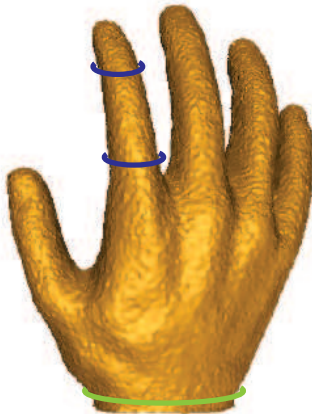


# Computing Optimal Homotopies

Erin Wolf Chambers  
Saint Louis University  
erin.chambers@slu.edu

# Motivation: Measuring Similarity Between Curves

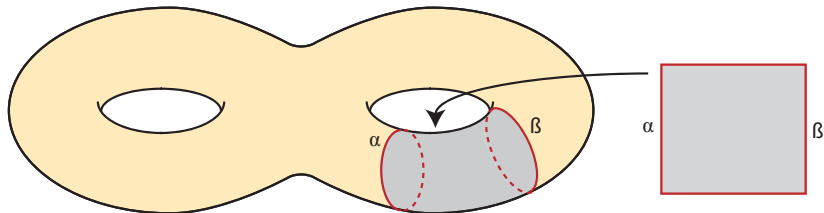
How can we tell when two cycles or curves are similar to each other?



# Recall: Homotopy

## Definition

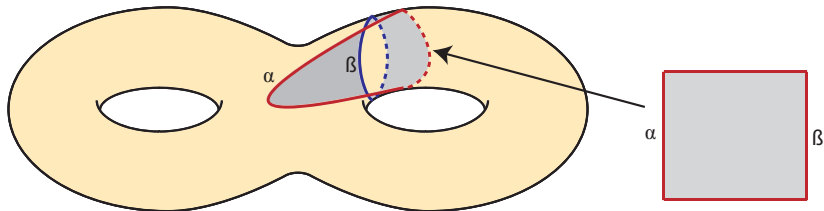
A **homotopy** is a continuous deformation of one path to another. More formally, a homotopy between two curves  $\alpha$  and  $\beta$  on a surface  $M$  is a continuous function  $H : [0, 1] \times [0, 1] \rightarrow M$  such that  $H(\cdot, 0) = \alpha(\cdot)$  and  $H(\cdot, 1) = \beta(\cdot)$ .



# Recall: Homotopy

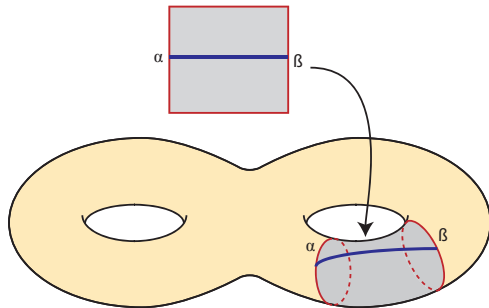
## Definition

A **homotopy** is a continuous deformation of one path to another. More formally, a homotopy between two curves  $\alpha$  and  $\beta$  on a surface  $M$  is a continuous function  $H : [0, 1] \times [0, 1] \rightarrow M$  such that  $H(\cdot, 0) = \alpha(\cdot)$  and  $H(\cdot, 1) = \beta(\cdot)$ .



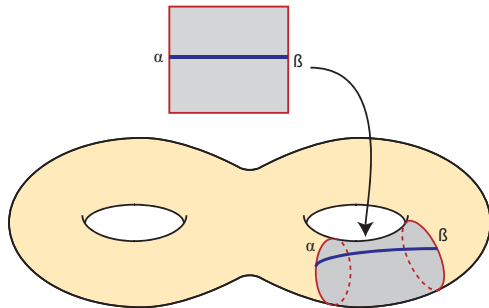
# Recall: Homotopic Fréchet Distance

Homotopic Fréchet distance generalizes the Fréchet distance, but adds the constraint that the curves must be homotopic, and the leashes must move continuously in the ambient space.



# Recall: Homotopic Fréchet Distance

Homotopic Fréchet distance generalizes the Fréchet distance, but adds the constraint that the curves must be homotopic, and the leashes must move continuously in the ambient space.

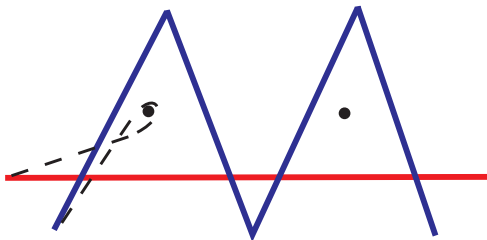


(We could just have really called this the *width* of the homotopy.)

# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{ \sup \{ |H(\cdot, t)| \mid t \in [0, 1] \} \}$$

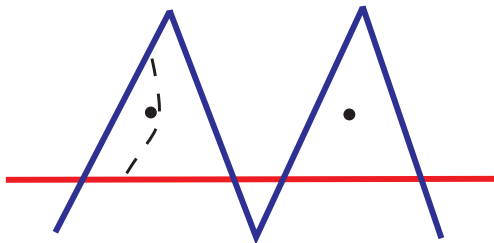


Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).

# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{ \sup \{ |H(\cdot, t)| \mid t \in [0, 1] \} \}$$



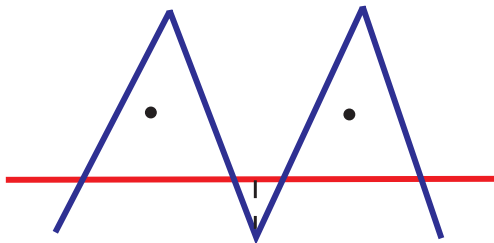
Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).



# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{\sup\{|H(\cdot, t)| \mid t \in [0, 1]\}\}$$

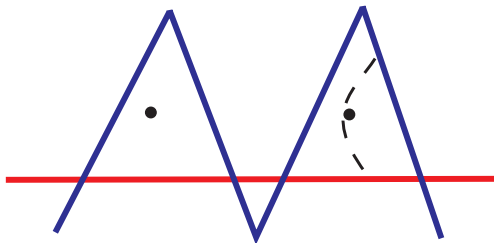


Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).

# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{ \sup \{ |H(\cdot, t)| \mid t \in [0, 1] \} \}$$

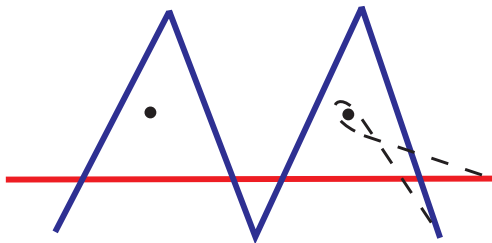


Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).

# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{ \sup \{ |H(\cdot, t)| \mid t \in [0, 1] \} \}$$

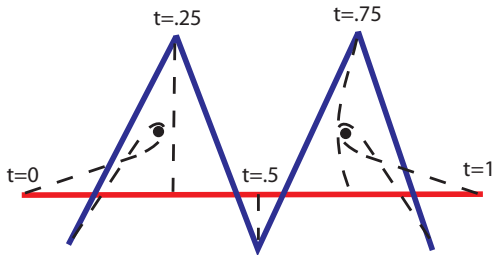


Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).

# Homotopic Fréchet Distance

The homotopic Fréchet distance is the length of the shortest leash we can use for our homotopy. Formally,

$$d_F(\gamma_1, \gamma_2) = \inf_{\text{homotopies } H} \{ \sup \{ |H(\cdot, t)| \mid t \in [0, 1] \} \}$$



Yesterday, we saw a polynomial algorithm to solve this in the plane (minus obstacles).

## Definition

An *isotopy* is a homotopy  $H$  such that for each fixed time  $t$ ,  $H(x, t)$  is a homeomorphism.

## Definition

An *isotopy* is a homotopy  $H$  such that for each fixed time  $t$ ,  $H(x, t)$  is a homeomorphism.

A *homeomorphism* is a function which is a continuous bijection where the inverse is also continuous. In our setting, this will mean that every intermediate curve in the homotopy must also have an image that is simple.

# Isotopic Fréchet Distance

In [C.-Ju-Letscher 2009], we introduced the idea of isotopic Fréchet distance:

$$\mathcal{I}(A, B) = \inf_{\substack{h : M \times I \rightarrow M \\ h(\cdot, t) \text{ homeomorphism} \\ h(x, 0) = x \ \forall x \in X \\ h(A, 1) = B}} \max_{x \in X} \text{len} h(x, \cdot)$$

In other words, what's the longest trajectory in an ambient isotopy?

# Isotopic Fréchet Distance

In [C.-Ju-Letscher 2009], we introduced the idea of isotopic Fréchet distance:

$$\mathcal{I}(A, B) = \inf_{\substack{h : M \times I \rightarrow M \\ h(\cdot, t) \text{ homeomorphism} \\ h(x, 0) = x \ \forall x \in X \\ h(A, 1) = B}} \max_{x \in X} \text{len} h(x, \cdot)$$

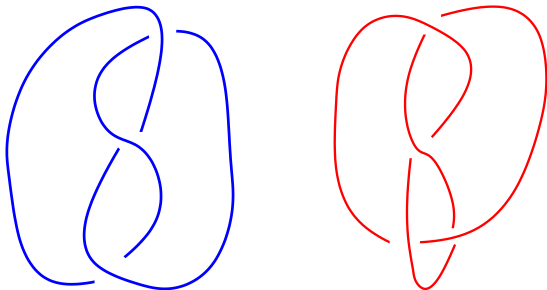
In other words, what's the longest trajectory in an ambient isotopy?

Note the difference with homotopy height: there, the intermediate curves stayed simple, but here, we want the leashes to form an isotopy as well.



# Isotopic Fréchet Distance

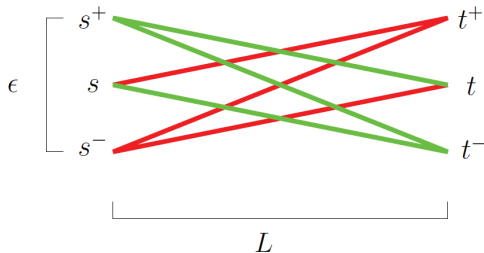
If  $A$  and  $B$  are not ambiently isotopic then  $\mathcal{I}(A, B) = \infty$ .



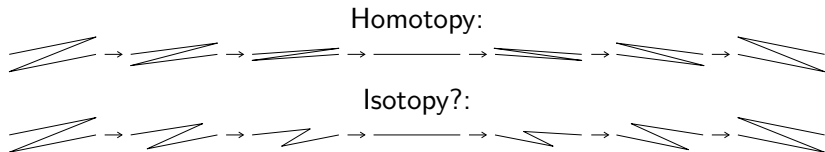
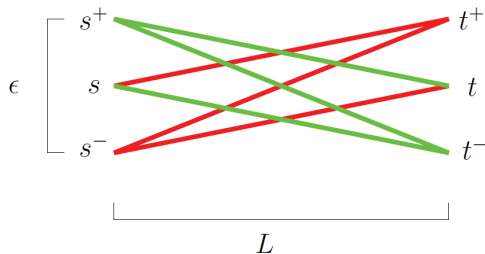
# Homotopic versus Isotopic Fréchet Distance

**Proposition** For any  $L > 0$  and  $\epsilon \in (0, L/2)$  there exists a pair of curves  $C_1, C_2 \in \mathbb{R}^2$  with

$$\begin{aligned}\mathcal{F}(C_1, C_2) = \mathcal{H}(C_1, C_2) &= \epsilon \\ \mathcal{I}(C_1, C_2) &\geq \frac{2}{9}L\end{aligned}$$

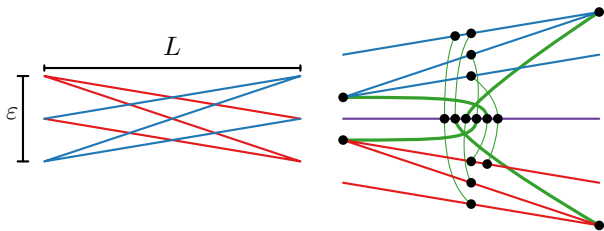


# The best homotopy versus an isotopy



# The best homotopy versus an isotopy

Actually, the best isotopy is even more complicated! The prior picture gave a distance of  $\sqrt{L^2 + \epsilon^2}$ . This was off by a factor of roughly 2 [Buchin-C.-Ophelders-Speckmann 2017]:



# An easy case: monotone curves

I have no idea how to compute monotone Fréchet isotopies (with no obstacles at all), or even any upper bound on the number of “bends” in the longest leash in an optimal one.

Motives the question: when do isotopies with no bends exist?

# An easy case: monotone curves

I have no idea how to compute monotone Fréchet isotopies (with no obstacles at all), or even any upper bound on the number of “bends” in the longest leash in an optimal one.

Motives the question: when do isotopies with no bends exist?

## Theorem

*If the curves  $A$  and  $B$  are  $x$ -monotone, then the isotopic Fréchet distance is equal to the Fréchet distance.*

*Sketch:* Since the curves are  $x$ -monotone, a natural choice of optimal Fréchet leashes (where each is a shortest path), parameterized along the leash as  $(1 - s)A(t) + sB(t)$  for  $s \in [0, 1]$ , will trace an ambient isotopy.

## Another restricted case

Consider “horizontally short” isotopies to a monotone curve: ignore the  $y$ -coordinate, so input curve lies in a small region  $\mathbb{R} \times [0, \epsilon]$  for some small  $\epsilon$ .

## Another restricted case

Consider “horizontally short” isotopies to a monotone curve: ignore the  $y$ -coordinate, so input curve lies in a small region  $\mathbb{R} \times [0, \epsilon]$  for some small  $\epsilon$ .



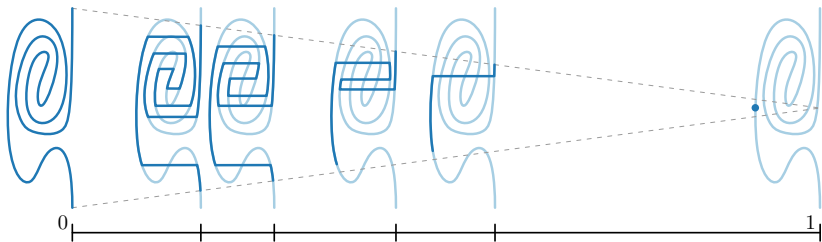
Not so easy even in this simple setting...



# Horizontal Fréchet isotopies

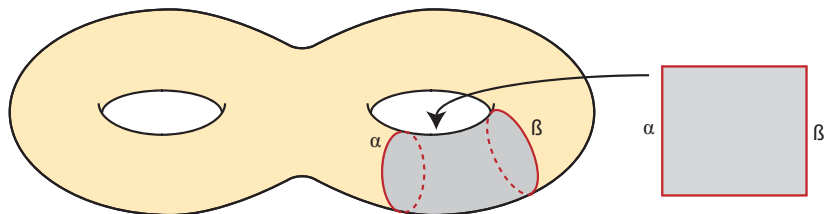
We (in full version of EuroCG paper) can compute optimal isotopic Fréchet to monotone curves, if you only consider the horizontal length of the leash.

Key idea of algorithm: move local minima up and local maxima down, until you can cancel a pair:



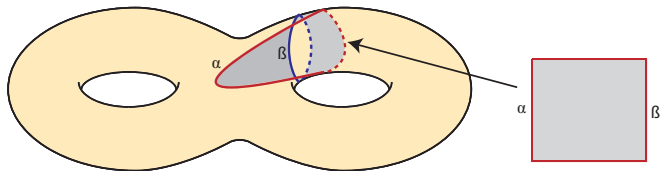
# Area of a homotopy

Recall that instead of focusing on the length or width, we can also examine the total area swept by a homotopy or homology.



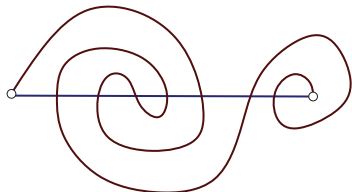
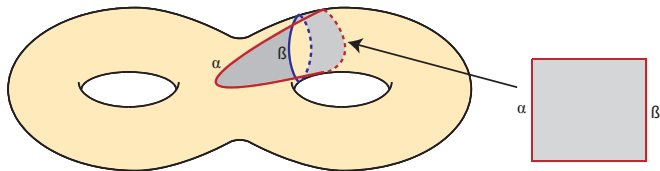
# Computing homotopy area

Surprisingly, this measure is much more tractable than any other measure based on homotopy, even for non-disjoint curves.



# Computing homotopy area

Surprisingly, this measure is much more tractable than any other measure based on homotopy, even for non-disjoint curves.



We'll start in the plane, and then generalize to surfaces...

# Definition: Homotopy area

More formally, given a homotopy  $H$ , the area of  $H$  is defined as:

$$\text{Area}(H) = \int_{s \in [0,1]} \int_{t \in [0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| ds dt$$

# Definition: Homotopy area

More formally, given a homotopy  $H$ , the area of  $H$  is defined as:

$$\text{Area}(H) = \int_{s \in [0,1]} \int_{t \in [0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| ds dt$$

We are then interested in the smallest such value:  $\inf_H \text{Area}(H)$ .

# Definition: Homotopy area

More formally, given a homotopy  $H$ , the area of  $H$  is defined as:

$$\text{Area}(H) = \int_{s \in [0,1]} \int_{t \in [0,1]} \left| \frac{dH}{ds} \times \frac{dH}{dt} \right| ds dt$$

We are then interested in the smallest such value:  $\inf_H \text{Area}(H)$ .

Note that in generally, this is an improper integral, and the value for any  $H$  is not necessarily even finite.

# Douglas and Rado's work

Douglas and Rado (1930's) were the first to consider this problem, as a variant of Plateau's problem (1847) dealing with soap bubbles and minimal surfaces.

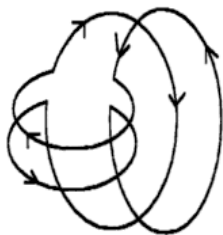


Fig. 4.1

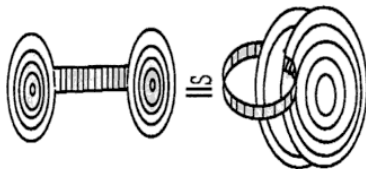


Fig. 4.2

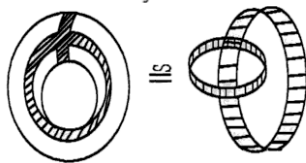


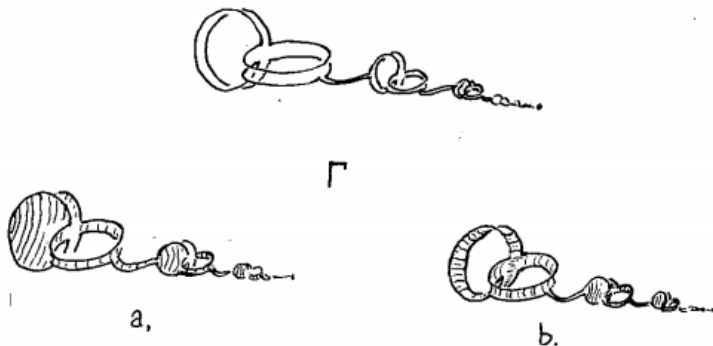
Fig. 4.3

[Minimal sub manifolds and related topics, Y. L. Xin]



# Realizing the minimum area

There is an additional problem in that to find the infimum, we might have a pathological case where a sequence of good  $H$ 's converge to something that is not even continuous.



[Lectures on Minimal Submanifolds, H. B. Lawson]

# Douglas' theorem

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

## Theorem

Let  $\gamma$  be a finite Jordan curve in  $\mathbb{R}^n$ . Then there exists a continuous map  $\Gamma : \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\} \rightarrow \mathbb{R}^n$  such that:

- 1  $\Gamma$  maps the boundary of the disk monotonically onto  $\gamma$ .
- 2  $\Gamma$  is harmonic and almost conformal
- 3  $\Gamma$  realizes the infimum of all areas

They developed a restricted version using Dirichlet integrals (or energy integrals) which allow control over the parameterizations of the minimal surfaces. These integrals not only minimize area, but also ensure (almost) conformal parameterizations in the space.

## Theorem

Let  $\gamma$  be a finite Jordan curve in  $\mathbb{R}^n$ . Then there exists a continuous map  $\Gamma : \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\} \rightarrow \mathbb{R}^n$  such that:

- 1  $\Gamma$  maps the boundary of the disk monotonically onto  $\gamma$ .
- 2  $\Gamma$  is harmonic and almost conformal
- 3  $\Gamma$  realizes the infimum of all areas

(Well, I'm hiding a few details about the Dirichlet integrals here...)

# Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in  $\mathbb{R}^2$  or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that  $H$  is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).

# Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in  $\mathbb{R}^2$  or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that  $H$  is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).
- We also assume the homotopy is monotone along the boundary of the domain and is regular on the interior (meaning intermediate curves are “kink-free”).

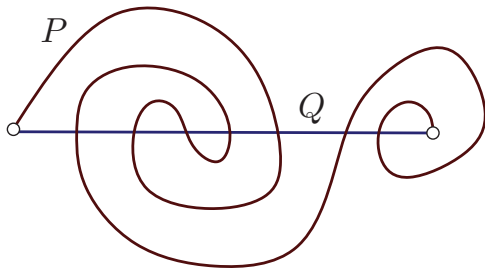
# Necessary assumptions

In [C-Wang 2013], we consider a much simpler setting - we are either in  $\mathbb{R}^2$  or a piecewise linear surface. However, we do need some assumptions in order for the minimum area homotopy to exist.

- We must assume that  $H$  is continuous and piecewise differentiable (so it is differentiable everywhere except at a finite set of points and arcs).
- We also assume the homotopy is monotone along the boundary of the domain and is regular on the interior (meaning intermediate curves are “kink-free”).
- Finally, we will assume our input curves (on  $M$ ) are simple and have a finite number of piecewise analytic components. (In practice, they will simply be PL curves.)

# Algorithm in the plane

In the plane, we consider the decomposition of the plane given by the union of the two curves.

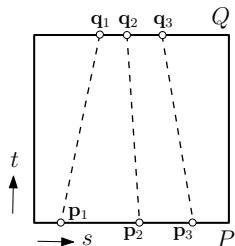
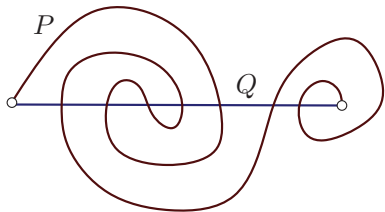


(I'm drawing continuous curves here for simplicity, but think of these as PL when we get to the running time.)



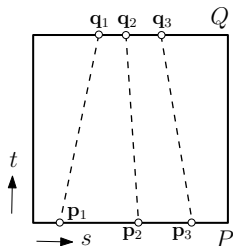
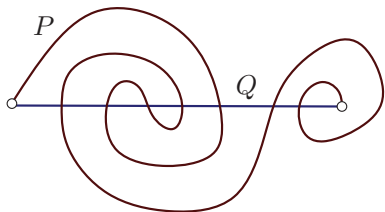
# Anchor points

Note that any vertex of intersection could either be fixed throughout the homotopy (we call this an *anchor point*) or could be moved by the homotopy.



# Anchor points

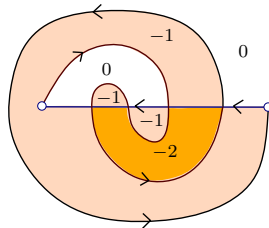
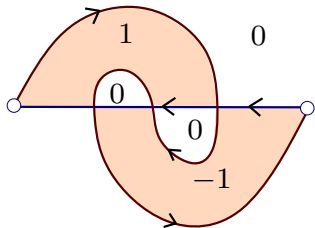
Note that any vertex of intersection could either be fixed throughout the homotopy (we call this an *anchor point*) or could be moved by the homotopy.



We prove that the ordering of the anchor points along the two curves  $P$  and  $Q$  will be the identical, and in between anchor points, we prove that the homotopy will always move locally forward.

# Winding numbers

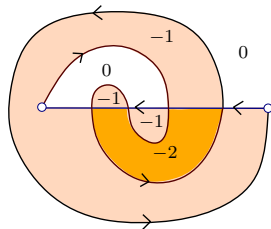
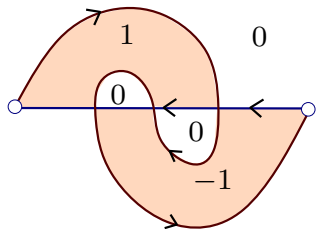
The *winding number* of a closed curve  $\gamma$  with respect to a point  $x$ ,  $wn(x; \gamma)$  is the number of times that curve travels counterclockwise around the point.



# Using the winding number

## Lemma

*Any homotopy with no anchor points will have consistent winding numbers (all non-negative or all non-positive).*

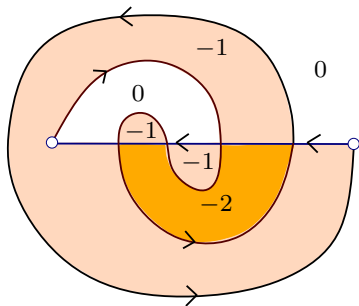


# Calculating with no anchor points

## Lemma

If  $P \circ Q$  has consistent winding numbers, then:

$$\inf_H \text{Area}(H) = \int_{\mathbb{R}^2} |\text{wn}(x; P \circ Q)| dx$$



# The algorithm: dynamic programming

Our algorithm now proceeds quite simply.

# The algorithm: dynamic programming

Our algorithm now proceeds quite simply.

- We can compute the winding number of each planar region. If all are non-positive or non-negative, then we simply sum the areas of each region with multiplicity given by the winding number.

# The algorithm: dynamic programming

Our algorithm now proceeds quite simply.

- We can compute the winding number of each planar region. If all are non-positive or non-negative, then we simply sum the areas of each region with multiplicity given by the winding number.
- If the numbers are not consistent, then we know there is at least one anchor point. Since the order of the anchor points along each curve is the same, we can enumerate all the possible sets of anchor points, and in between the anchor points compute the winding numbers again.



# Running time in the plane

Let  $I$  be the number of intersections and  $n$  be the complexity of the input curves.

# Running time in the plane

Let  $I$  be the number of intersections and  $n$  be the complexity of the input curves.

We give an algorithm that can be implemented in  $O(I^2n)$  time using dynamic programming, which simply builds up the sets of anchor points iteratively and uses previous solutions to speed up future computation.

# Running time in the plane

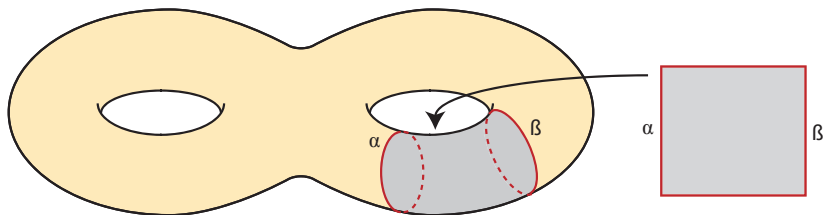
Let  $I$  be the number of intersections and  $n$  be the complexity of the input curves.

We give an algorithm that can be implemented in  $O(I^2 n)$  time using dynamic programming, which simply builds up the sets of anchor points iteratively and uses previous solutions to speed up future computation.

However, this can be improved to  $O(I^2 \log I)$  time with  $O(I \log I + n)$  preprocessing if we are more careful about how we compute the winding numbers.

# Homotopy area on a surface

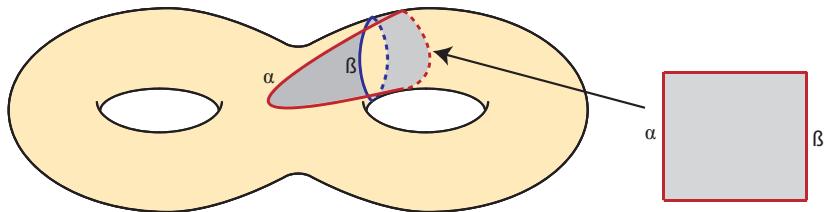
Our paper [C.-Wang] also considers the algorithm for surfaces, which builds upon the planar algorithm.



Consider two homotopic curves on a triangulated surface  $M$  with positive genus.

# Homotopy area on a surface

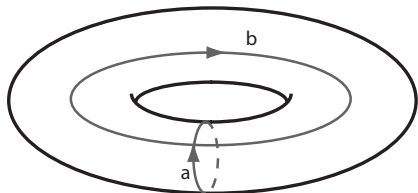
Our paper [C.-Wang] also considers the algorithm for surfaces, which builds upon the planar algorithm.



Consider two homotopic curves on a triangulated surface  $M$  with positive genus.

# The universal cover

Let  $U(M)$  be the universal covering space of  $M$ . This is a simply connected (i.e. planar) domain, along with an associated map  $\phi : U(M) \rightarrow M$  which is continuous, surjective, and a local homeomorphism.



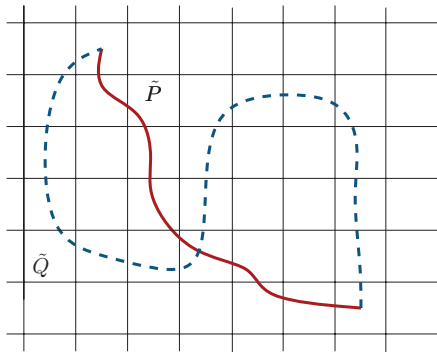
	$(-2, 2)$			$(2, 2)$	
		$(-1, 1)$	$(0, 1)$	$(1, 1)$	
		$(-1, 0)$	$(0, 0)$	$(1, 0)$	
			$(0, -1)$		
		$(-1, -1)$	$(0, -1)$	$(1, -1)$	
	$(-2, -2)$	$(-1, -2)$	$(0, -2)$	$(1, -2)$	$(2, -2)$

Arrows labeled 'a' and 'b' indicate the mapping of paths from the torus to the grid. Path 'a' is represented by vertical arrows pointing up from  $(-1, 0)$  to  $(-1, 1)$  and from  $(0, -1)$  to  $(0, 0)$ . Path 'b' is represented by horizontal arrows pointing right from  $(-1, 0)$  to  $(0, 0)$  and from  $(0, 1)$  to  $(1, 1)$ .

# Lifting $P$ and $Q$

If we fix a lift for the endpoints of  $P$  and  $Q$  in the universal cover  $U(M)$ , then  $P \circ Q$  lifts to a unique closed curve in  $U(M)$ .

Therefore, any homotopy between  $P$  and  $Q$  on  $M$  will correspond to a homotopy between their lifts in  $U(M)$  with the same area.



# Homotopy area on surfaces

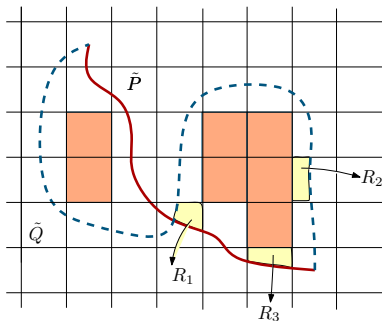
We construct a portion of the universal cover which contains the lifts of  $P$  and  $Q$  as well as the regions inside their concatenation.



# Homotopy area on surfaces

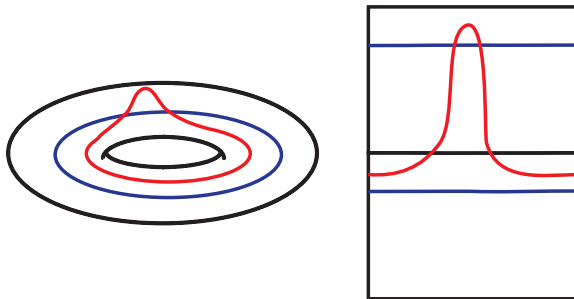
We construct a portion of the universal cover which contains the lifts of  $P$  and  $Q$  as well as the regions inside their concatenation.

We then use our planar algorithm in  $U(M)$ , since similar results about the winding number will hold. Since we can simplify much of the interior of the regions in our representation, the total running time here is  $O(gK \log K + l^2 \log l + ln)$ .



# Open question: cycles on surfaces

We don't know how to compute homotopy area for cycles on surfaces, however:



# More recent algorithms for homotopy area

In our paper, we require the two input curves to be simple (although they can intersect each other), and the homotopy must be regular. There has been more recent work to compute the best area homotopy that contracts an arbitrary curve to a point.

This can be used to compute similarity of two non-simple loops as well: simply concatenate the two curves (perhaps by adding connecting curve).

Nie 2014 considers the weighted cancellation norm, and shows that the word problem in this setting can be solved in polynomial time via dynamic programming:

**Definition 2.6.** The weighted cancellation norm of a word  $w = l_1 \cdots l_n$  over a symmetric set without identity  $S$  with a weight function  $\text{wt}$  is defined as

$$\|w\| := \min_{\mathcal{F}} \sum_i \text{wt}(l_i)$$

where  $\mathcal{F}$  ranges over all foldings of  $w$  and  $i$  ranges over all unpaired positions in  $\mathcal{F}$ .

Nie 2014 considers the weighted cancellation norm, and shows that the word problem in this setting can be solved in polynomial time via dynamic programming:

**Definition 2.6.** The weighted cancellation norm of a word  $w = l_1 \cdots l_n$  over a symmetric set without identity  $S$  with a weight function  $\text{wt}$  is defined as

$$\|w\| := \min_{\mathcal{F}} \sum_i \text{wt}(l_i)$$

where  $\mathcal{F}$  ranges over all foldings of  $w$  and  $i$  ranges over all unpaired positions in  $\mathcal{F}$ .

He then builds the fundamental group from the plane minus the curve, and weights each generator by the area of that cycle. Since this gives a weighted cancellation norm and distance, the optimal area homotopy can be calculated.

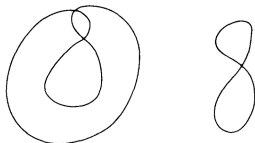
## Second approach

Fasy-Karakoc-Wenk 2016 consider a different approach which is more geometric: They consider self-overlapping curves - curves that are the boundary of an immersed disk - and show that they can do the same winding number calculation to find optimal area homotopies.

## Second approach

Fasy-Karakoc-Wenk 2016 consider a different approach which is more geometric: They consider self-overlapping curves - curves that are the boundary of an immersed disk - and show that they can do the same winding number calculation to find optimal area homotopies.

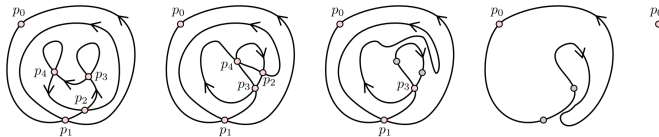
Self overlapping curves were considered previously [Shor-Van Wyk 1992], who gave a polynomial time algorithm to recognize them. (Exactly  $O(n^3)$  assuming generic point intersections.)



Shor-Van Wyk

## Second approach (cont)

For non-self-overlapping curves, they show that a minimum area homotopy can be obtained by contracting a sequence of self-overlapping subcurves, each based at intersection points of the curve:



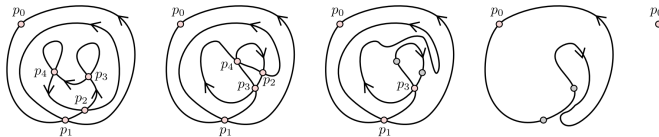
■ **Figure 1** A minimum homotopy is given as a sequence of homotopy moves. The initial curve is self-overlapping.

Fasy-Selcuk-Wenk 2017



## Second approach (cont)

For non-self-overlapping curves, they show that a minimum area homotopy can be obtained by contracting a sequence of self-overlapping subcurves, each based at intersection points of the curve:



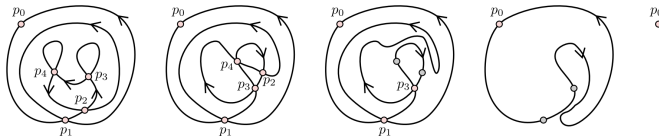
**Figure 1** A minimum homotopy is given as a sequence of homotopy moves. The initial curve is self-overlapping.

Fasy-Selcuk-Wenk 2017

Final running time is exponential, although they are working on dynamic programming techniques to speed this up.

## Second approach (cont)

For non-self-overlapping curves, they show that a minimum area homotopy can be obtained by contracting a sequence of self-overlapping subcurves, each based at intersection points of the curve:



**Figure 1** A minimum homotopy is given as a sequence of homotopy moves. The initial curve is self-overlapping.

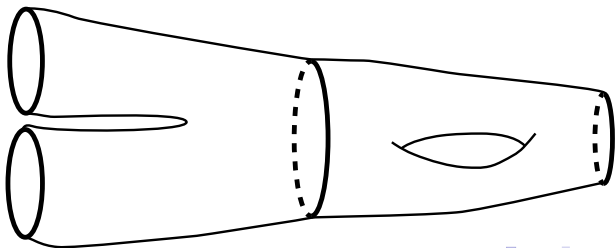
Fasy-Selcuk-Wenk 2017

Final running time is exponential, although they are working on dynamic programming techniques to speed this up.

They also have a working implementation of their approach!  
(Demo time...)

# Using homology?

- Homology is a coarser invariant than homotopy - all homotopies produce homologies, but not all homologies come from homotopies.
- In general, much more tractable - reduces to a linear algebra problem, and software is widely available and highly optimized.
- Potentially much wider applications: works in arbitrary dimensions.
- However, MUCH less intuitive or clear what you are measuring, and not as connected to flow!



# Defining homology

Let me take a minute to actually define it more formally:

## Definition

Given any simplicial complex, the set of  $k$ -chains is just the vector space of linear combinations of  $k$ -simplices, with coefficients from some group or field.

# Defining homology

Let me take a minute to actually define it more formally:

## Definition

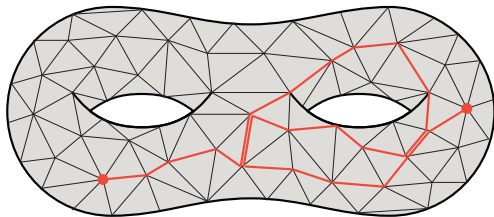
Given any simplicial complex, the set of  $k$ -chains is just the vector space of linear combinations of  $k$ -simplices, with coefficients from some group or field.

Coefficients in this vector space can be from any group or field. So  $k$ -chains are just formal sums of simplices, which form a linear algebra structure.

(Computer scientists are usually comfortable with this, since a 1-chain is just a walk in a graph.)

# K-chains: an example

Triangulation of a 2-manifold: If we use  $F_2$  coefficients on a 2-complex, we get sums of edges, where repeated edges cancel.

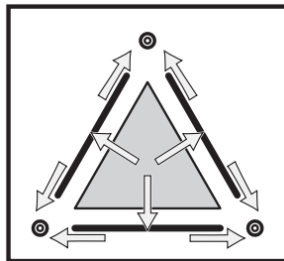


So the set of 1-chains with coefficients from  $F_2$  on a surface are the same as the set of even subgraphs.

# Boundary maps

The boundary map  $\delta_k$  on  $k$ -chains simply takes any simplex to the sum of the  $(k - 1)$ -chains that bound it.

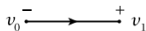
If coefficients are from  $F_2$  then there is no real idea of direction or orientation: an edge is either present or it is not.



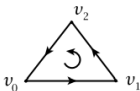
Ghrist 2014

# More boundary maps

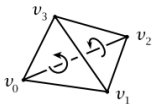
If linear combinations don't come from  $F_2$  then things get a bit more complex:



$$\partial[v_0, v_1] = [v_1] - [v_0]$$



$$\partial[v_0, v_1, v_2] = [v_1, v_2] - [v_0, v_2] + [v_0, v_1]$$



$$\begin{aligned} \partial[v_0, v_1, v_2, v_3] = & [v_1, v_2, v_3] - [v_0, v_2, v_3] \\ & + [v_0, v_1, v_3] - [v_0, v_1, v_2] \end{aligned}$$

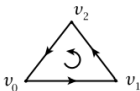


# More boundary maps

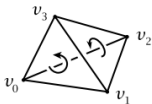
If linear combinations don't come from  $F_2$  then things get a bit more complex:

$$v_0^- \longrightarrow v_1^+$$

$$\partial[v_0, v_1] = [v_1] - [v_0]$$



$$\partial[v_0, v_1, v_2] = [v_1, v_2] - [v_0, v_2] + [v_0, v_1]$$



$$\begin{aligned} \partial[v_0, v_1, v_2, v_3] = & [v_1, v_2, v_3] - [v_0, v_2, v_3] \\ & + [v_0, v_1, v_3] - [v_0, v_1, v_2] \end{aligned}$$

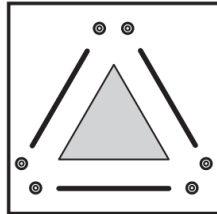
Still - at its heart, this is just linear algebra! (Plus something like the right hand rule, if you remember physics.)

# Boundaries on chains

We can think of the boundary maps on the chain complexes in series:  $C_d \rightarrow C_{d-1} \rightarrow \dots \rightarrow C_1 \rightarrow C_0$ . This is called the chain complex of  $M$ .

These maps and how they treat the chain complexes are the key to homology.

Aside: It's worth noting here that the boundary of a boundary is always empty.



# Boundaries and cycles

The image of an element of  $C_{k-1}$  when the map  $\delta_k : C_k \rightarrow C_{k-1}$  is applied is called a *boundary*.

# Boundaries and cycles

The image of an element of  $C_{k-1}$  when the map  $\delta_k : C_k \rightarrow C_{k-1}$  is applied is called a *boundary*.

Then  $B_k = \text{im } \delta_{k+1}$  is called the boundary group - or  $B_k(C, F_2)$ , for coefficients from  $F_2$ .

# Boundaries and cycles

The image of an element of  $C_{k-1}$  when the map  $\delta_k : C_k \rightarrow C_{k-1}$  is applied is called a *boundary*.

Then  $B_k = \text{im } \delta_{k+1}$  is called the boundary group - or  $B_k(C, F_2)$ , for coefficients from  $F_2$ .

A cycle is any element with empty boundary; these are in the kernel of  $\delta_k$ , since they are sent to nothing under the map.

Think cycles in graph theory - these are precisely 1-chains that have no boundary

# Boundaries and cycles

The image of an element of  $C_{k-1}$  when the map  $\delta_k : C_k \rightarrow C_{k-1}$  is applied is called a *boundary*.

Then  $B_k = \text{im } \delta_{k+1}$  is called the boundary group - or  $B_k(C, F_2)$ , for coefficients from  $F_2$ .

A cycle is any element with empty boundary; these are in the kernel of  $\delta_k$ , since they are sent to nothing under the map.

Think cycles in graph theory - these are precisely 1-chains that have no boundary

Similarly, a union of triangles that form a sphere is a 2-chain that has no boundary, but a disk would have boundary.

# Boundaries and cycles

The image of an element of  $C_{k-1}$  when the map  $\delta_k : C_k \rightarrow C_{k-1}$  is applied is called a *boundary*.

Then  $B_k = \text{im } \delta_{k+1}$  is called the boundary group - or  $B_k(C, F_2)$ , for coefficients from  $F_2$ .

A cycle is any element with empty boundary; these are in the kernel of  $\delta_k$ , since they are sent to nothing under the map.

Think cycles in graph theory - these are precisely 1-chains that have no boundary

Similarly, a union of triangles that form a sphere is a 2-chain that has no boundary, but a disk would have boundary.

Then  $Z_k = \ker \delta_k$  is the set of these cycles.

So: we have a chain complex:

$$C_d \rightarrow C_{d-1} \rightarrow \dots \rightarrow C_1 \rightarrow C_0$$

And two subgroups inside each  $C_k$ :  $Z_k = \ker \delta_k$  and  $B_k = \text{im } \delta_{k+1}$ .



So: we have a chain complex:

$$C_d \rightarrow C_{d-1} \rightarrow \dots \rightarrow C_1 \rightarrow C_0$$

And two subgroups inside each  $C_k$ :  $Z_k = \ker \delta_k$  and  $B_k = \text{im } \delta_{k+1}$ .

We then say two chains in  $Z_k$  are *homologous* if they differ by a boundary.

So: we have a chain complex:

$$C_d \rightarrow C_{d-1} \rightarrow \dots \rightarrow C_1 \rightarrow C_0$$

And two subgroups inside each  $C_k$ :  $Z_k = \ker \delta_k$  and  $B_k = \text{im } \delta_{k+1}$ .

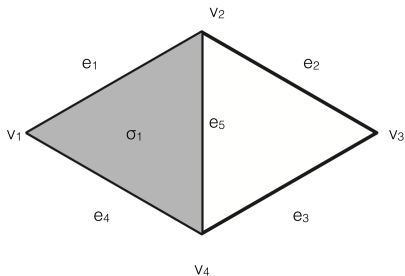
We then say two chains in  $Z_k$  are *homologous* if they differ by a boundary.

In other words,  $H_k = Z_k/B_k$ .

Let's look at what this means...

# A simple complex

Consider a simple complex: a filled in triangle, glued to an empty triangle:



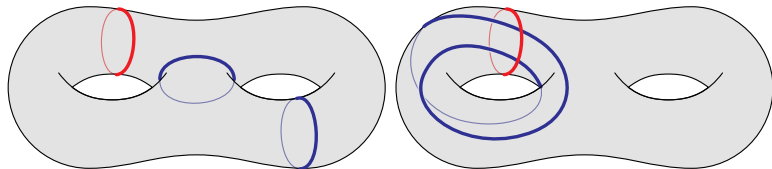
Here,  $e_1 + e_4 + e_5$  is in image  $\delta_2$ , since it is the boundary of triangle  $\sigma_1$ .

So  $e_2 + e_3 + e_5$  is homologous to  $e_1 + e_2 + e_3 + e_4$ . (Note that in this case, they are also homotopic to each other.)

## Another example: surfaces

Recall my example of surfaces from yesterday: we have triangles, edges, and vertices, so we get  $C_2 \rightarrow C_1 \rightarrow C_0$ .

Here, two cycles of edges in  $C_1$  are homologous if they differ by a boundary of some 2d region built from triangles:



Hence, two things are homologous if they form a cut graph - one side of the cut is the “homology”, or union of faces which have these edges as a boundary.

# Computing homology

In order to compute homology groups, we need to realize that the boundary operator is written as a matrix,

If  $C_p = \{\alpha_1, \alpha_2, \dots, \alpha_{n_p}\}$  and  $C_{p-1} = \{\tau_1, \tau_2, \dots, \tau_{n_{p-1}}\}$ , then the boundary map  $\delta_p$  is:

$$M_p = \begin{matrix} & \alpha_1 & \alpha_2 & \dots & \alpha_{n_p} \\ \tau_1 & \left( \begin{matrix} a_1^1 & a_1^2 & \dots & a_1^{n_p} \\ a_2^1 & a_2^2 & \dots & a_2^{n_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \dots & a_{n_{p-1}}^{n_p} \end{matrix} \right) \end{matrix}$$

From Yusu Wang's lecture notes

# Computing homology

In order to compute homology groups, we need to realize that the boundary operator is written as a matrix,

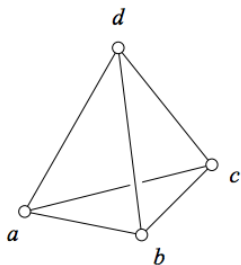
If  $C_p = \{\alpha_1, \alpha_2, \dots, \alpha_{n_p}\}$  and  $C_{p-1} = \{\tau_1, \tau_2, \dots, \tau_{n_{p-1}}\}$ , then the boundary map  $\delta_p$  is:

$$M_p = \begin{matrix} & \alpha_1 & \alpha_2 & \dots & \alpha_{n_p} \\ \tau_1 & a_1^1 & a_1^2 & \dots & a_1^{n_p} \\ \tau_2 & a_2^1 & a_2^2 & \dots & a_2^{n_p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_{n_{p-1}} & a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \dots & a_{n_{p-1}}^{n_p} \end{matrix}$$

From Yusu Wang's lecture notes

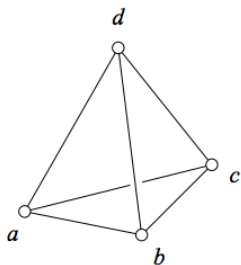
This is why computing ranks of homology groups and testing if two things are homologous essentially boils down to matrix multiplication.

# Another example: a tetrahedra



$$M_2 = \begin{array}{c} abc \quad abd \quad acd \quad bcd \\ \begin{array}{l} ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{array} \end{array} \begin{pmatrix} 1 & 1 & & \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & & & 1 \\ & 1 & & 1 \\ & & 1 & 1 \end{pmatrix}$$

## Another example: a tetrahedra



$$M_2 = \begin{array}{c} abc \quad abd \quad acd \quad bcd \\ \begin{array}{l} ab \\ ac \\ ad \\ bc \\ bd \\ cd \end{array} \end{array} \begin{pmatrix} 1 & 1 & & \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & & & 1 \\ & 1 & & 1 \\ & & 1 & 1 \end{pmatrix}$$

The boundary matrix  $M_2$  (for  $F_2$  coefficients) just indicates which edges bound which triangles.

If we want to find the boundary of a collection of 2-faces, we can multiply the vector of 2-faces by this matrix to find the edges that bound it.

So  $\delta_k$  really turns into a simple linear map between vector spaces.



# Larger complexes

Of course, even simple complexes are much larger:

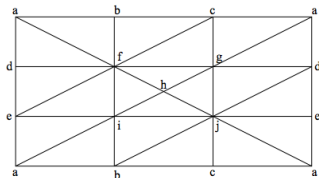
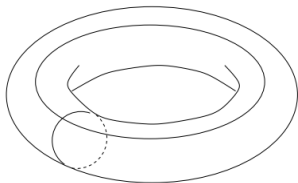


Image from Vanessa Robin's thesis

This triangulation will lead to a dim 2 boundary matrix that is 20 by 30, and a dim 1 that is 30 by 10.

# Larger complexes

Of course, even simple complexes are much larger:

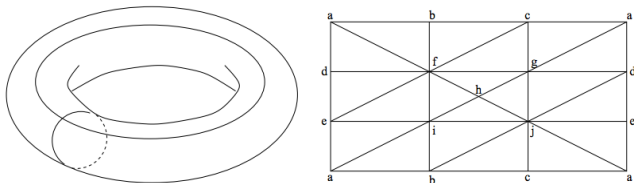


Image from Vanessa Robin's thesis

This triangulation will lead to a dim 2 boundary matrix that is 20 by 30, and a dim 1 that is 30 by 10.

However, easy to calculate with: just import your favorite linear algebra library that can reduce matrices.

Or (better yet) use one of the many that already exist: mapper, dionysus, javaplex, CTL,...

# How to compute homology area

However, I again am interested not just in a homology, but one with minimum area.

(Note: this is joint work with Mikael Vejdemo Johansson, and originally considered in slightly more restricted settings in Dey, Hirani and Krishnamoorthy):

# How to compute homology area

However, I again am interested not just in a homology, but one with minimum area.

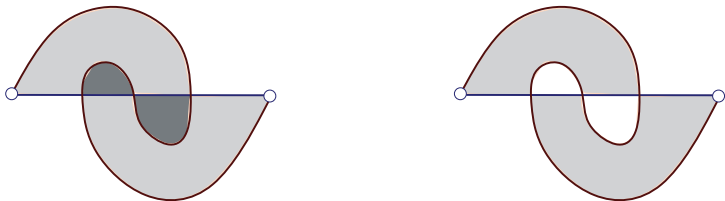
(Note: this is joint work with Mikael Vejdemo Johansson, and originally considered in slightly more restricted settings in Dey, Hirani and Krishnamoorthy):

More formally:

- Given cycles  $\alpha$  and  $\beta$ , try to compute  $z$  such that  $\delta z = \alpha - \beta$ .
- Goal: compute  $z$  with a smallest area. Recall that  $\delta$  is a linear operator (that big matrix), and  $z$  and  $\alpha - \beta$  are vectors.
- Optimization problem is then:  
 $\arg \min_z (\text{area } z)$ , subject to  $dz = \alpha - \beta$ .

# Homology versus homotopy area

Note again that this is NOT the same as homotopy area, at least for  $d \leq 3$ :



For homology area, we just need a collection of faces whose boundary is the curve.

- With area annotations (in a diagonal matrix  $A$ ) for triangulation, optimization turns into:  $\arg \min_z |Az|_1$ , subject to  $dz = \alpha - \beta$ .
- By inverting  $A$ , optimization turns into:  $\arg \min_z |z|_1$ , subject to  $dA^{-1}z = \alpha - \beta$ .
- This is a standard  $L_1$  optimization problem: well studied in compressed sensing.

- $L_2$ -minimal solutions to linear equations have analytic solutions and are much faster.
- $L_2$ -minimal is not the same as  $L_1$ -minimal, however.
- It comes close(ish) though:  
 $|x|_2 \leq |x|_1 \leq \sqrt{n}|x|_2$   
for  $n =$  number of triangles in the triangulation.

# An $L_2$ approximation

- Given the speed, we try to solve  $dz = \alpha - \beta$  subject to minimizing  $z^T Az$ .
- We repeat the trick of baking in the triangle area as a diagonal matrix:

$$\zeta = \sqrt{A}z$$

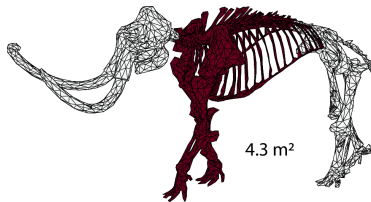
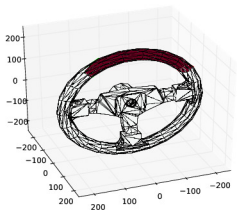
$$\zeta^T \zeta = (\sqrt{A}z)^T \sqrt{A}z = z^T A z$$

- So we try to compute  $\arg \min_{\zeta} |\zeta|_2$  subject to  $d(\sqrt{A})^{-1}\zeta = \alpha - \beta$
- Closed form solution: write  $W = d(\sqrt{A})^{-1}$ . Then  $\zeta = (W^T W)^{-1} W^T (\alpha - \beta)$ . Matrix multiplication time.



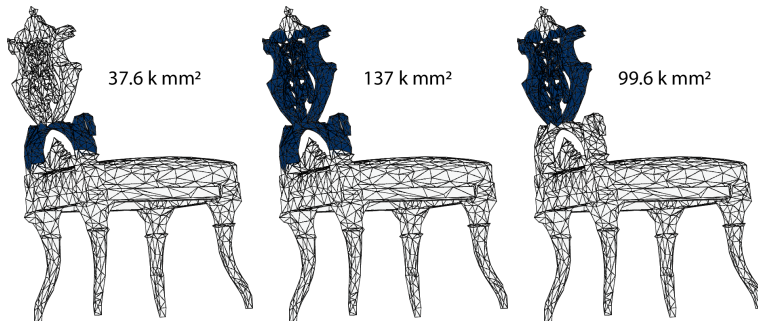
# Final algorithm for homology area

In matrix multiply time, we can compute the best area homology on meshes:

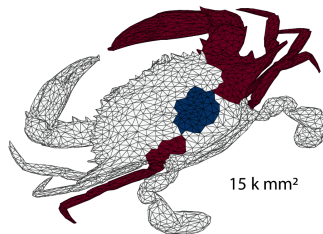
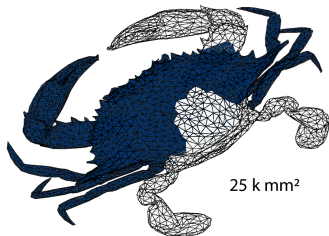


(Python code available on github - see paper for link.)

# Chair model



# Crab model



Thanks for your attention!

Questions?