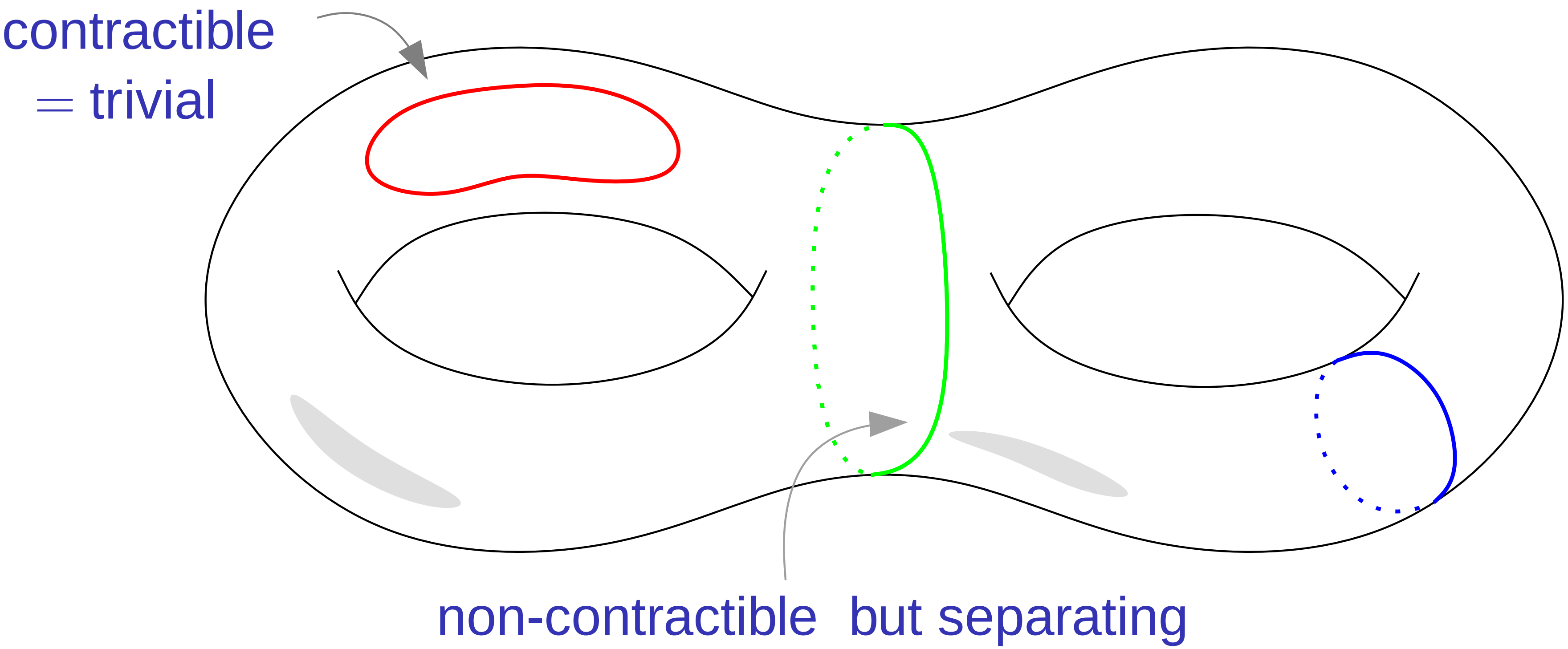


Non-Trivial Cycles

Two cycles on a surface $\mathcal{M}$ are *equivalent*
if one can be continuously transformed into the other.

Cycle γ *contractible* (*trivial*): γ equivalent to a point.

Cycle γ *separating*: cut surface $\mathcal{M}$ ✂ γ disconnected.

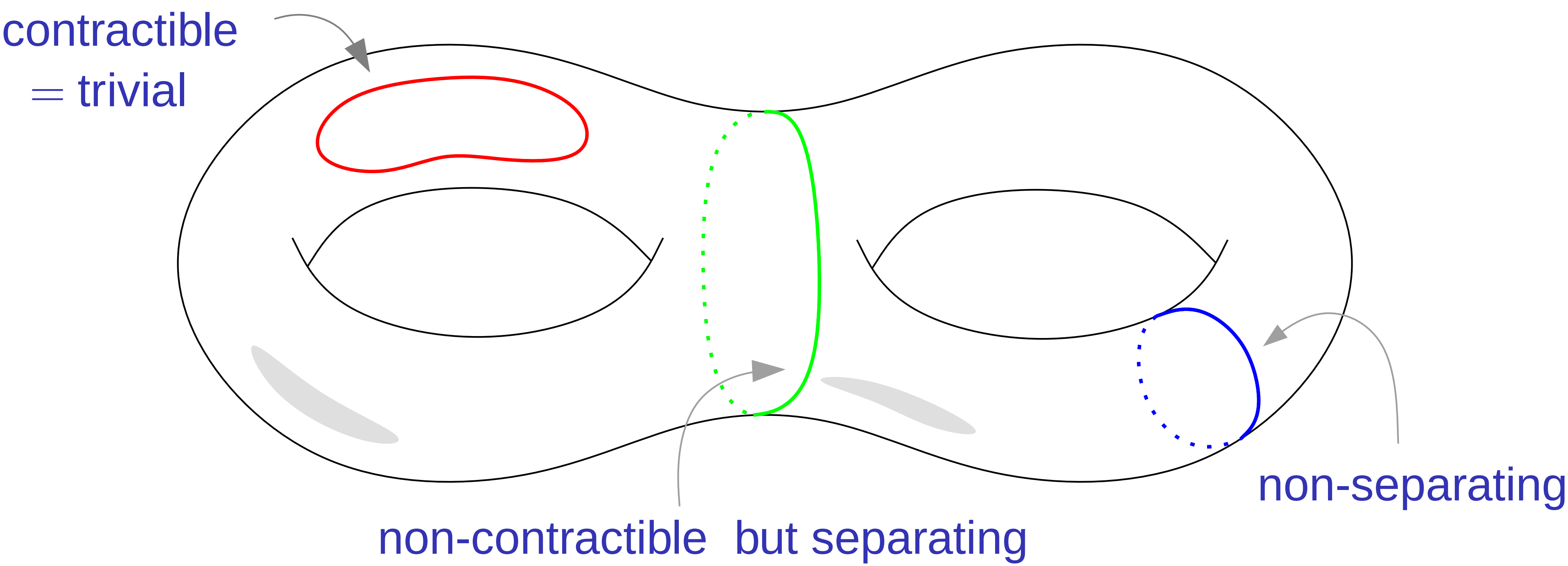


Non-Trivial Cycles

Two cycles on a surface $\mathcal{M}$ are *equivalent*
if one can be continuously transformed into the other.

Cycle γ *contractible* (*trivial*): γ equivalent to a point.

Cycle γ *separating*: cut surface $\mathcal{M}$ ✂ γ disconnected.



Result

Theorem. **[K, SoCG 2006]**

On orientable surfaces of bounded genus,
shortest non-contractible and shortest non-separating cycles
can be computed in $O(n \log n)$ time.

Result

Theorem. **[K, SoCG 2006]**

On orientable surfaces of bounded genus,
shortest non-contractible and shortest non-separating cycles
can be computed in $O(n \log n)$ time.

Why short cycles?

Why Short Cycles?

- Cutting along non-trivial cycles makes surface topologically simpler
- want to do this with *short* cycles.

Why Short Cycles?

- Cutting along non-trivial cycles makes surface topologically simpler
- want to do this with *short* cycles.
- Short non-trivial cycles are primitives in other algorithms:
  - cutting a surface into a disk

    [Erickson & Har-Peled, SoCG 2002]
  - tightening paths and cycles

    [Colin de Verdiére & Erickson, SODA 2006]

Why Short Cycles?

- Cutting along non-trivial cycles makes surface topologically simpler
- want to do this with *short* cycles.

- Short non-trivial cycles are primitives in other algorithms:
  - cutting a surface into a disk

    [Erickson & Har-Peled, SoCG 2002]
  - tightening paths and cycles

    [Colin de Verdiére & Erickson, SODA 2006]
  - shortest splitting cycles

    [Chambers et al., SoCG 2006]

Why Short Cycles?

- Cutting along non-trivial cycles makes surface topologically simpler
- want to do this with *short* cycles.

- Short non-trivial cycles are primitives in other algorithms:
  - cutting a surface into a disk

    [Erickson & Har-Peled, SoCG 2002]
  - tightening paths and cycles

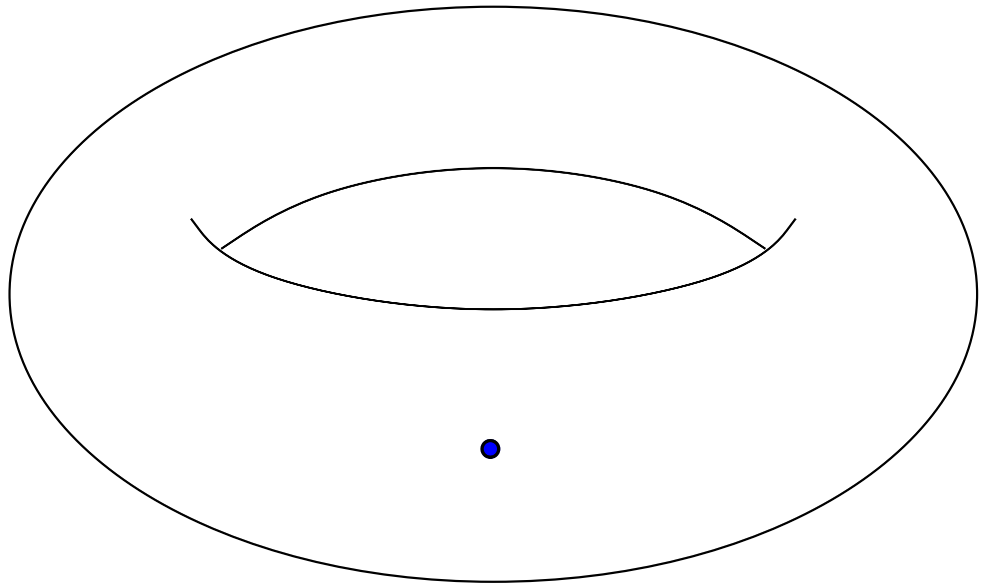
    [Colin de Verdiére & Erickson, SODA 2006]
  - shortest splitting cycles

    [Chambers et al., SoCG 2006]

- Computing short non-trivial cycles turned out to be a core problem in computational topology.

Shortest Cycles Through a Basepoint

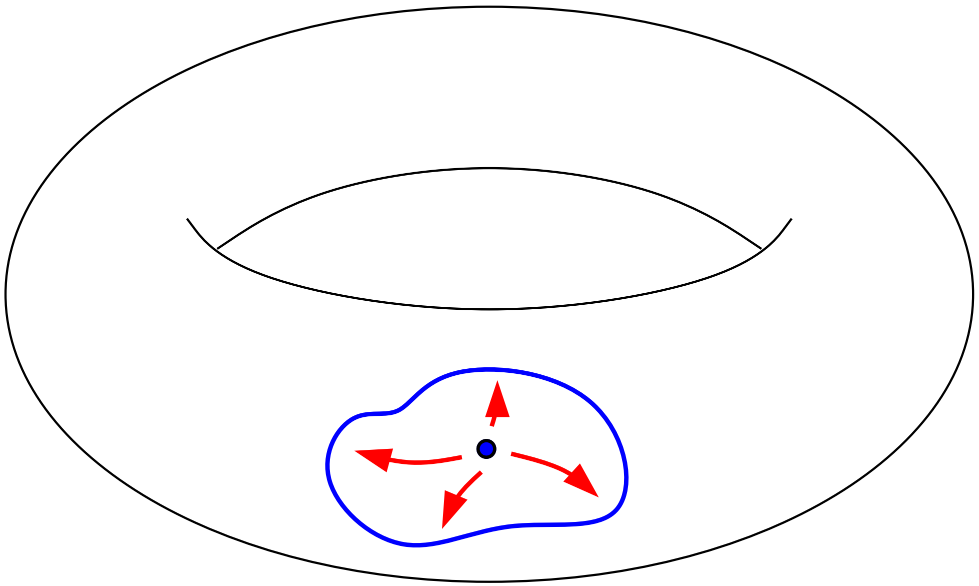
Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.



Shortest Cycles Through a Basepoint

Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.

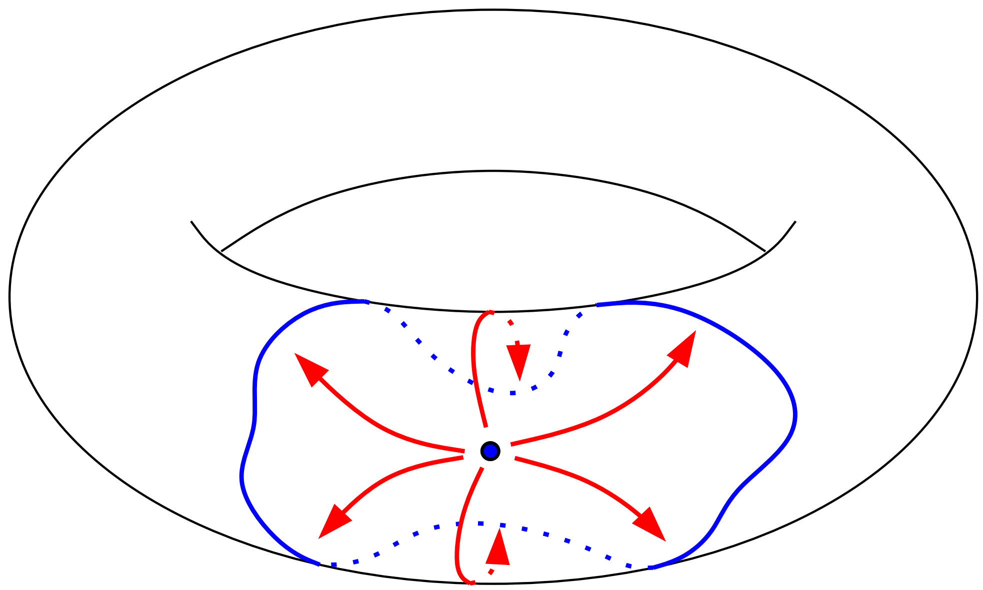
- start Dijkstra's shortest-paths algorithm from the basepoint



Shortest Cycles Through a Basepoint

Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.

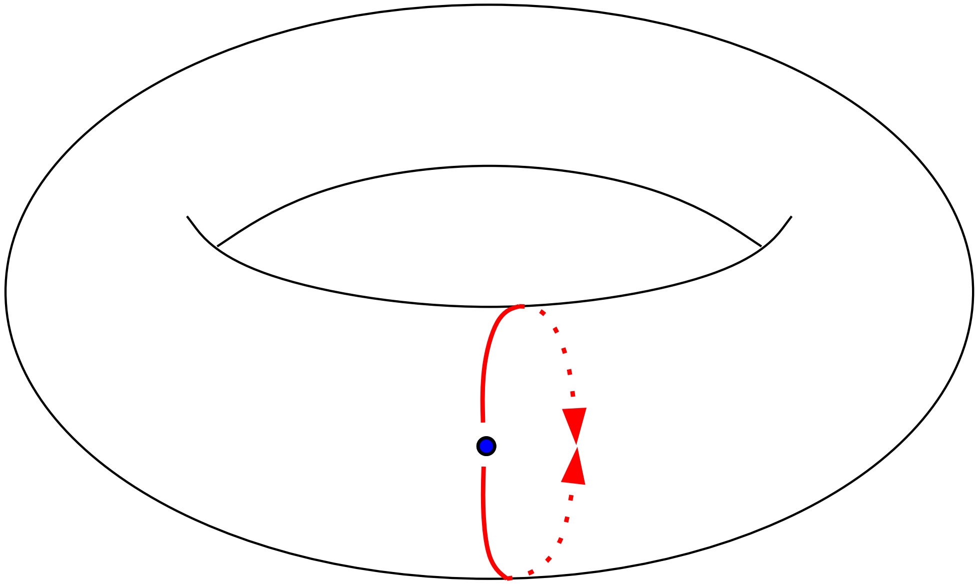
- start Dijkstra's shortest-paths algorithm from the basepoint



Shortest Cycles Through a Basepoint

Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.

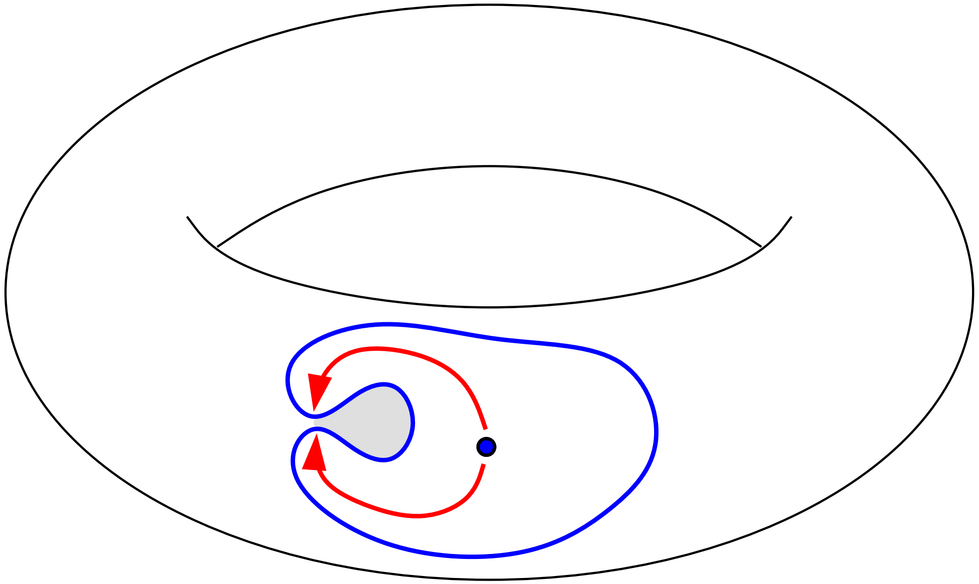
- start Dijkstra's shortest-paths algorithm from the basepoint
- stop as soon as the wave front hits itself non-trivially



Shortest Cycles Through a Basepoint

Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.

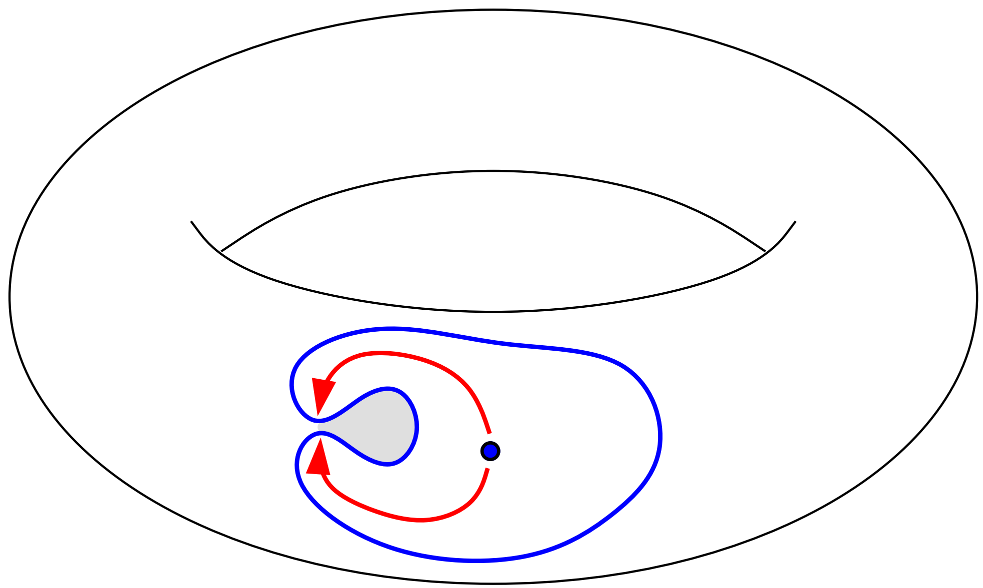
- start Dijkstra's shortest-paths algorithm from the basepoint
- stop as soon as the wave front hits itself non-trivially
- discover trivial enclosures by Euler's formula



Shortest Cycles Through a Basepoint

Erickson & Har-Peled (SoCG 2002): Shortest non-trivial cycle through a given basepoint in $O(n \log n)$ time.

- start Dijkstra's shortest-paths algorithm from the basepoint
- stop as soon as the wave front hits itself non-trivially
- discover trivial enclosures by Euler's formula



n-fold execution yields globally shortest cycle in $O(n^2 \log n)$ time.

The Genus of a Surface

Intuition: $\Omega(n^2)$ running time should not be necessary

The Genus of a Surface

Intuition: $\Omega(n^2)$ running time should not be necessary

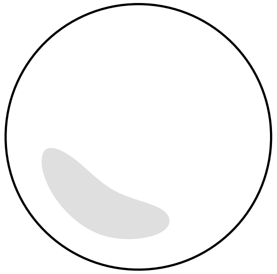
. . . at least if the *genus* of the surface is bounded.

The Genus of a Surface

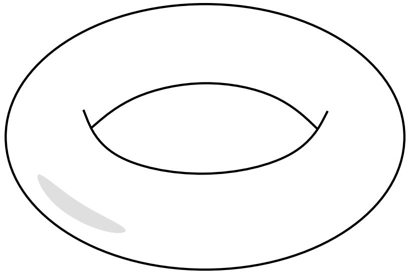
Intuition: $\Omega(n^2)$ running time should not be necessary

. . . at least if the *genus* of the surface is bounded.

genus g = number of "holes" (handles)



$g = 0$



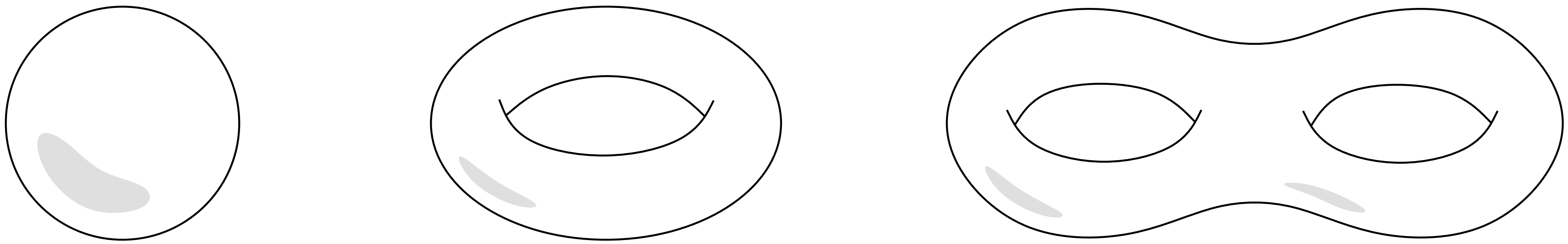
$g = 1$

The Genus of a Surface

Intuition: $\Omega(n^2)$ running time should not be necessary

... at least if the *genus* of the surface is bounded.

genus $g =$ number of "holes" (handles)



$g = 0$ $g = 1$ $g = 2$

Improvement

Thm. [Cabello & Mohar, ESA 2005] On surfaces of bounded genus, shortest non-contractible cycles can be computed in $O(n^{3/2})$ time. ($O(n^{3/2} \log n)$ for non-separating cycles)

"most appealing open problem:" almost-linear time possible?

Improvement

Thm. [Cabello & Mohar, ESA 2005] On surfaces of bounded genus, shortest non-contractible cycles can be computed in $O(n^{3/2})$ time.
($O(n^{3/2} \log n)$ for non-separating cycles)

"most appealing open problem:" almost-linear time possible?

New Result:

On orientable surfaces of bounded genus,
shortest non-contractible and shortest non-separating cycles
can be computed in $O(n \log n)$ time.

Improvement

Thm. [Cabello & Mohar, ESA 2005] On surfaces of bounded genus, shortest non-contractible cycles can be computed in $O(n^{3/2})$ time. ($O(n^{3/2} \log n)$ for non-separating cycles)

"most appealing open problem:" almost-linear time possible?

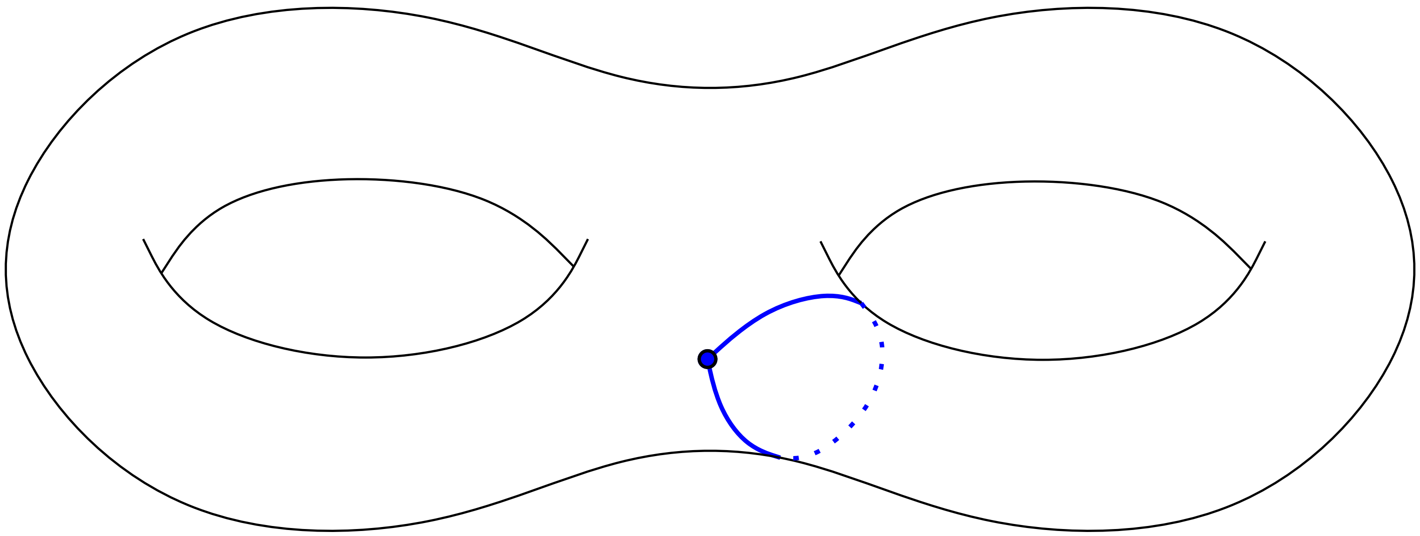
New Result:

On orientable surfaces of bounded genus,
shortest non-contractible and shortest non-separating cycles
can be computed in $O(n \log n)$ time.

(*Reminder:* exponential dependence on g!
genus-independent possible in $O(n^2 \log n)$)

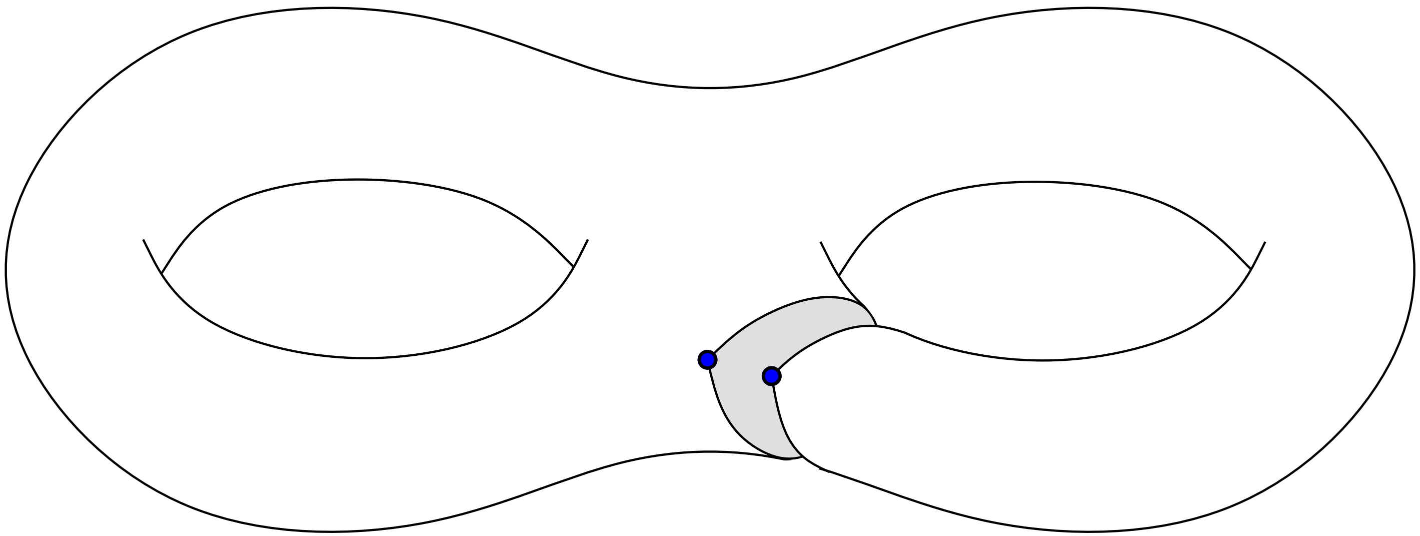
Technical Tools I: Minimal System of Loops

- fix arbitrary basepoint x
- find shortest non-separating loop ℓ_1 through x



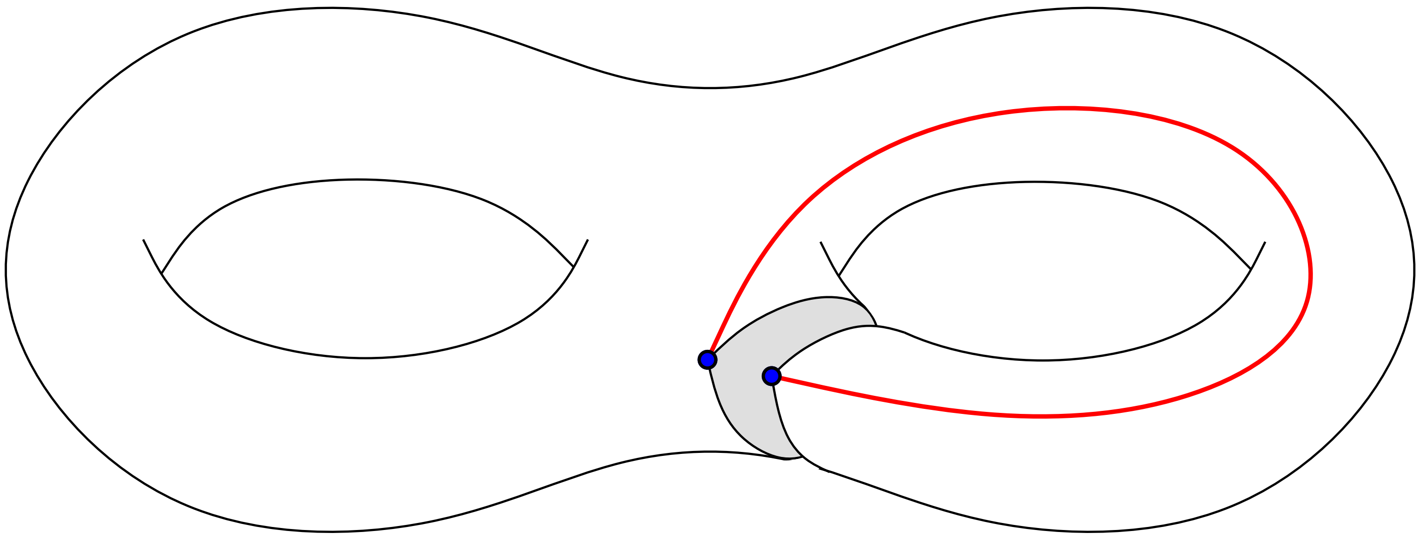
Technical Tools I: Minimal System of Loops

- fix arbitrary basepoint x
- find shortest non-separating loop ℓ_1 through x
- cut $\mathcal{M}$ along ℓ_1 (duplicating vertices and edges)



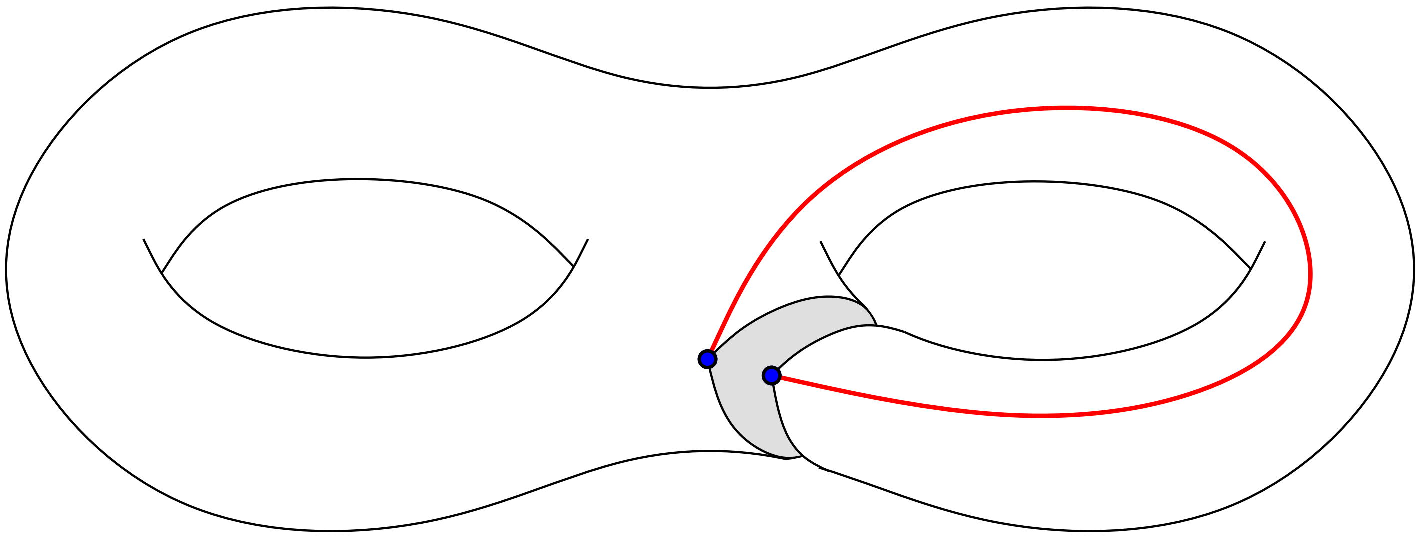
Technical Tools I: Minimal System of Loops

- fix arbitrary basepoint x
- find shortest non-separating loop ℓ_1 through x
- cut $\mathcal{M}$ along ℓ_1 (duplicating vertices and edges)
- find shortest non-separating loop ℓ_2 in $\mathcal{M}$ ✂ ℓ_1 from x to x



Technical Tools I: Minimal System of Loops

- fix arbitrary basepoint x
- find shortest non-separating loop ℓ_1 through x
- cut $\mathcal{M}$ along ℓ_1 (duplicating vertices and edges)
- find shortest non-separating loop ℓ_2 in $\mathcal{M}$ ✂ ℓ_1 from x to x
- cut along ℓ_2

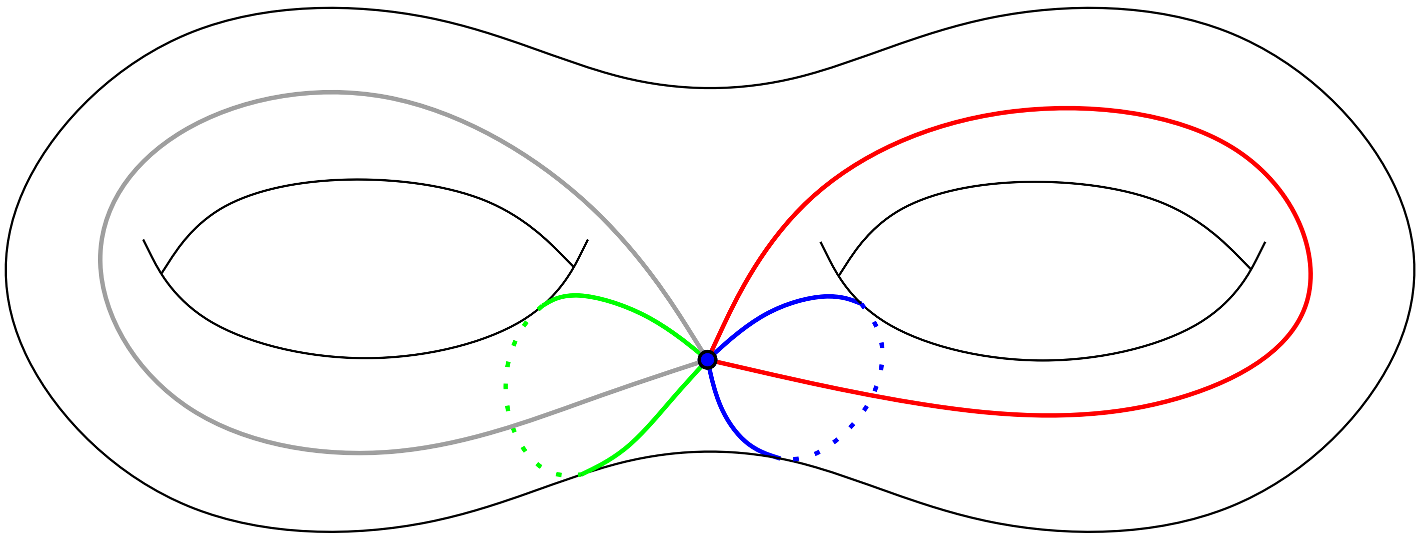


Technical Tools I: Minimal System of Loops

- fix arbitrary basepoint x
- find shortest non-separating loop ℓ_1 through x
- cut $\mathcal{M}$ along ℓ_1 (duplicating vertices and edges)

$$\vdots$$

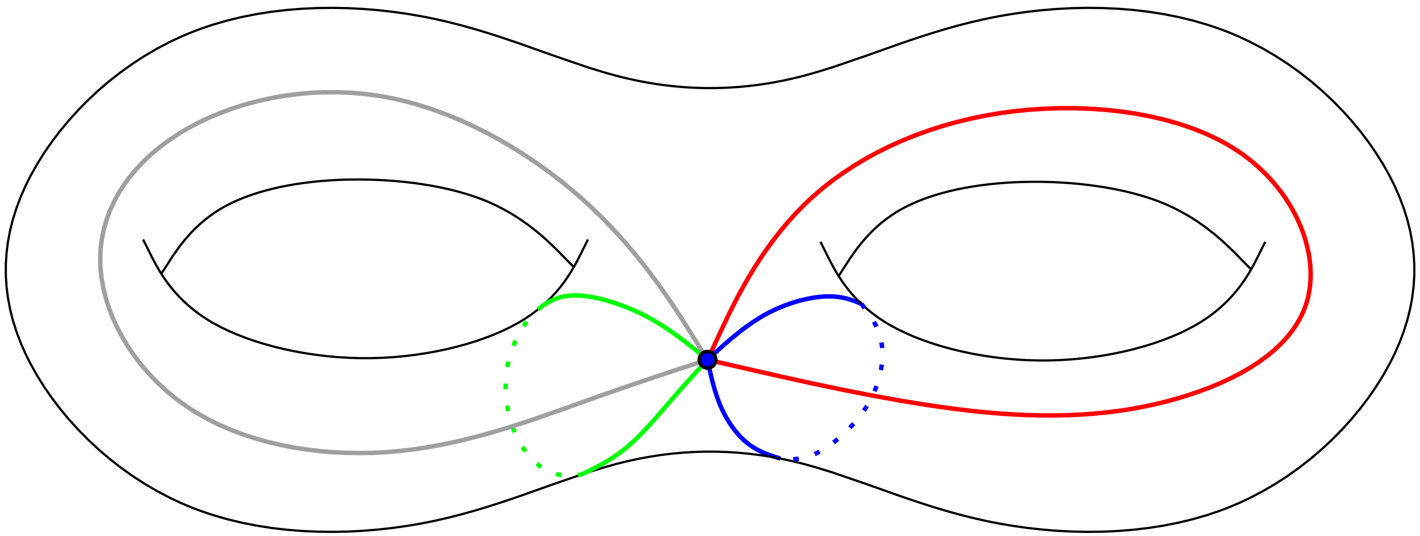
- find shortest non-separating loop ℓ_{2g} in $\mathcal{M}$ ✂ $(\ell_1 \cup \cdots \cup \ell_{2g-1})$ form x to x



Technical Tools I: Minimal System of Loops

Such a *minimal system of loops*

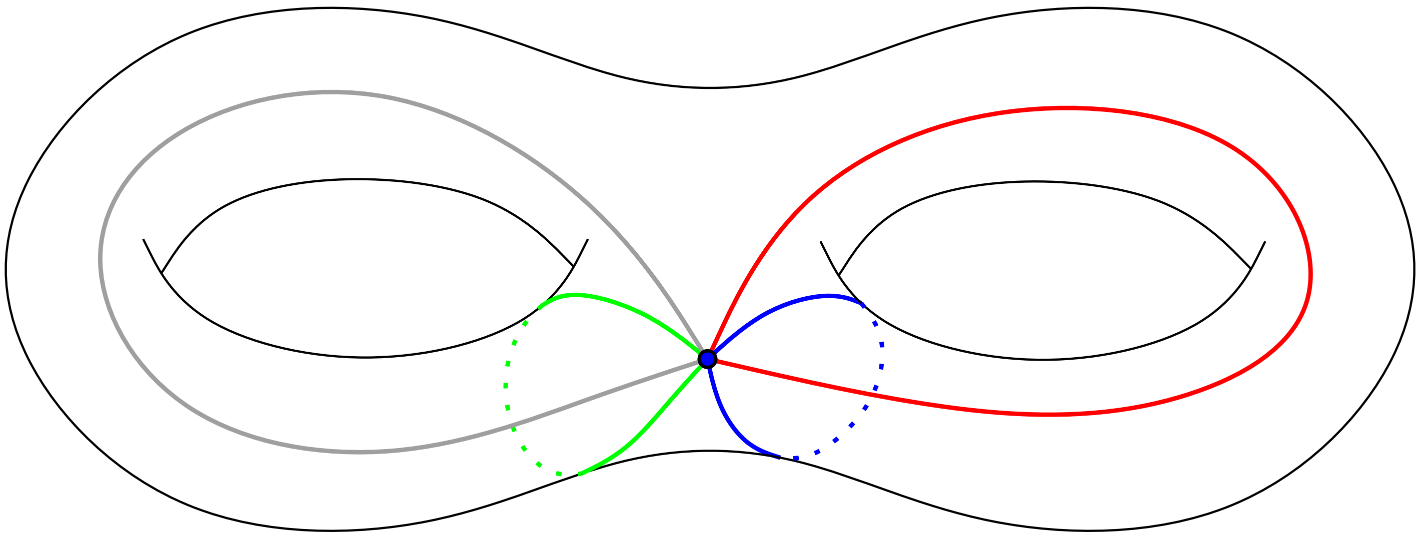
- decomposes the surface into a disk!



Technical Tools I: Minimal System of Loops

Such a *minimal system of loops*

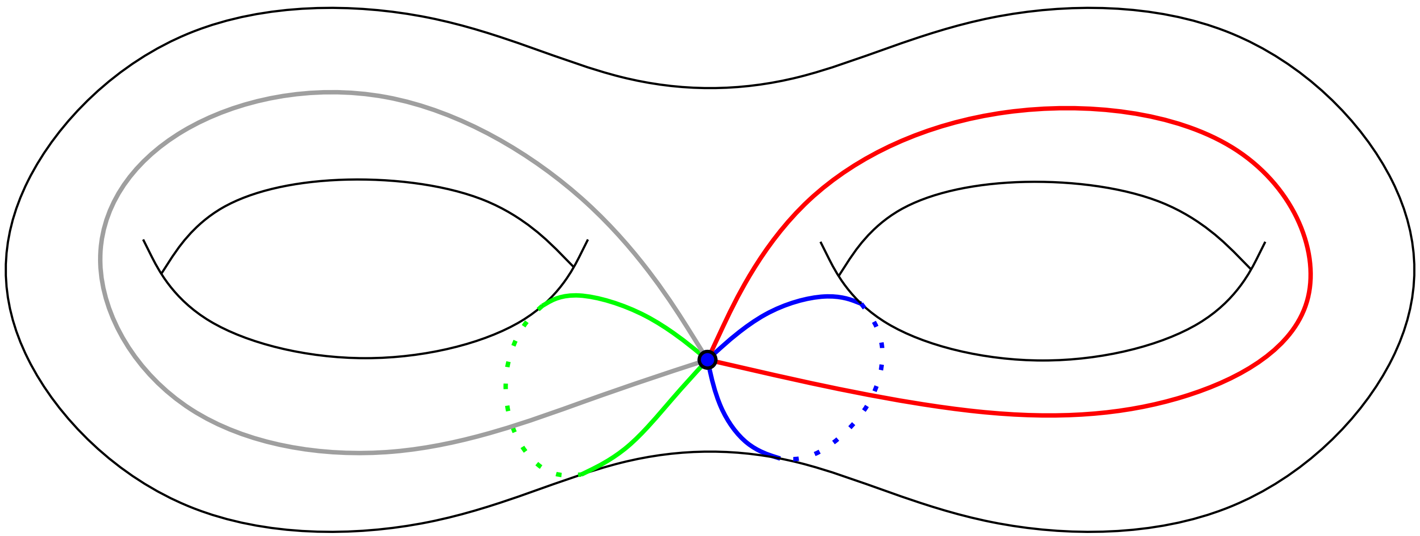
- decomposes the surface into a disk!
- is *minimal* (each loop is minimal in its homotopy class)



Technical Tools I: Minimal System of Loops

Such a *minimal system of loops*

- decomposes the surface into a disk!
- is *minimal* (each loop is minimal in its homotopy class)
- can be computed in linear time (instead of just $O(gn \log n)$) using *Eppstein's tree-cotree decomposition* [Erickson & Whittlesey, SODA 2005]



Technical Tools II

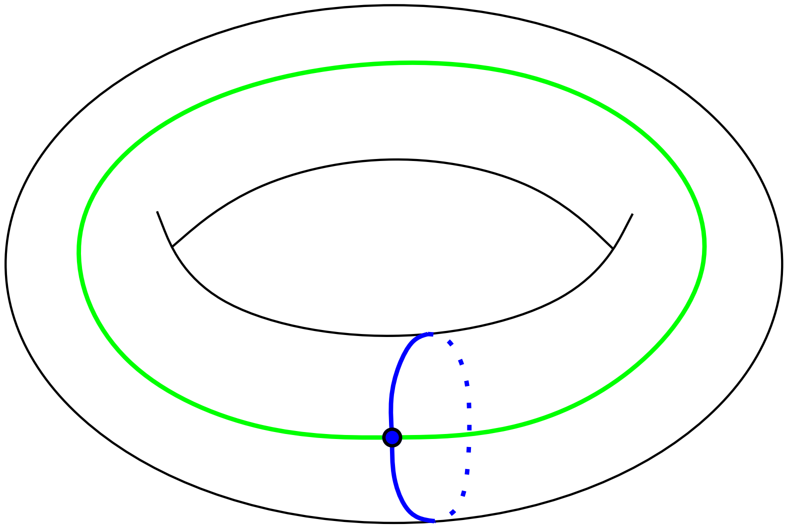
Use system of loops to form *universal cover*:

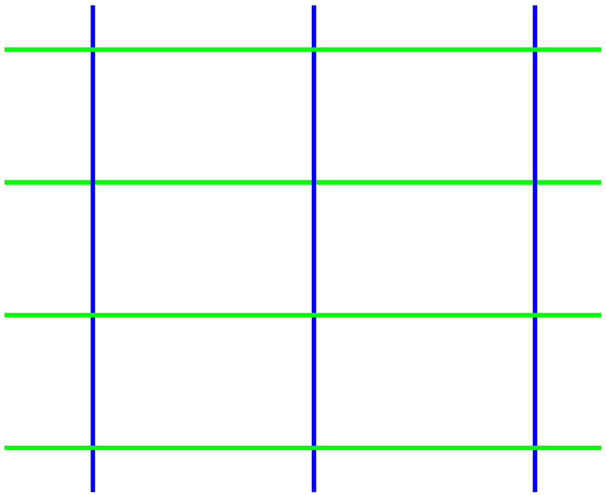
- *fundamental domain* $F := \mathcal{M} ✂ \bigcup \ell_i$,

Technical Tools II

Use system of loops to form *universal cover* :

- *fundamental domain* $F := \mathcal{M} ✂ \bigcup \ell_i$,
- glue "infinitely many" F-copies along the boundaries ℓ_i to form an infinite plane $\mathcal{H}$



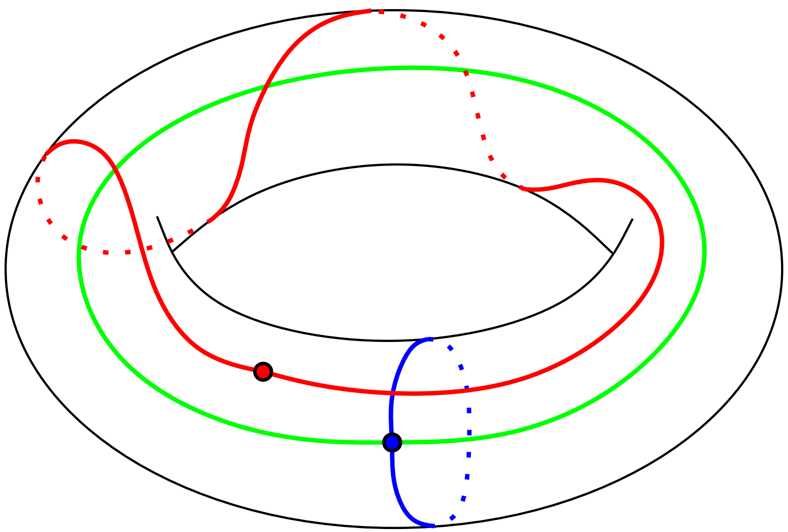


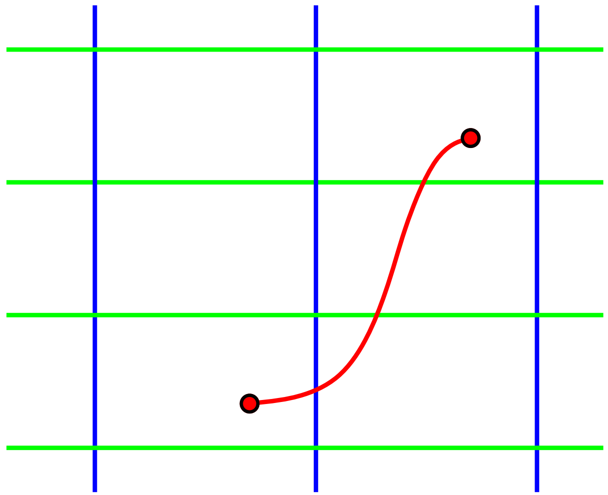
Technical Tools II

Use system of loops to form *universal cover* :

- *fundamental domain* $\mathsf{F} := \mathcal{M}$ ✂ $\bigcup \ell_i$,
- glue "infinitely many" F-copies along the boundaries ℓ_i to form an infinite plane $\mathcal{H}$, so that

non-trivial cycles in $\mathcal{M}$ $\longleftrightarrow$ non-closed paths in $\mathcal{H}$



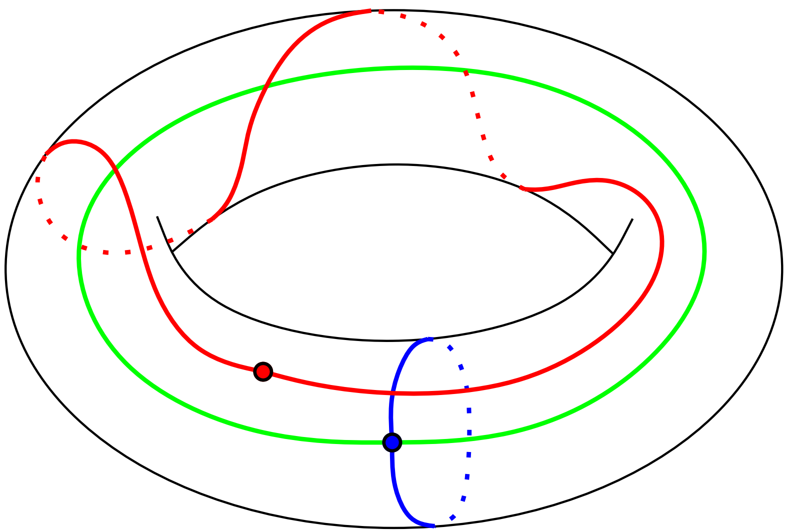


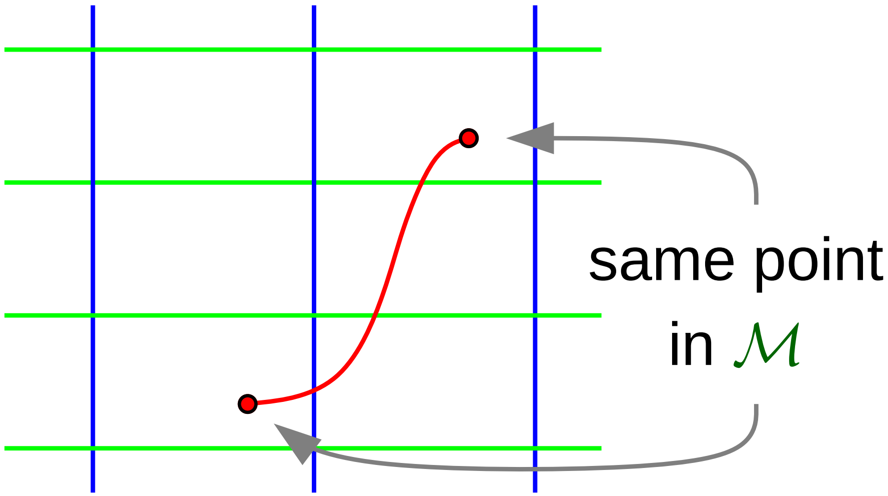
Technical Tools II

Use system of loops to form *universal cover* :

- *fundamental domain* $\mathsf{F} := \mathcal{M} \text{✂} \bigcup \ell_i$,
- glue "infinitely many" F-copies along the boundaries ℓ_i to form an infinite plane $\mathcal{H}$, so that

non-trivial cycles in $\mathcal{M}$ $\longleftrightarrow$ non-closed paths in $\mathcal{H}$

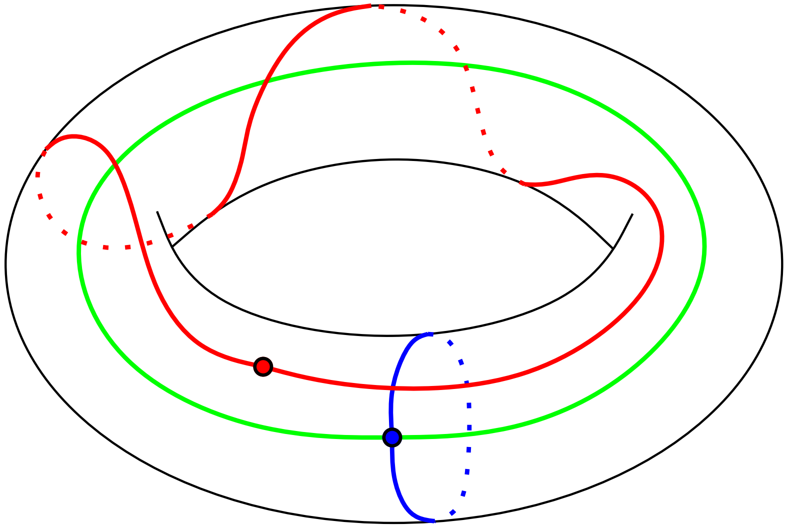


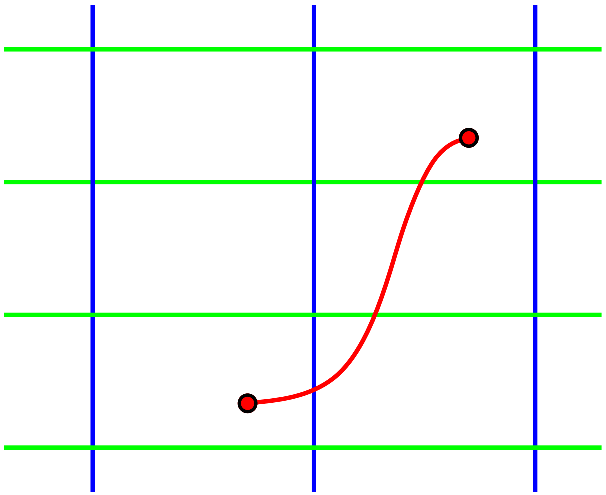


Technical Tools III

Lem. [Cabello & Mohar, ESA 2005] (*Crossing Bound*)
Let $\ell_1, \ldots, \ell_{2g}$ be a minimal system of loops.
Then there exists a shortest non-contractible (non-separating) cycle that crosses each loop ℓ_i at most twice.

Proof via *3-Path Condition* [Thomassen, JCT(A) 1990]



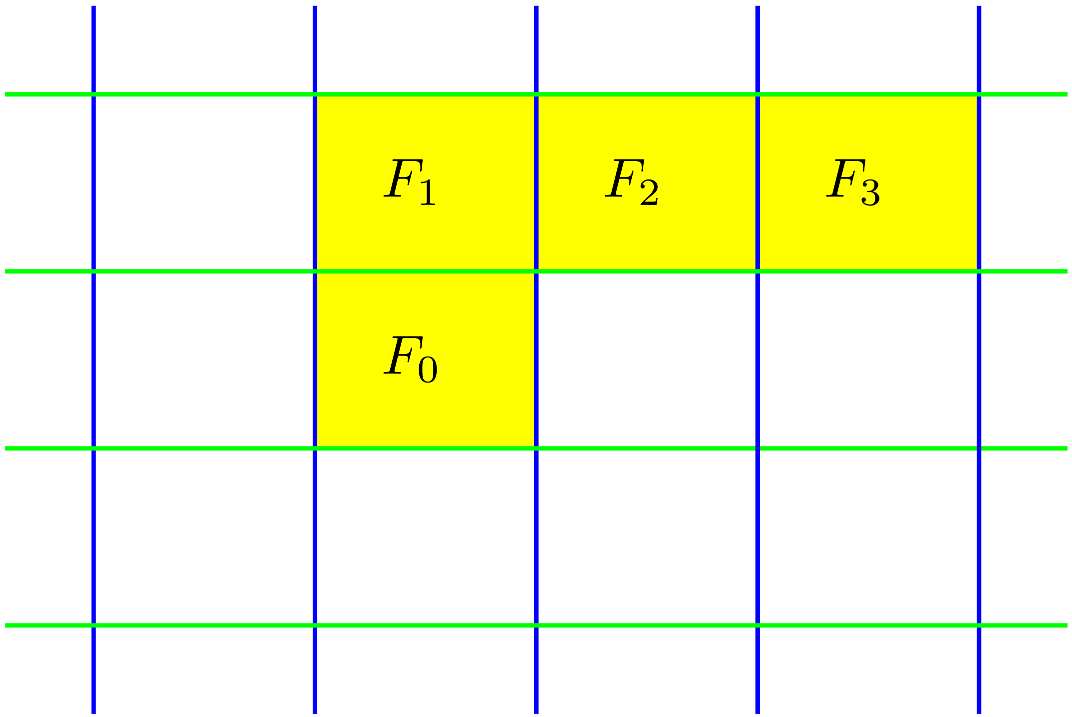


Meta Paths

- Crossing bound guarantees a shortest non-trivial cycle through $\leq 4g$ fundamental domains

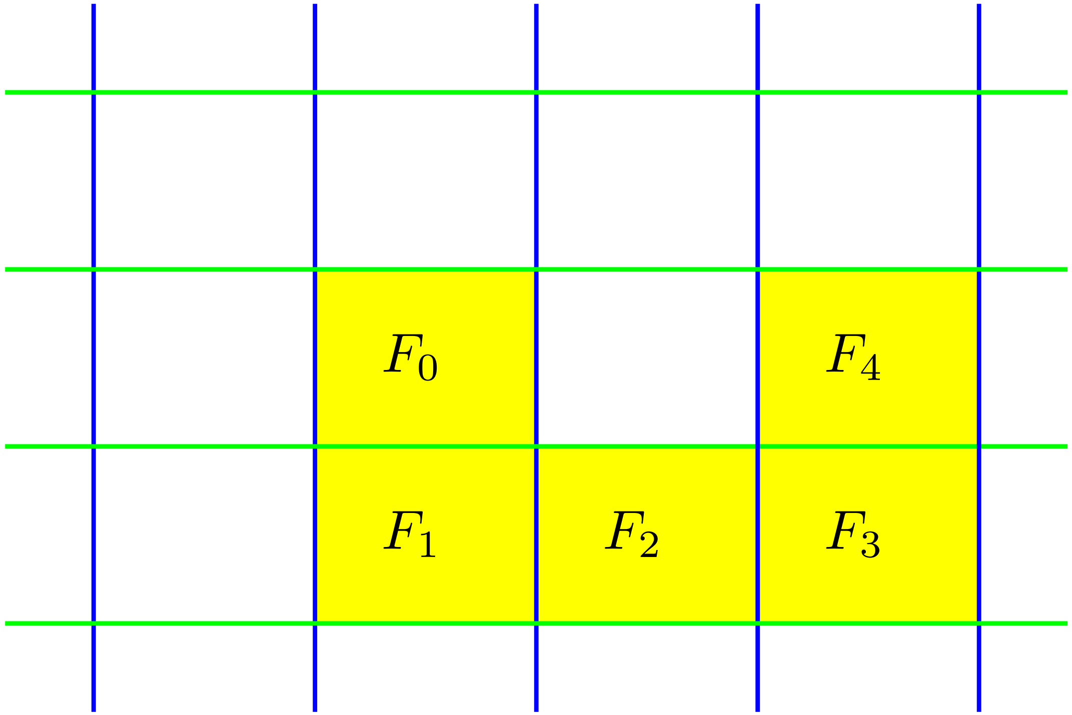
Meta Paths

- Crossing bound guarantees a shortest non-trivial cycle through $\leq 4g$ fundamental domains
- Idea: test all possible types of such "meta paths"



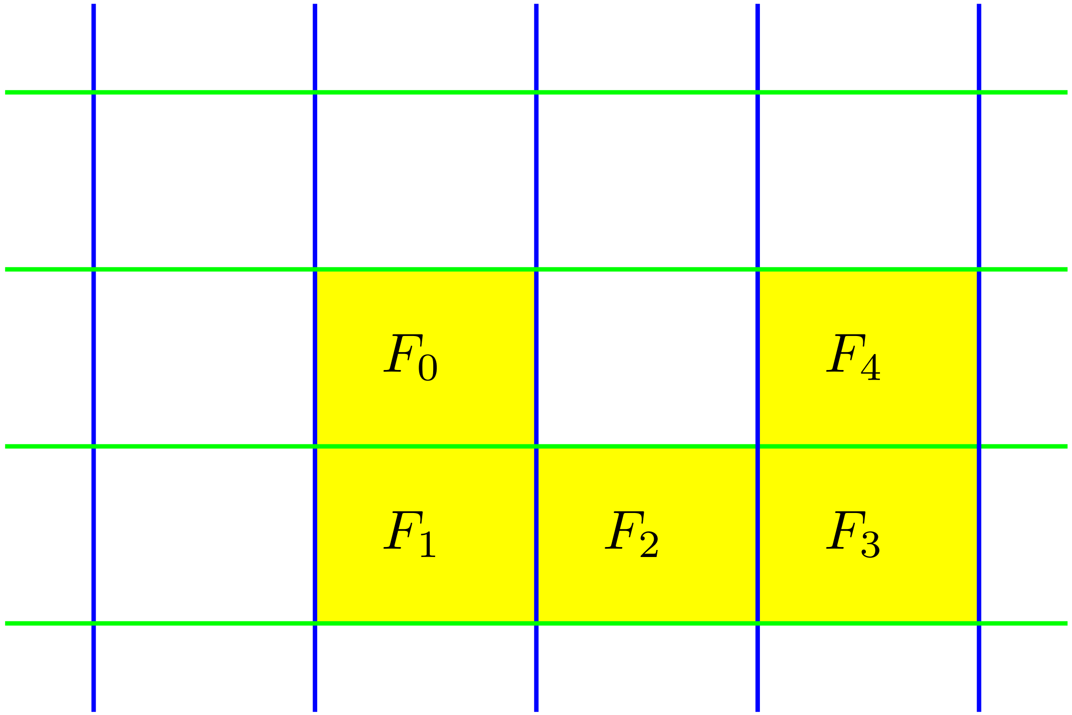
Meta Paths

- Crossing bound guarantees a shortest non-trivial cycle through $\leq 4\mathrm{g}$ fundamental domains
- Idea: test all possible types of such "meta paths"



Meta Paths

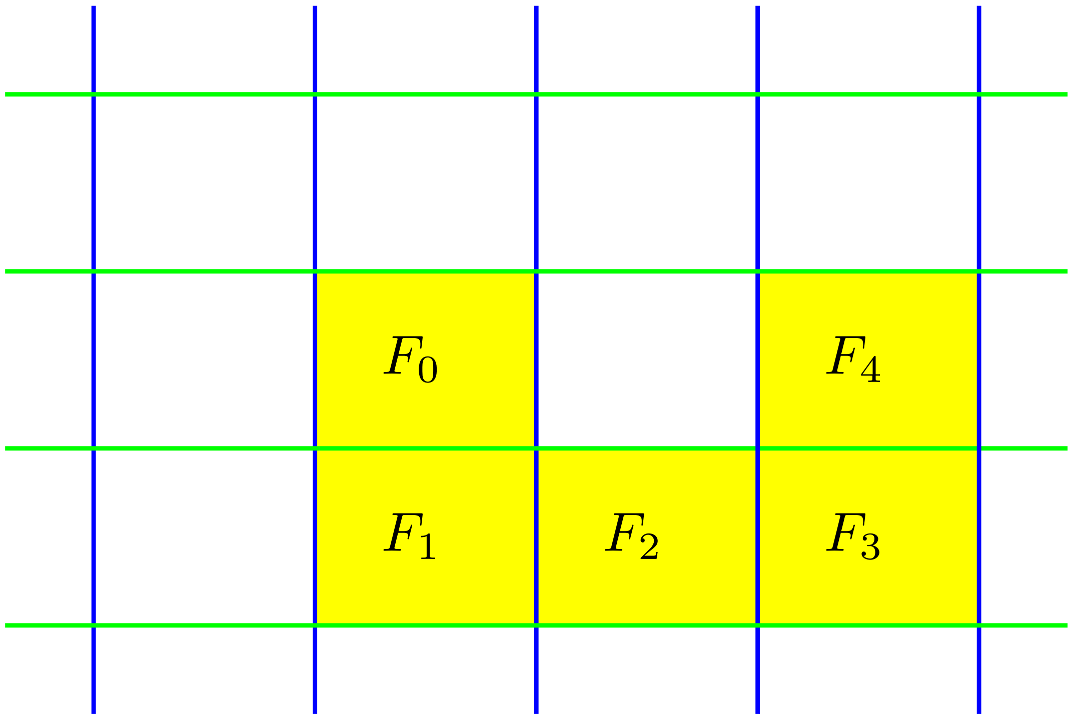
- Crossing bound guarantees a shortest non-trivial cycle through $\leq 4g$ fundamental domains
- Idea: test all possible types of such "meta paths"
- How to find a shortest cycle in such a meta path?



Meta Paths — *Naive Search*

For each meta path:

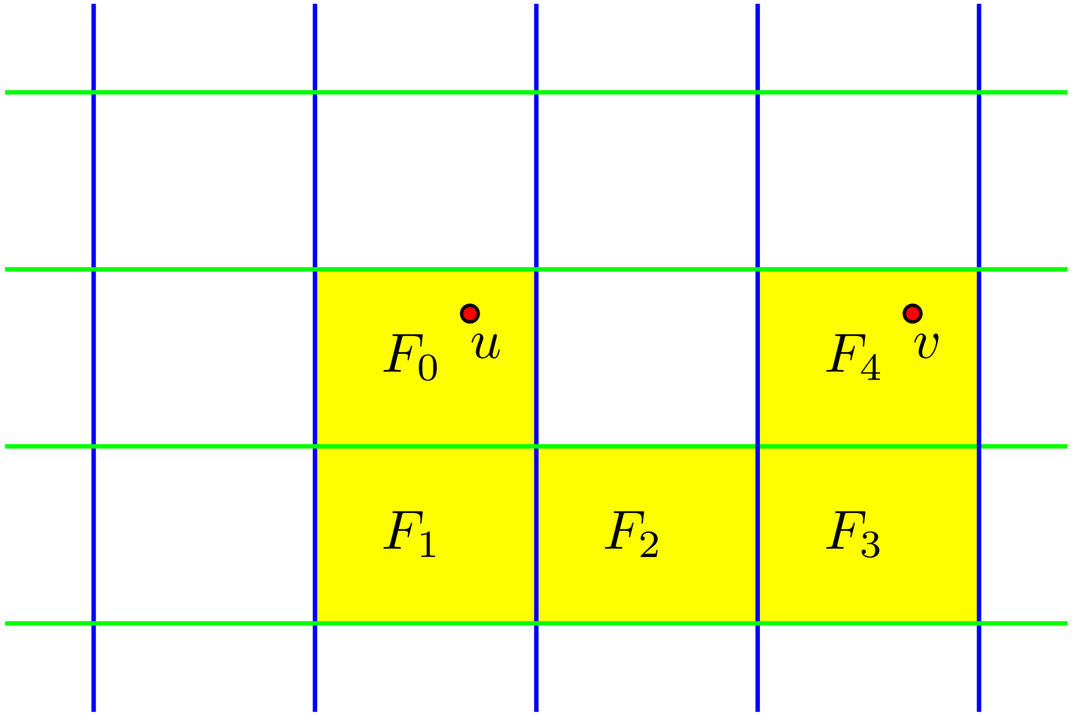
- consider n corresponding points pairs u, v



Meta Paths — *Naive Search*

For each meta path:

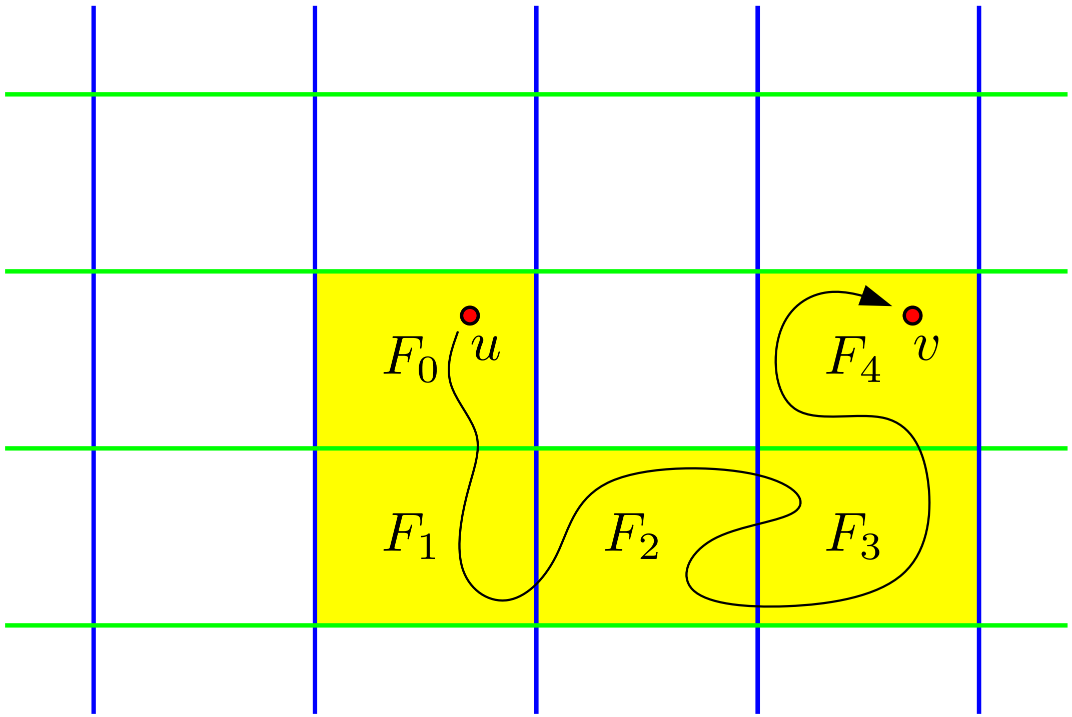
- consider n corresponding points pairs u, v



Meta Paths — *Naive Search*

For each meta path:

- consider n corresponding points pairs u, v
- perform shortest-path search for each pair

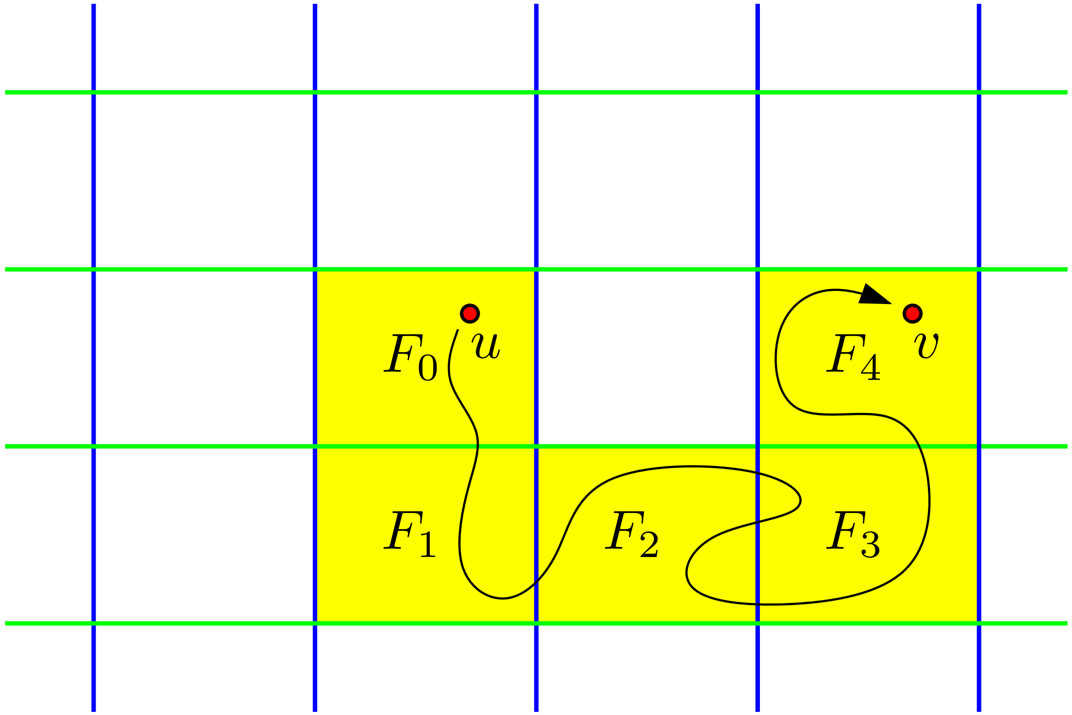


Meta Paths — *Naive Search*

For each meta path:

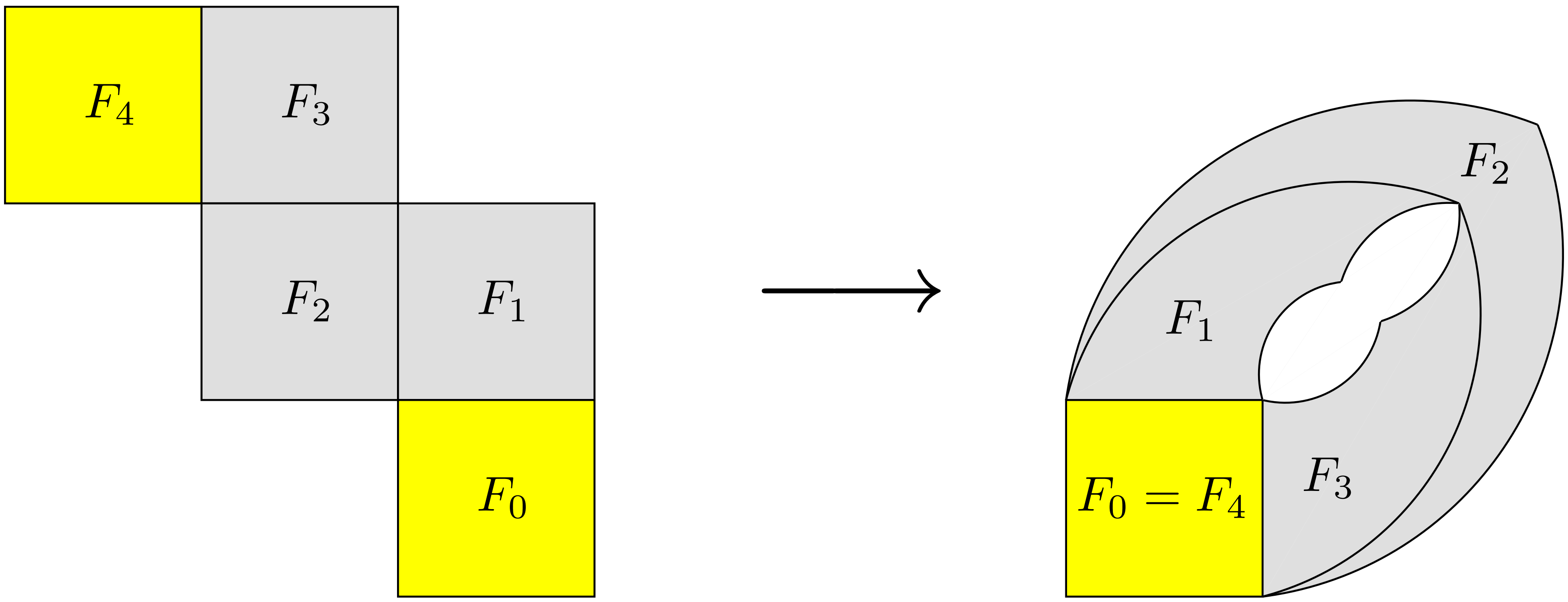
- consider n corresponding points pairs u, v
- perform shortest-path search for each pair

Total (worst-case) running time: $\Omega(n^2)$



Forming Cylinders

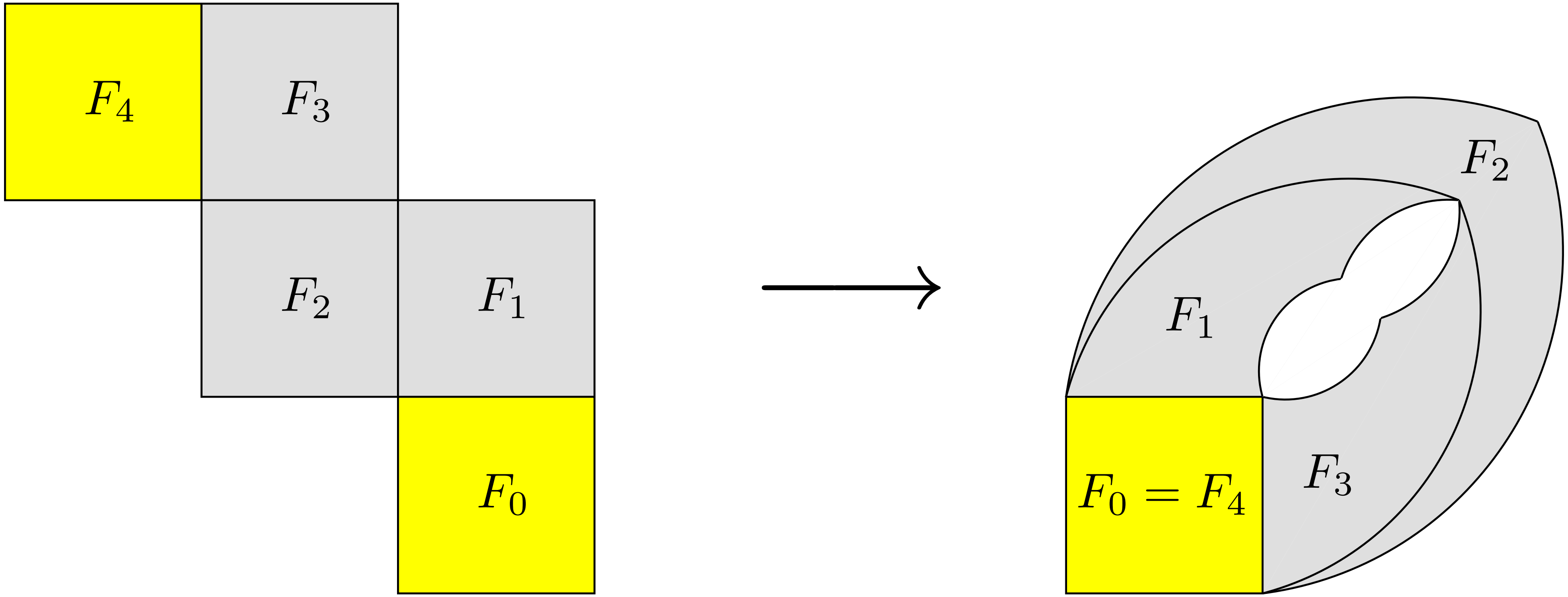
Trick: identify the terminal domains of a meta path to form a cylinder



Forming Cylinders

Trick: identify the terminal domains of a meta path to form a cylinder

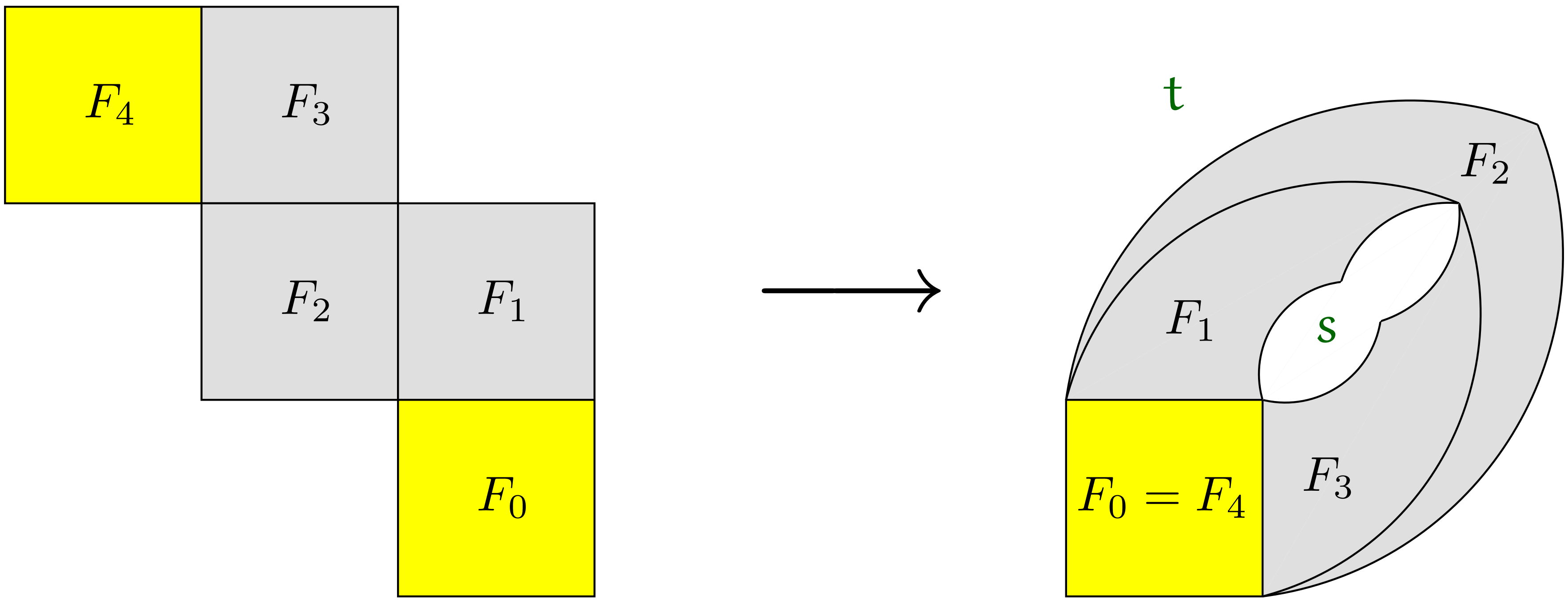
Thm. [Frederickson, 1987] In a planar graph with two vertices s,t, one can find a minimal s-t-cut in $O(n \log n)$ time.



Forming Cylinders

Trick: identify the terminal domains of a meta path to form a cylinder

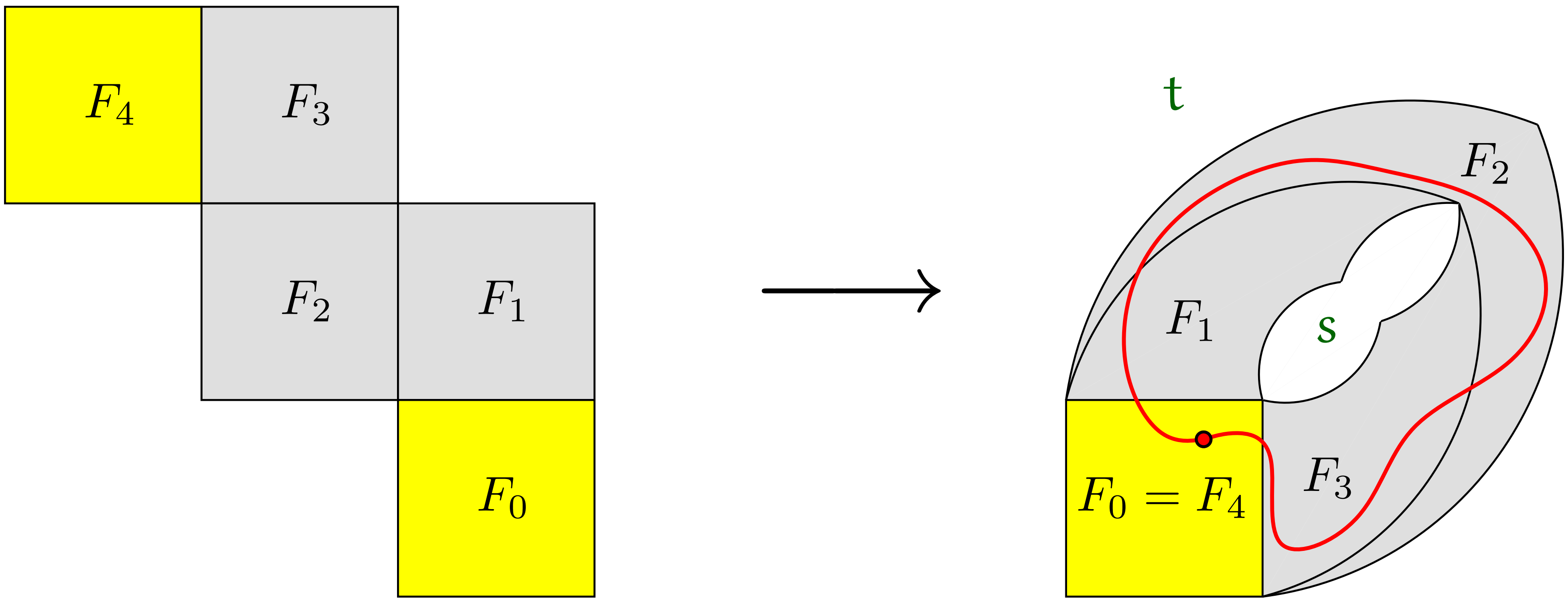
Thm. [Frederickson, 1987] In a planar graph with two vertices s,t, one can find a minimal s-t-cut in $O(n \log n)$ time.



Forming Cylinders

Trick: identify the terminal domains of a meta path to form a cylinder

Thm. [Frederickson, 1987] In a planar graph with two vertices s,t, one can find a minimal s-t-cut in $O(n \log n)$ time.

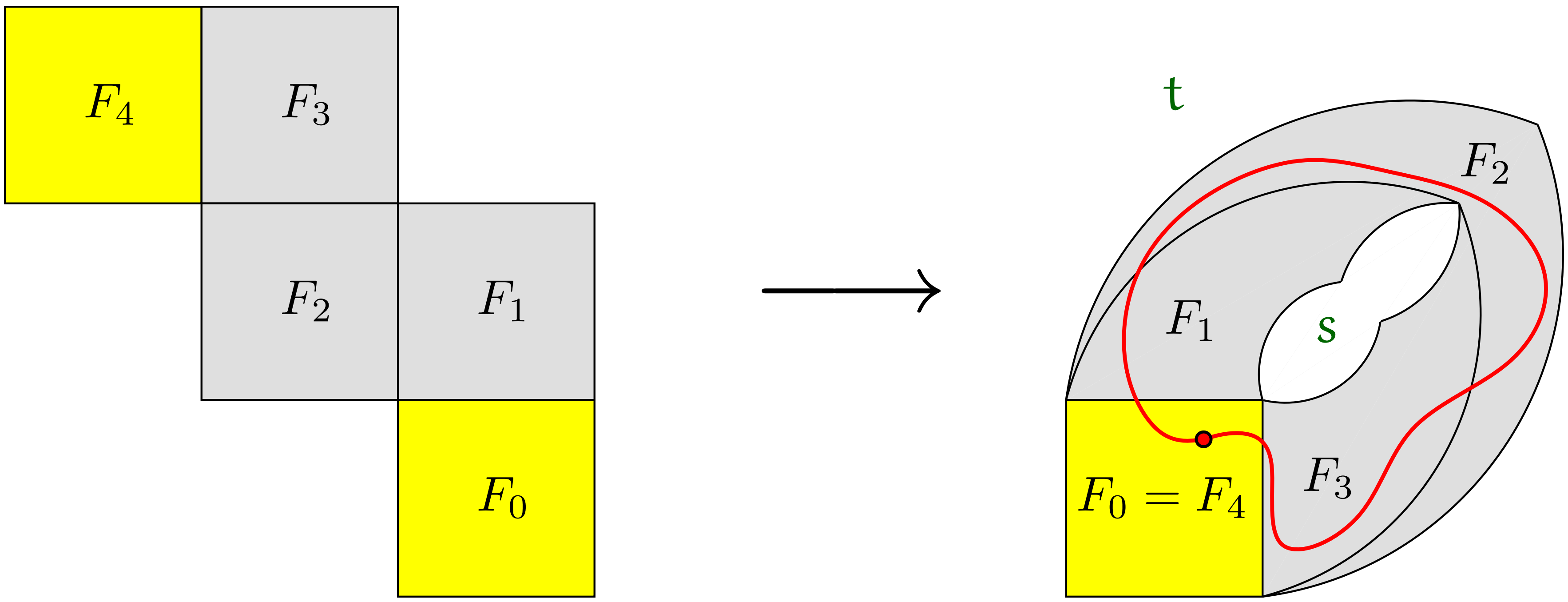


Forming Cylinders

Trick: identify the terminal domains of a meta path to form a cylinder

Thm. [Frederickson, 1987] In a planar graph with two vertices s,t, one can find a minimal s-t-cut in $O(n \log n)$ time.

Warning : meta paths must be manyfolds! (can be guaranteed)



The Whole Algorithm

- generate minimal system of loops $\ell_1, \ldots, \ell_{2g}$ and fundamental domain $F := \mathcal{M} ✂ \bigcup \ell_i$,

The Whole Algorithm

- generate minimal system of loops $\ell_1, \ldots, \ell_{2g}$ and fundamental domain $F := \mathcal{M} ✂ \bigcup \ell_i$,
- consider all planar meta paths $F_0, F_1, F_2, \ldots, F_k$ of length $k \leq 4g$

The Whole Algorithm

- generate minimal system of loops $\ell_1, \ldots, \ell_{2g}$ and fundamental domain $F := \mathcal{M} ✂ \bigcup \ell_i$,
- consider all planar meta paths $F_0, F_1, F_2, \ldots, F_k$ of length $k \leq 4g$
- merge each such meta path into a cylinder C

The Whole Algorithm

- generate minimal system of loops $\ell_1, \ldots, \ell_{2g}$ and fundamental domain $F := \mathcal{M} ✂ \bigcup \ell_i$,
- consider all planar meta paths $F_0, F_1, F_2, \ldots, F_k$ of length $k \leq 4g$
- merge each such meta path into a cylinder C
- ...and compute shortest cycle around C

The Whole Algorithm

- generate minimal system of loops $\ell_1, \ldots, \ell_{2g}$
  and fundamental domain $F := \mathcal{M} \mathbin{\text{✂}} \bigcup \ell_i$,
- consider all planar meta paths $F_0, F_1, F_2, \ldots, F_k$
  of length $k \leq 4g$
- merge each such meta path into a cylinder C
- . . . and compute shortest cycle around C

Theorem. On orientable surfaces of bounded genus, shortest non-contractible and shortest non-separating cycles can be computed in $O(n \log n)$ time.

Conclusion

Theorem. On orientable surfaces of bounded genus, shortest non-contractible and shortest non-separating cycles can be computed in $O(n \log n)$ time.

Conclusion

Theorem. On orientable surfaces of bounded genus, shortest non-contractible and shortest non-separating cycles can be computed in $O(n \log n)$ time.

Non-orientable manifolds: [Cabello, unpublished / –written]

Conclusion

Theorem. On orientable surfaces of bounded genus, shortest non-contractible and shortest non-separating cycles can be computed in $O(n \log n)$ time.

Non-orientable manifolds: [Cabello, unpublished / –written]

Open problem: How to avoid the exponential dependence on g?

(*Remember:* genus-independent in $O(n^2 \log n)$)

Conclusion

Theorem. On orientable surfaces of bounded genus, shortest non-contractible and shortest non-separating cycles can be computed in $O(n \log n)$ time.

Non-orientable manifolds: [Cabello, unpublished / –written]

Open problem: How to avoid the exponential dependence on g?

(*Remember:* genus-independent in $O(n^2 \log n)$)

Maybe trade-off possible between n and g?

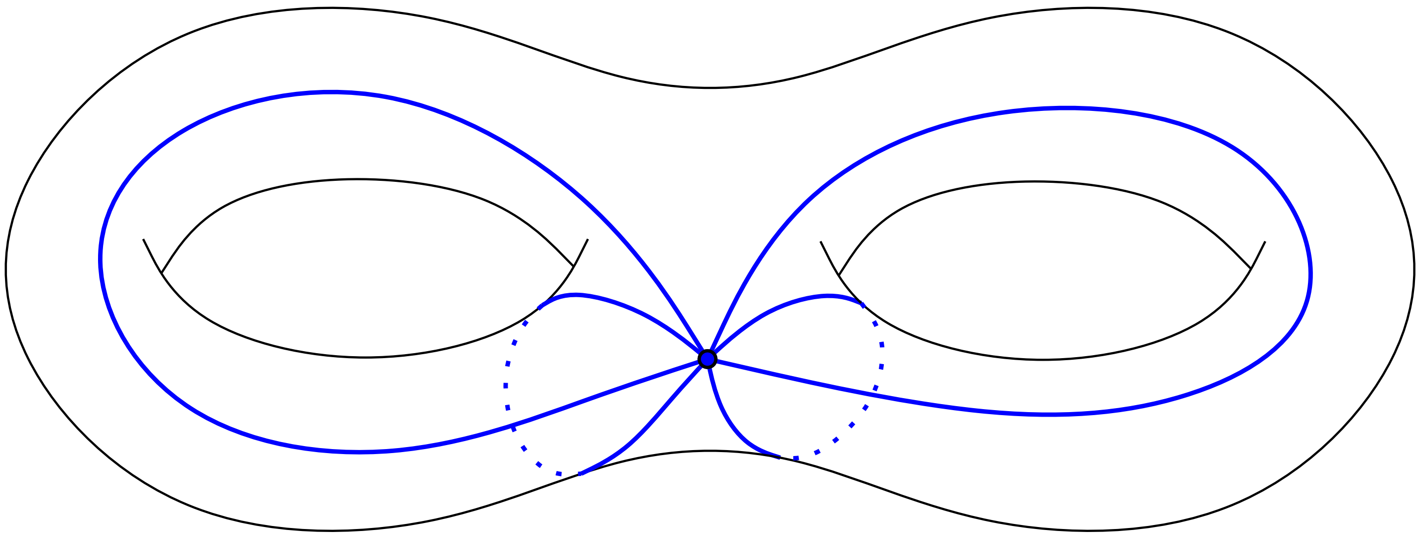
Outlook: Cut Graphs

Want to cut $\mathcal{M}$ (along several paths) to obtain a (topological) disk.

Outlook: Cut Graphs

Want to cut $\mathcal{M}$ (along several paths) to obtain a (topological) disk.

Example: $\mathcal{M}$ $-$ system of loops $=$ disk

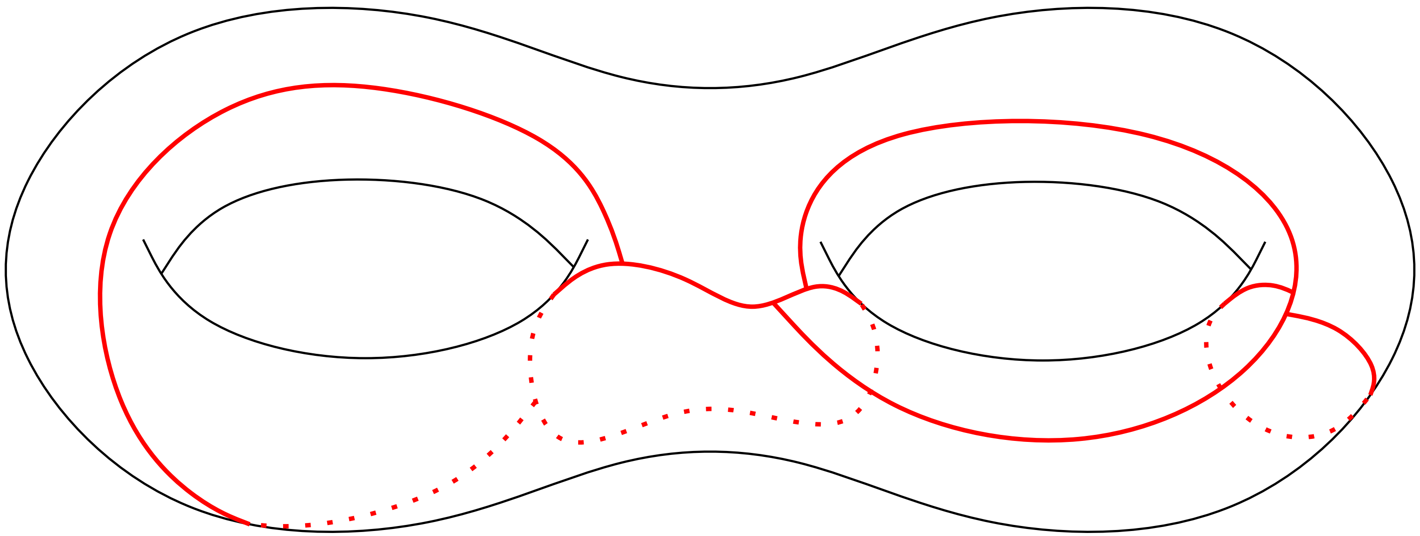


Outlook: Cut Graphs

Want to cut $\mathcal{M}$ (along several paths) to obtain a (topological) disk.

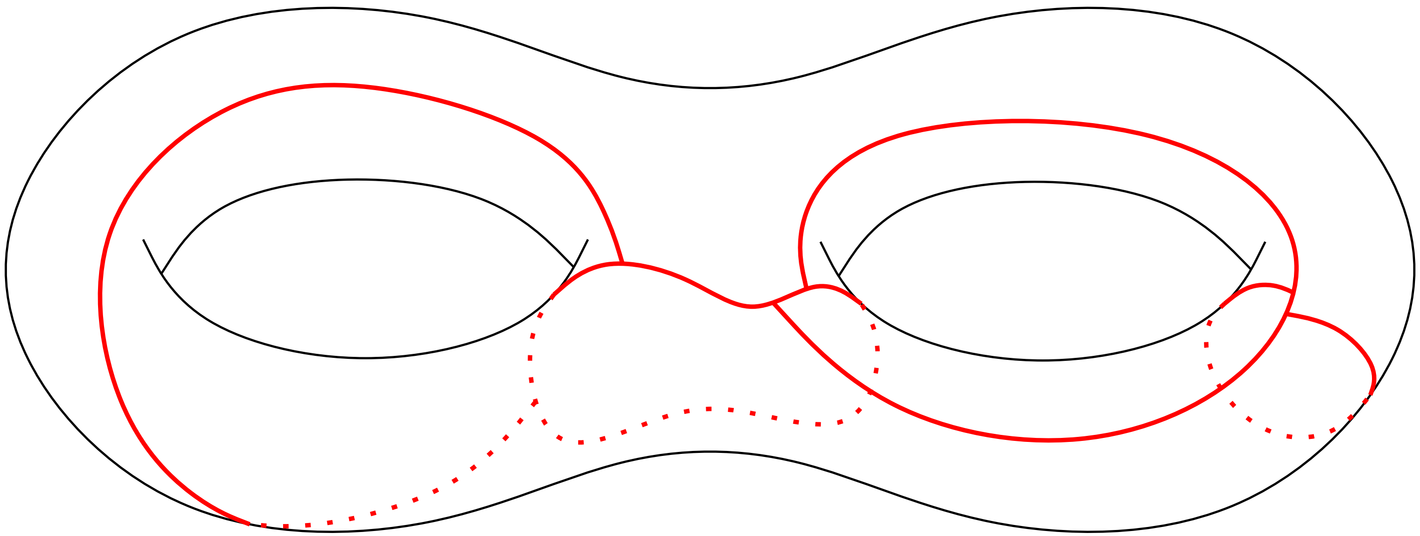
Example: $\mathcal{M}$ $-$ system of loops $=$ disk

Def. a *cut graph* is a collection of paths that cut $\mathcal{M}$ into a disk.



Outlook: Cut Graphs

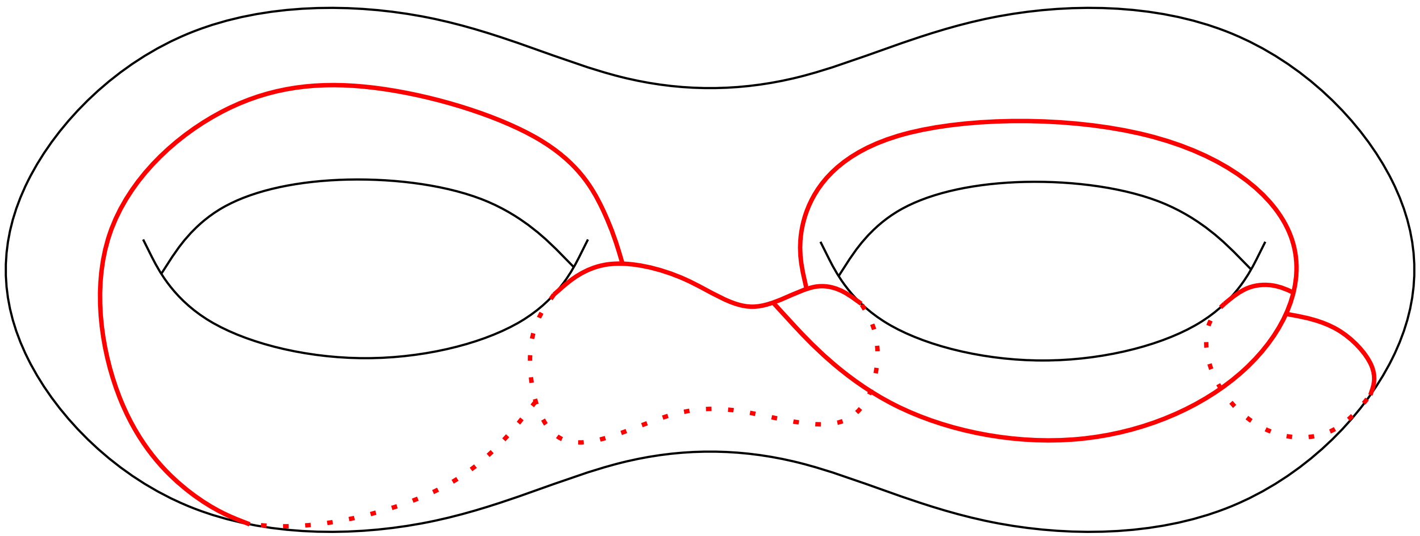
Def. a *cut graph* is a collection of paths that cut $\mathcal{M}$ into a disk.



Outlook: Cut Graphs

Def. a *cut graph* is a collection of paths that cut $\mathcal{M}$ into a disk.

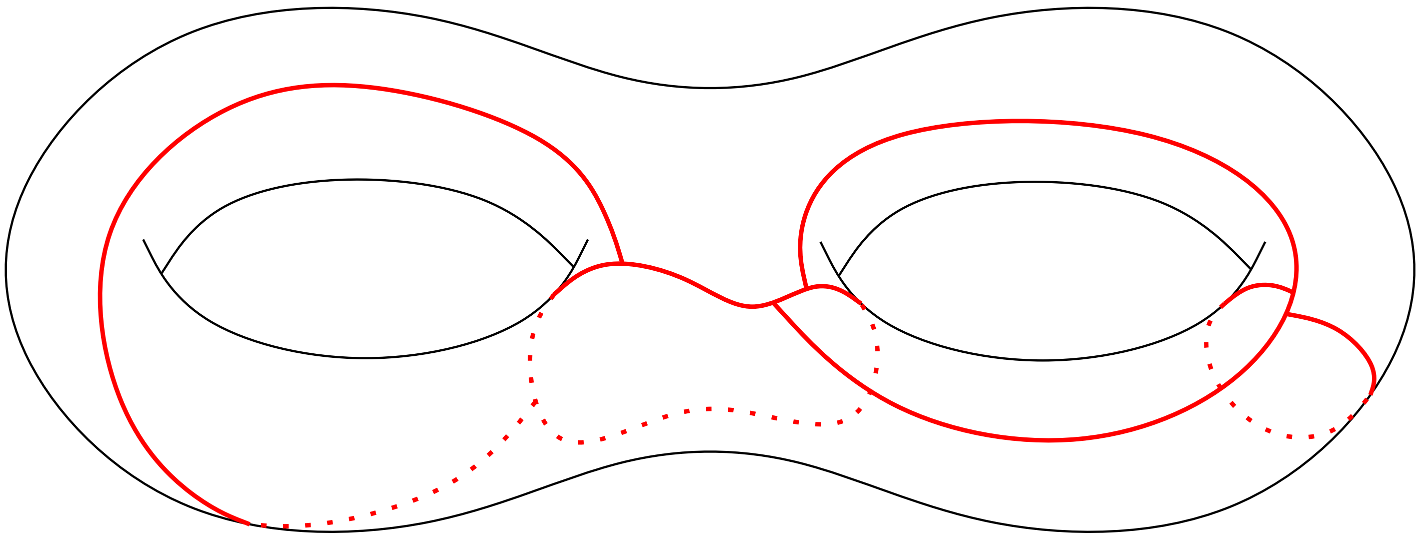
Thm. [Erickson & Har-Peled, SoCG 2002]
Computing a minimum cut graph is NP-hard. (requires large genus)



Outlook: Cut Graphs

Def. a *cut graph* is a collection of paths that cut $\mathcal{M}$ into a disk.

Thm. [Erickson & Har-Peled, SoCG 2002]
Computing a minimum cut graph is NP-hard. (requires large genus)
But can be $O(\log^2 g)$-approximated in $O(g^2 n \log n)$ time.

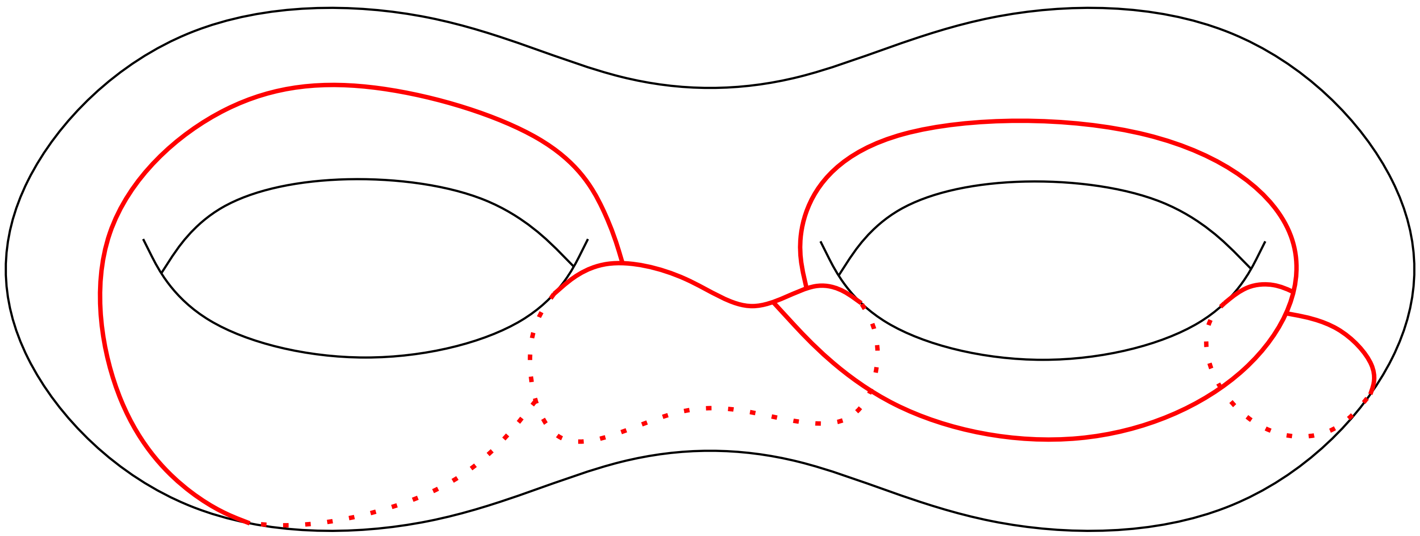


Outlook: Cut Graphs

Def. a *cut graph* is a collection of paths that cut $\mathcal{M}$ into a disk.

Thm. [Erickson & Har-Peled, SoCG 2002]
Computing a minimum cut graph is NP-hard. (requires large genus)
But can be $O(\log^2 g)$-approximated in $O(g^2 n \log n)$ time.

Question: Fixed-parameter tractable? (in g)
Because of similarity to Steiner-tree problem.



Outlook: Cut Graphs

Theorem. [K & Steurer, unpublished / –written]
Approximating the minimum cut graph up to a constant factor is fixed-parameter tractable (in g).