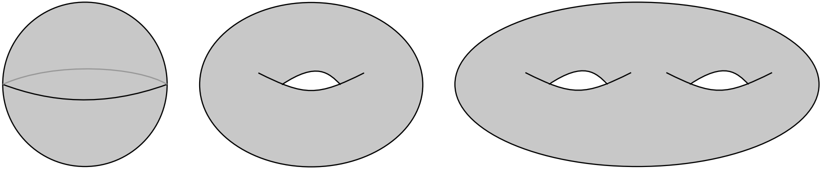
Surfaces and commutators (Geometry REU) Class 1

Alden Walker
(later: Danny Calegari)

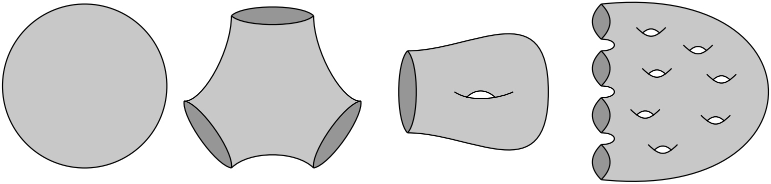
Summer 2013

Surfaces

Some surfaces:



Some surfaces with boundary:

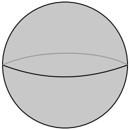


The *genus* is the number of holes.

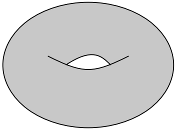
Euler characteristic

Euler characteristic $\chi(S)$ measures the complexity of S.

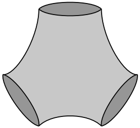
$\chi(S) = 2 - 2(\text{genus}) - (\#\ \text{boundaries})$



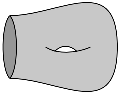
$g = 0$
$\#b = 0$
$\chi = 2$



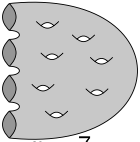
$g = 1$
$\#b = 0$
$\chi = 0$



$g = 0$
$\#b = 3$
$\chi = -1$



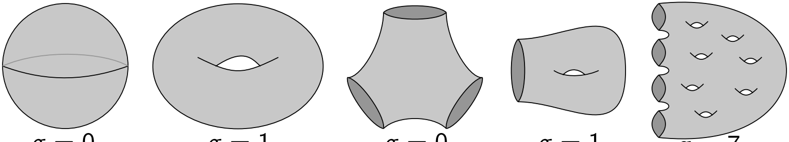
$g = 1$
$\#b = 1$
$\chi = -1$



$g = 7$
$\#b = 4$
$\chi = -16$

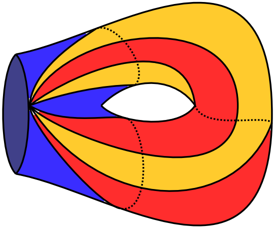
Euler characteristic

Euler characteristic $\chi(S)$ measures the complexity of S.
$\chi(S) = 2 - 2(\text{genus}) - (\#\text{ boundaries})$



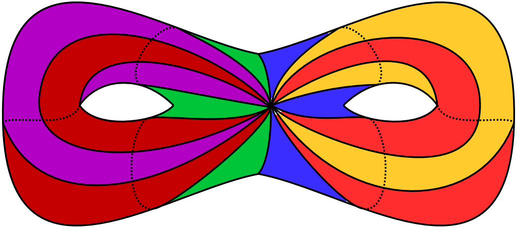
$g = 0$	$g = 1$	$g = 0$	$g = 1$	$g = 7$
$\#b = 0$	$\#b = 0$	$\#b = 3$	$\#b = 1$	$\#b = 4$
$\chi = 2$	$\chi = 0$	$\chi = -1$	$\chi = -1$	$\chi = -16$

In (any) triangulation of S, $\chi(S) = V - E + F$:

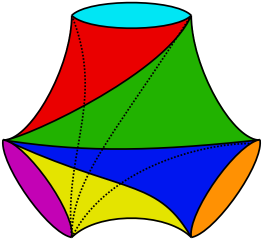


$V - E + F = 1 - 5 + 3 = -1$

More triangulations

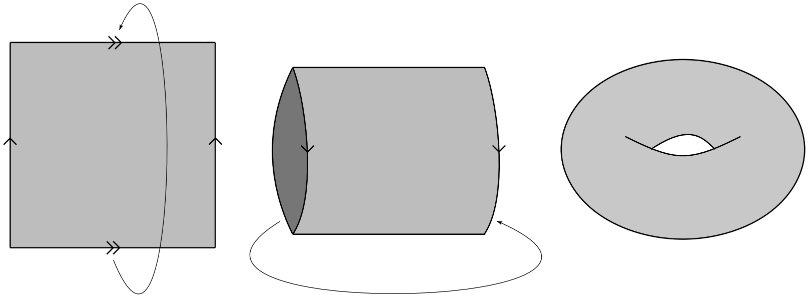


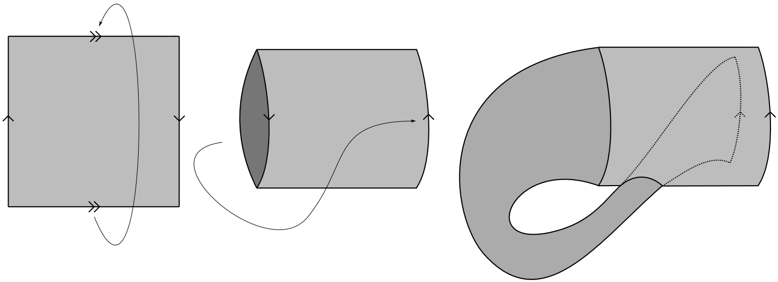
$$-2 = 2 - 2g - p = V - E + F = 1 - 9 + 6 = -2$$



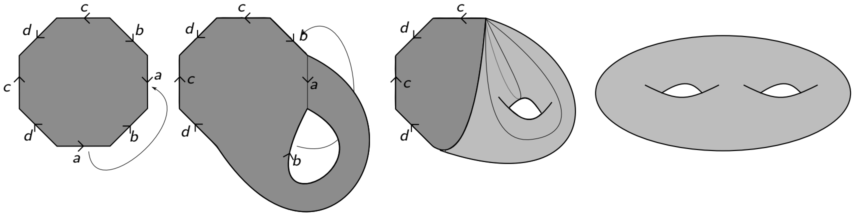
$$-1 = 2 - 2g - p = V - E + F = 6 - 15 + 8 = -1$$

Gluing polygons to get surfaces

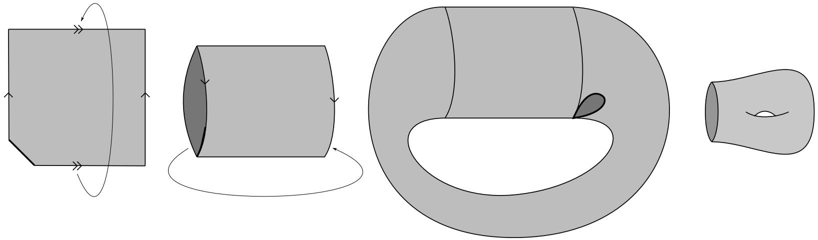


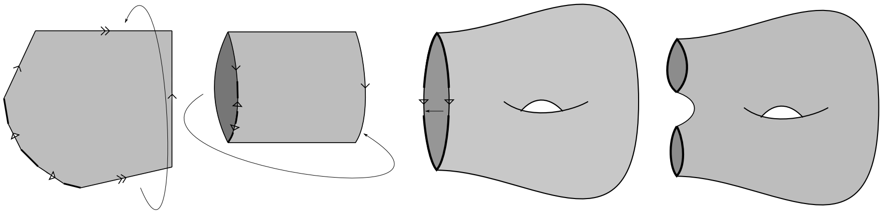


Gluing polygons to get surfaces

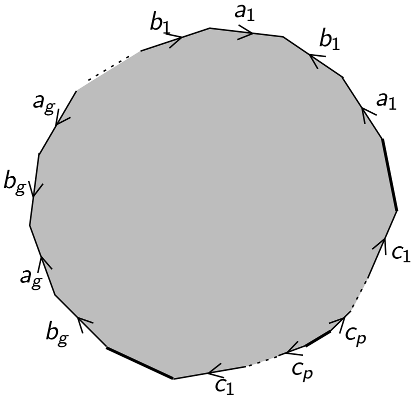


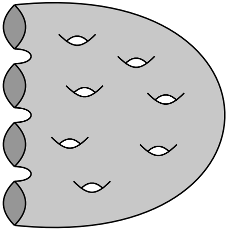
Gluing polygons to get surfaces with boundary





Gluing polygons to get surfaces with boundary

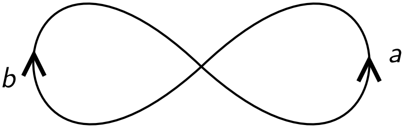




Gluing produces a genus g surface with p boundaries.

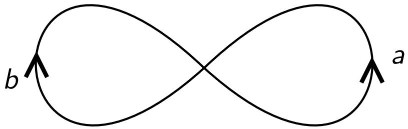
Free groups

A wedge X of two circles:



Each loop has a direction labeled, a means forward, A means backward. Loops are recorded by a sequence of letters, e.g. $abAB$. The special loop e is the "loop" which is constant at the base point.

Free groups

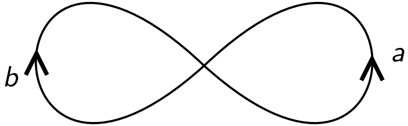


Concatenation: if x and y are loops, then xy (the *product*) is the loop x followed by y.

Adjacent opposite letters cancel, since $aA = bB = e$, so e.g.

$$(abABA)(abAAb) = abAAAb$$

Free groups



Let F be the set of all loops. The set F with the product operation is the *free group*.

Facts:

- We define $x^n = xxx \cdots x$.
- For all $x \in F$, there is $x^{-1} \in F$ so that $xx^{-1} = x^{-1}x = e$. To get x^{-1}, reverse and swap letters e.g.

$$(abAABB)^{-1} = bbaaBA$$

Commutators

Let x and y be loops. The *commutator* of x and y is the loop

$$[x, y] = xyx^{-1}y^{-1}$$

so

$$[abAAB, bA] = (abAAB)(bA)(baaBA)(aB) = abAAAbaaBB$$

The set of all products of commutators is called the *commutator subgroup*, denoted $[F, F]$.

Example: the loop $abbaBAAbABBa$ is in the commutator subgroup, because it is a product of commutators:

$$[ab, ba][ba, Ab] = (abbaBAAB)(baAbABBa) = abbaBAAbABBa$$

Commutators

Some random facts:

1. $[x, y]^{-1} = [y, x]$. We check:

$$[x, y][y, x] = xyx^{-1}y^{-1}yxy^{-1}x^{-1} = e$$

2. $z[x, y]z^{-1} = [zxz^{-1}, zyz^{-1}]$, since:

$$[zxz^{-1}, zyz^{-1}] = zxz^{-1}zyz^{-1}zx^{-1}z^{-1}zy^{-1}z^{-1} = zxyx^{-1}y^{-1}z^{-1}$$

Commutators

Let $|x|_z$ denote the number of occurrences of the letter z in the word x, so $|abAABB|_a = 1$, and $|abAABB|_B = 2$.

Lemma

For a loop $x \in F$, we have $x \in [F, F]$ iff $|x|_a = |x|_A$ and $|x|_b = |x|_B$.

So $abABAbaB \in [F, F]$.

The commutator subgroup

Lemma

For a loop $x \in F$, we have $x \in [F, F]$ iff $|x|_a = |x|_A$ and $|x|_b = |x|_B$.

Proof.

By induction on the length of x. WLOG, assume the last letter is a. Find an A in x, and write $x = sAta$, where s and t are words. Then

$$\begin{aligned}(sAta)[(ta)^{-1}, a] &= (sAta)(At^{-1})(a)(ta)(A) \\ &= st\end{aligned}$$

So

$$sAta = st([(ta)^{-1}, a])^{-1} = st[a, (ta)^{-1}]$$

The word st is shorter than x, and still has matching numbers of a, A and b, B. By induction, st is a product of commutators, so x is.

□

The commutator subgroup

Question: if $|x|_a = |x|_A$ and $|x|_b = |x|_B$, then $x \in [F, F]$, so x is a product of commutators. What is the smallest number?

I.e. what is that smallest k so that there exist y_i and z_i so that

$$x = [y_1, z_1][y_2, z_2] \cdots [y_k, z_k]$$

We call this k the *commutator length* $\mathrm{cl}(x)$ of x.

Commutator length

Example (Culler)

$$[x, y]^3 = [xyx^{-1}, y^{-1}xyx^{-2}][y^{-1}xy, yy]$$

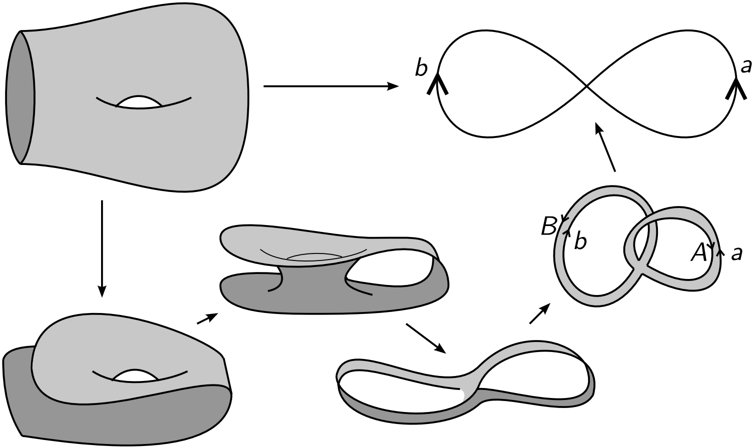
So, $[x, y]^3$ can obviously be written as a product of three commutators, so $\mathrm{cl}([x, y]^3) \leq 3$. But it can secretly be written as a product of two, so $\mathrm{cl}([x, y]^3) \leq 2$.

Finding $\mathrm{cl}(x)$ is a hard problem that can be solved using surfaces.

Surface maps into a free group

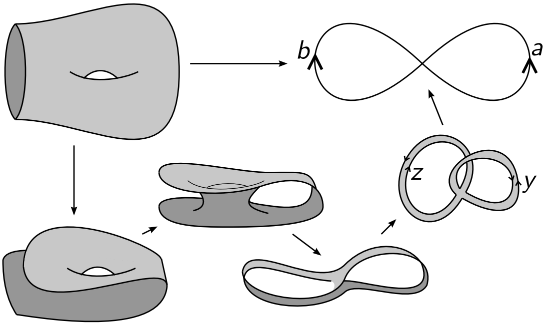
Find a surface in X with boundary loop $abAB$. How can a surface map to a wedge of two loops?

Stretch the surface to make it skinny:



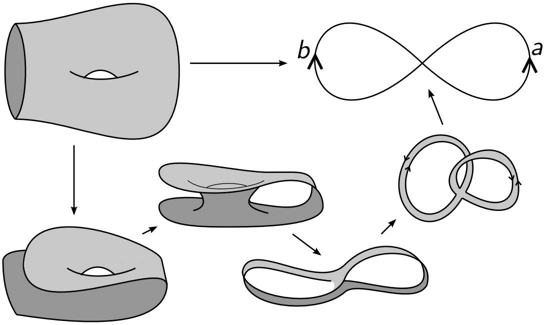
Surface maps into a free group

Suppose we have a once-punctured torus mapping into a free group so that the boundary maps to a loop $x \in F$.



Consider the two loops y and z in the surface. They map into loops y and z in F, and the boundary of the surface maps to $x = [y, z]$.

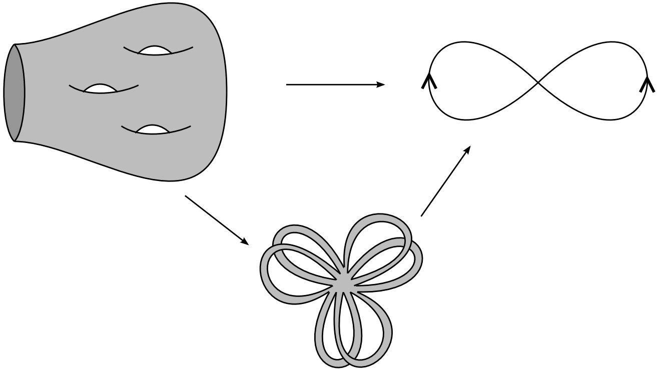
Surface maps into a free group



Lemma

For $x \in F$, x is a commutator iff there is a map of a once-punctured torus into F so that the boundary maps to x.

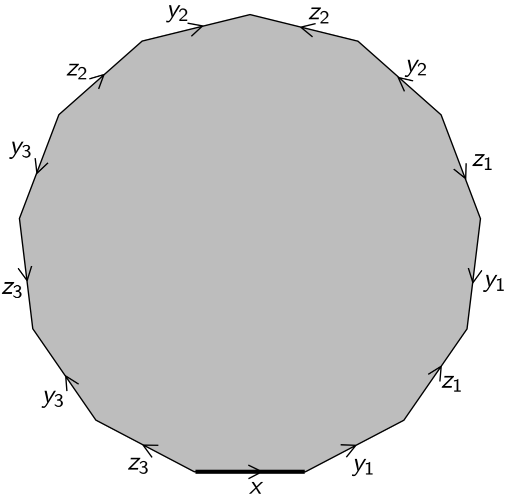
Surface maps into a free group



Lemma

In general, if S is a surface of genus g with one boundary component, then a map $S \to F$ taking the boundary of S to x is equivalent to an expression of x as a product of g commutators.

Surface maps into a free group



We can also see this by looking at a gluing polygon. Here

$$x = [y_1, z_1][y_2, z_2][y_3, z_3]$$

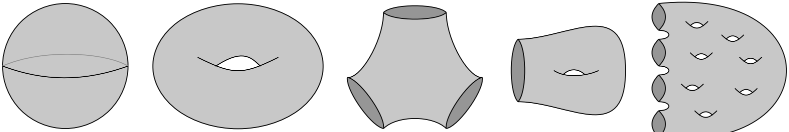
Surfaces and commutators (Geometry REU) Class 2

Alden Walker
(Later: Danny Calegari)

Summer 2013

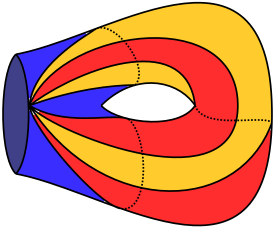
Review

Euler characteristic $\chi(S)$ measures the complexity of S.
$\chi(S) = 2 - 2(\text{genus}) - (\# \text{ boundaries})$



$g = 0$	$g = 1$	$g = 0$	$g = 1$	$g = 7$
$\#b = 0$	$\#b = 0$	$\#b = 3$	$\#b = 1$	$\#b = 4$
$\chi = 2$	$\chi = 0$	$\chi = -1$	$\chi = -1$	$\chi = -16$

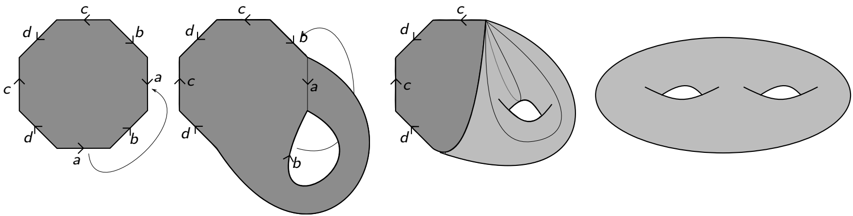
In (any) triangulation of S, $\chi(S) = V - E + F$:



$V - E + F = 1 - 5 + 3 = -1$

Review

There are several ways to get a surface, for example by gluing together triangles, or by gluing up a polygon:



Review

The *free group* generated by a and b is be the set F of all reduced words in the letters a, b, A, B, with the product operation of concatenation (and canceling adjacent letter-inverse pairs):

$$(abAAB)(baAb) = abAAb$$

Every word $x \in F$ has a unique *inverse* word x^{-1}, for which $xx^{-1} = x^{-1}x = e$. It is obtained by reversing x and swapping the letter case:

$$(abAAB)^{-1} = baaBA$$

Review

The *commutator* $[x, y]$ of words x and y is defined

$$[x, y] = xyx^{-1}y^{-1}$$

The *commutator subgroup* $[F, F]$ of F is the set of all products of commutators in F.

Example

The loop *abbaBAAbABBa* is in the commutator subgroup, because it is a product of commutators:

$$[ab, ba][ba, Ab] = (abbaBAAB)(baAbABBa) = abbaBAAbABBa$$

Review

Lemma

For a loop $x \in F$, we have $x \in [F, F]$ iff $|x|_a = |x|_A$ and $|x|_b = |x|_B$.

So it is easy to check if x is a product of commutators.

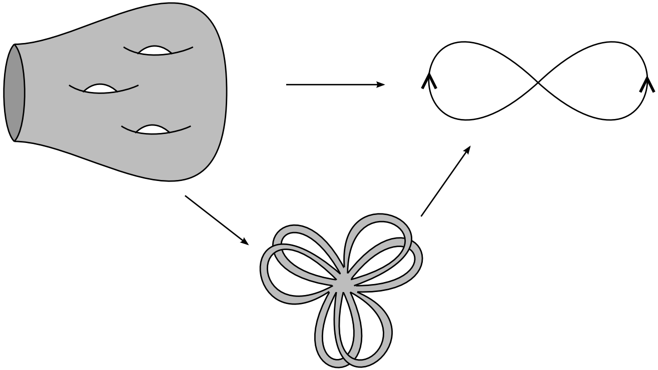
But, it is *hard* to know how many commutators are necessary. If $x \in [F, F]$, we define $\mathrm{cl}(x)$ to be the minimum k so that x is a product of k commutators.

Example (Culler)

$$[x, y]^3 = [xyx^{-1}, y^{-1}xyx^{-2}][y^{-1}xy, yy]$$

Obviously, $\mathrm{cl}([x, y]^3) \leq 3$. But it can secretly be written as a product of two commutators, so $\mathrm{cl}([x, y]^3) \leq 2$.

Review



Lemma

If S is a surface of genus g with one boundary component, then a map $S \to F$ taking the boundary of S to x is equivalent to an expression of x as a product of g commutators.

Finding surface maps

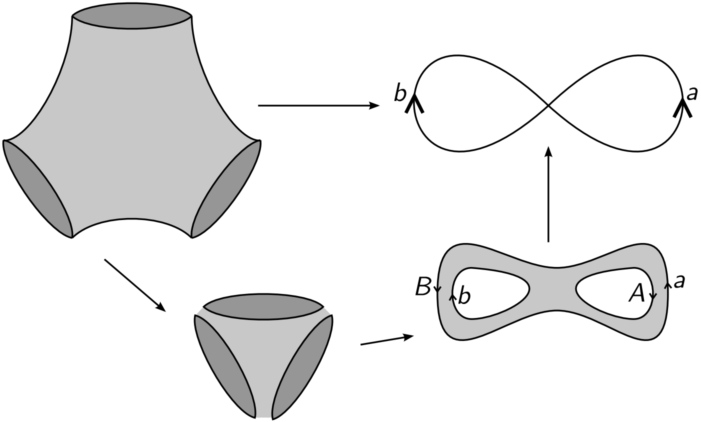
How can we find a map from a surface to a free group?

By building it out of pieces

Let us forget commutators for now and just try to find a surface map with a given boundary.

Finding surface maps

Note we could ask for a surface with multiple boundary loops.
We'll show how to build surfaces with any desired boundaries.

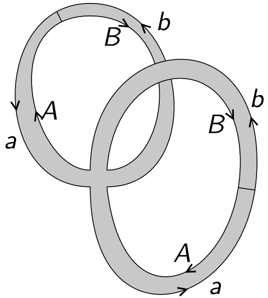


The skinny surface has boundary $aB + b + A$.

Finding surface maps

Any surface can be stretched into a skinny surface (a *fatgraph*). To map a fatgraph into the free group, each junction maps to the basepoint, and each strip follows some loop. Notice one side of each strip is labeled by a generator, and the opposite side by the inverse.

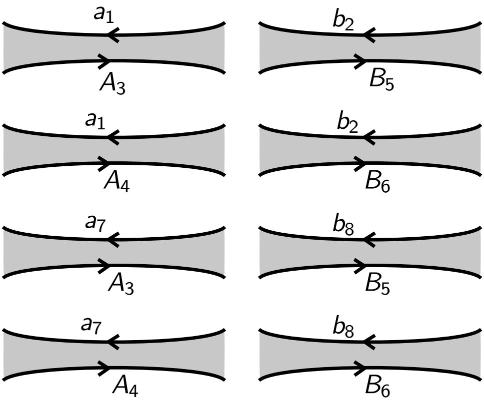
That is, surface maps into a free group are in 1-1 correspondence with *labeled fatgraphs*



This surface map has boundary $[ab, AB] = abABBAba$.

Finding surface maps

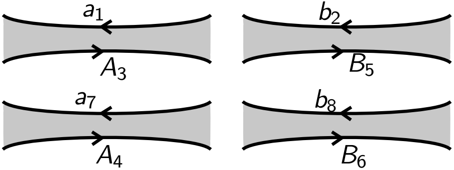
Let us look for a labeled fatgraph with boundary $abAABB + ab$. The strips (*rectangles*) that can occur are labeled with a letter-inverse pair.

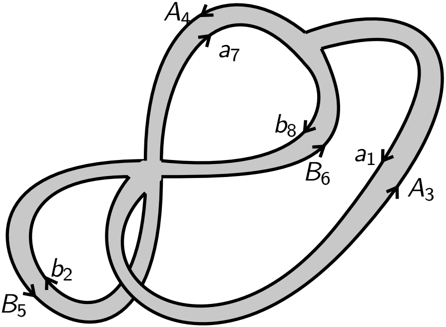


These are all possible strips; the letters are $a_1 b_2 A_3 A_4 B_5 B_6 + a_7 b_8$.

Finding surface maps

Pick rectangles that contain every letter once, and glue up:

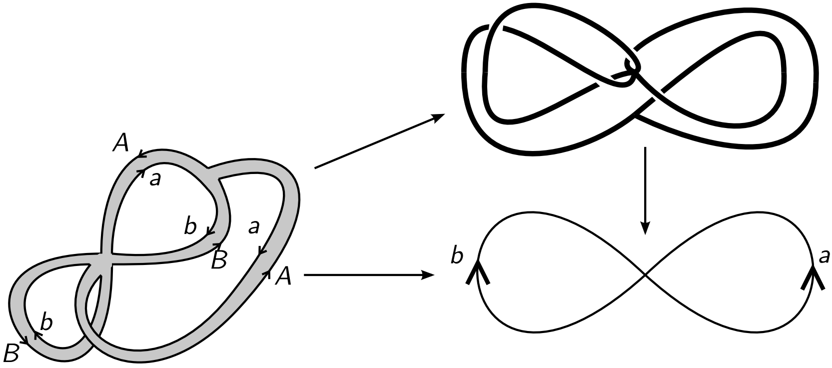




Boundary is $a_1 b_2 A_3 A_4 B_5 B_6 + a_7 b_8$.

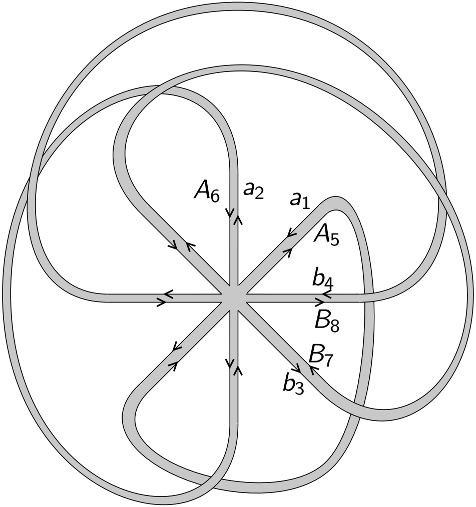
How to map labeled fatgraphs

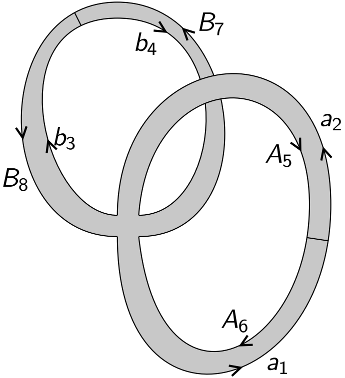
We just built a labeled fatgraph. The labels instruct us how to get a map into the wedge of loops:



Comparing skinny surfaces

There are multiple ways to pair up the letters to get skinny surfaces with a set boundary. Both of these pairings give surfaces with boundary *aabbAABB*.





Comparing fatgraphs/skinny surfaces

Recall that for a surface S, $\chi(S) = 2 - 2g - p$, and $\chi(S) = V - E + F$ for any triangulation. It's hard to immediately see the genus of a fatgraphs, but it is simple to compute χ:

Lemma

For a skinny surface S built out of J junctions and R rectangles, we have $\chi(S) = J - R$.

Proof.

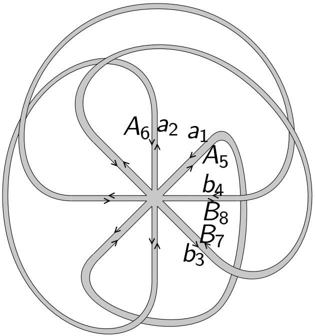
Triangulate S with 2 triangles for every rectangle, and $k - 2$ triangles for every junction with k adjacent rectangles. This triangulation has 2 vertices for every rectangle, 5 edges for every rectangle, and $k - 3$ (extra) edges at the junctions. So

$$\chi = 2R - (5R + \sum_j k_j - 3) + 2R + \sum_j (k_j - 2) = J - R$$

□

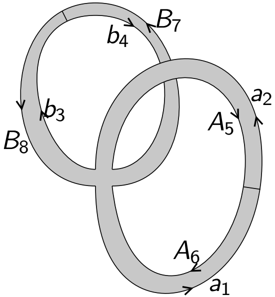
Comparing fatgraphs

Therefore, we can easily compute the genus of these two surfaces



$\chi(S) = 1 - 4 = -3$

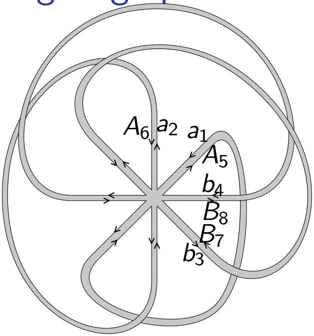
So $g = -((1 - \chi)/2) = 2$



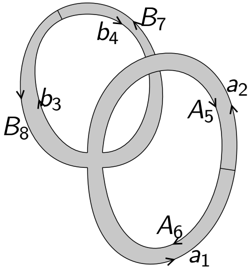
$\chi(S) = 3 - 4 = -1$

So $g = -((1 - \chi)/2) = 1$

Comparing fatgraphs



$\chi(S) = 1 - 4 = -3$
So $g = -((1 - \chi)/2) = 2$



$\chi(S) = 3 - 4 = -1$
So $g = -((1 - \chi)/2) = 1$

The left surface shows that *aabbAABB* can be written as a product of two commutators. The right shows it can be written as a single commutator. (this is pretty obvious, since $[aa, bb] = aabbAABB$). So $\mathrm{cl}(aabbAABB) = 1$.

Commutator length

Algorithm to compute $\text{cl}(x)$:

1. Build all possible rectangles that can occur in a skinny surface with boundary x.
2. Take all possible subcollections of the rectangles so that every letter in x appears exactly once.
3. For each subcollection, glue up the rectangles, and compute the genus of the surface.
4. The smallest possible genus is $\text{cl}(x)$.

Example

Recall, obviously $\text{cl}([a,b]^3) \leq 3$, and being clever, we showed $\text{cl}([a,b]^3) \leq 2$. Doing the algorithm proves that $\text{cl}([a,b]^3) = 2$.

Stable commutator length

For $x \in [F, F]$, we define the *stable commutator length*

$$\mathrm{scl}(x) = \lim_{n \to \infty} \frac{\mathrm{cl}(x^n)}{n}$$

(Fact: this limit exists). We have $\mathrm{cl}(x^n) \leq n\mathrm{cl}(x)$, so $\mathrm{scl}(x) \leq \mathrm{cl}(x)$.

Example

Clearly, $\mathrm{cl}([a, b]) = 1$. But we saw that $\mathrm{cl}([a, b]^3) = 2$, so $\mathrm{cl}([a, b]^{3n}) \leq 2n$, and

$$\mathrm{scl}([a, b]) = \lim_{n \to \infty} \frac{\mathrm{cl}([a, b]^n)}{n} = \lim_{n \to \infty} \frac{\mathrm{cl}([a, b]^{3n})}{3n} \leq \frac{2}{3}$$

Stable commutator length

Why should the limit $\mathrm{scl}(x) = \lim_{n\to\infty} \mathrm{cl}(x^n)/n$ exist?

Lemma

The sequence $\mathrm{cl}(x^n)$ *is subadditive, i.e.*

$$\mathrm{cl}(x^{n+m}) \leq \mathrm{cl}(x^n) + \mathrm{cl}(x^m)$$

Lemma (Fekete)

Let a_n *be any subadditive sequence* ($a_{n+m} \leq a_n + a_m$) *with all* a_n *positive. Then* $\lim_{n\to\infty} \frac{a_n}{n}$ *exists, and* $\lim_{n\to\infty} \frac{a_n}{n} = \inf_n \frac{a_n}{n}$.

Subadditive sequences (an aside)

Lemma (Fekete)

Let a_n be any subadditive sequence ($a_{n+m} \leq a_n + a_m$) with all a_n positive. Then $\lim_{n\to\infty} \frac{a_n}{n}$ *exists, and* $\lim_{n\to\infty} \frac{a_n}{n} = \inf_n \frac{a_n}{n}$.

Proof.

The sequence a_n/n is bounded below by 0, so $L = \inf_n \frac{a_n}{n}$ exists. Given $\epsilon > 0$, pick m so $\frac{a_m}{m} < L + \frac{\epsilon}{2}$ (L is the inf). Let $C = \max_{k<m} a_k$. Pick $N > m$ so that $\frac{C}{N} < \frac{\epsilon}{2}$. Now, given $n > N$, we write $n = qm + r$ for $r < m$ (quotient and remainder), and we have

$$L \leq \frac{a_n}{n} = \frac{a_{qm+r}}{qm+r} \leq \frac{qa_m + a_r}{qm+r} \leq \frac{a_m}{m} + \frac{a_r}{N} < L + \frac{\epsilon}{2} + \frac{\epsilon}{2}$$

So $|\frac{a_n}{n} - L| < \epsilon$, as desired. □

Stable commutator length

Stable commutator length $\mathrm{scl}(x)$ measures how many commutators, on average, are required per copy of x in a large power of x. This can be smaller than $\mathrm{cl}(x)$:

Example

word x	$\mathrm{cl}(x)$	$\mathrm{scl}(x)$
$AbAbaaabAAbbABBBaaBB$	2	1
$bABBaaBAAbbabaBA$	2	3/4
$ABaBAAAbabABABaBAbaabbabAbaBaB$	2	29/24
$babaBBAbaBABABBabABaBAAbaBabbbabAAABaabA$	2	819/619
$[a, b]$	1	1/2
$[a, b]^2$	2	1
$[a, b]^3$	3	3/2
$[a, b]^4$	3	2
$[a, b]^5$	3	5/2
$[a, b]^6$	4	3

From the definition, $\mathrm{scl}([a, b]^n) = n\mathrm{scl}([a, b]) = \frac{n}{2}$. I am not aware of a formula for $\mathrm{cl}([a, b]^n)$.

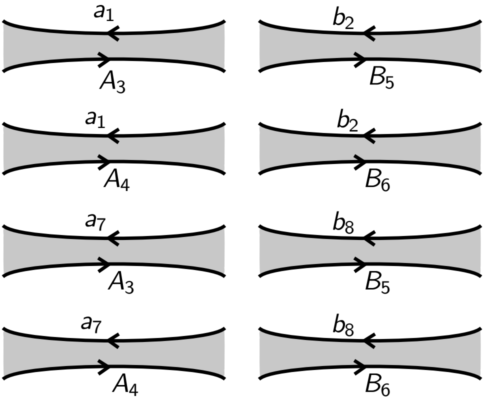
Surfaces and commutators
(Geometry REU)
Class 3

Alden Walker
(Later: Danny Calegari)

Summer 2013

Review (Building surfaces)

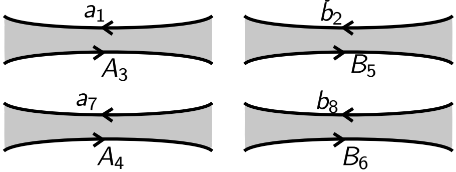
Let us look for a labeled fatgraph with boundary $abAABB + ab$.
We differentiate the letters by writing them $a_1b_2A_3A_4B_5B_6 + a_7b_8$.
The rectangles that can occur are labeled with a letter-inverse pair.

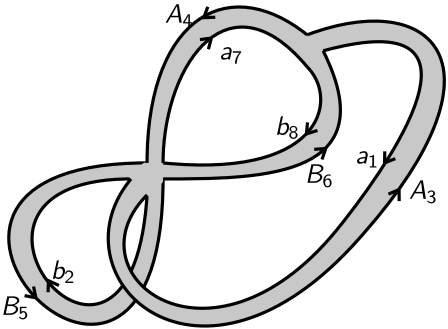


These are all possible rectangles.

Review (Building surfaces)

For every collection of rectangles which contains every letter exactly once, there is a determined way to glue them up to get a labeled fatgraph with the desired boundary.





Review (cl(x) computation)

Algorithm to compute cl(x) (or rather, to compute the smallest genus of a surface with boundary x):

1. Build all possible rectangles that can occur in a skinny surface with boundary x.
2. Take all possible subcollections of the rectangles so that every letter in x appears exactly once.
3. For each subcollection, glue up the rectangles, and compute the genus of the surface.
4. The smallest possible genus is cl(x).

Example

Recall, obviously $\text{cl}([a,b]^3) \leq 3$, and being clever, we showed $\text{cl}([a,b]^3) \leq 2$. Doing the algorithm proves that $\text{cl}([a,b]^3) = 2$.

Clarification

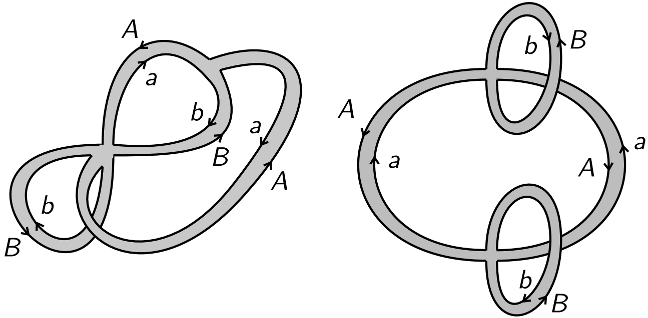
The algorithm solves the problem: given any loops $x, y, z, \ldots$, find the least-genus surface which has boundaries $x, y, z, \ldots$. In the case that you specify a single boundary component x, this finds $\mathrm{cl}(x)$.

If there are more boundaries, it finds the least-genus surface with those boundaries, but the equivalent algebraic statement for multiple boundaries is messy and not so illuminating.

Clarification 2

Each surface can map into a free group in many ways. (For example, every commutator corresponds to a different map of the same once-punctured torus).

Equivalently, there are many labeled fatgraphs which are actually the same underlying surface.



These labeled fatgraphs give two distinct maps into a free group of a genus two surface with two boundaries. On the right, the boundaries are $abAABB + ab$, and on the left, $abAb + ABaB$.

Review (scl)

For $x \in [F, F]$, we define the *stable commutator length*

$$\mathrm{scl}(x) = \lim_{n\to\infty} \frac{\mathrm{cl}(x^n)}{n}$$

(Fact: this limit exists). We have $\mathrm{cl}(x^n) \leq n\mathrm{cl}(x)$, so $\mathrm{scl}(x) \leq \mathrm{cl}(x)$.

Example

Clearly, $\mathrm{cl}([a, b]) = 1$. But we saw that $\mathrm{cl}([a, b]^3) = 2$, so $\mathrm{cl}([a, b]^{3n}) \leq 2n$, and

$$\mathrm{scl}([a, b]) = \lim_{n\to\infty} \frac{\mathrm{cl}([a, b]^n)}{n} = \lim_{n\to\infty} \frac{\mathrm{cl}([a, b]^{3n})}{3n} \leq \frac{2}{3}$$

Review (scl)

Stable commutator length scl(x) measures how many commutators, on average, are required per copy of x in a large power of x. This can be smaller than cl(x):

Example

word x	cl(x)	scl(x)
$AbAbaaabAAbbABBBaaBB$	2	1
$bABBaaBAAbbabaBA$	2	3/4
$ABaBAAAbabABABaBAbaabbabAbaBaB$	2	29/24
$babaBBAbaBABABBabABaBAAbaBabbbabAAABaabA$	2	819/619
$[a,b]$	1	1/2
$[a,b]^2$	2	1
$[a,b]^3$	3	3/2
$[a,b]^4$	3	2
$[a,b]^5$	3	5/2
$[a,b]^6$	4	3

From the definition, $\mathrm{scl}([a,b]^n) = n\mathrm{scl}([a,b]) = \frac{n}{2}$. I am not aware of a formula for $\mathrm{cl}([a,b]^n)$.

Stable commutator length

Amazing fact: it is possible to compute $\mathrm{scl}(x)$. In fact, it is *easier* to compute than $\mathrm{cl}(x)$.

Efficient surface maps

Computing scl is related to finding *efficient* surface maps.

Original question: given $x \in [F, F]$, find the surface map into F whose boundary maps to x with the smallest genus.

Better question: given $x \in [F, F]$, find the most *efficient* surface mapping to F whose boundaries all map to powers of x.

Let us think about the latter question, setting aside scl for the moment.

Efficient surface maps

What does *efficient* mean?

We say a surface S is *admissible* for x if all of the (potentially many) boundaries of S are powers of x. Given an admissible S, we let $n(S)$ be the total number of copies of x appearing in ∂S.

Compute:

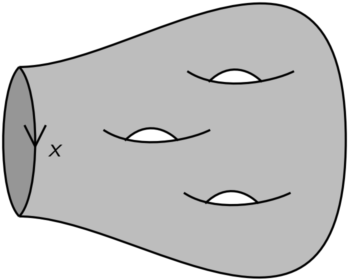
$$\inf_S \frac{-\chi(S)}{2n(S)}$$

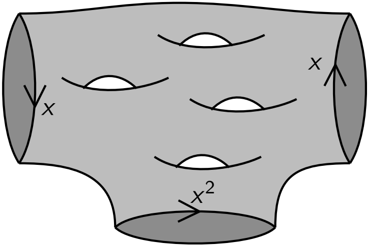
over all admissible surfaces S (there are infinitely many).

If a surface S has small $\frac{-\chi(S)}{2n(S)}$, then it has small *average complexity* per copy of x.

(It's $\chi/2n$ not χ/n because it doesn't really matter and $\chi/2n$ is approximately the genus).

Efficient surfaces





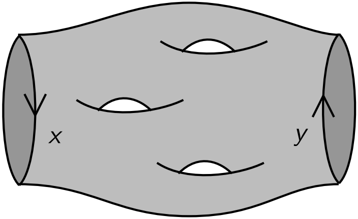
$$\chi(S) = 2 - 2(3) - 1 = -5$$
$$n(S) = 1$$
$$\frac{-\chi(S)}{2n(S)} = \frac{5}{2} = 2.5$$

$$\chi(S) = 2 - 2(4) - 3 = -9$$
$$n(S) = 4$$
$$\frac{-\chi(S)}{2n(S)} = \frac{9}{8} = 1.125$$

So while the surface on the right is more complicated, it is also more *efficient*.

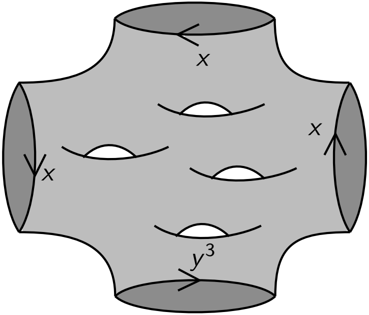
Efficient surfaces

We can search for efficient surfaces not just for a word x, but for several words x, y, z. In this case, we require that every boundary component is a power of x, y, or z, and we require that the total number of copies of each word is the same (and we denote it $n(S)$).



$\chi(S) = -6$, $n(S) = 1$

$\frac{-\chi(S)}{2n(S)} = 3$

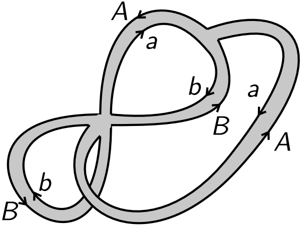


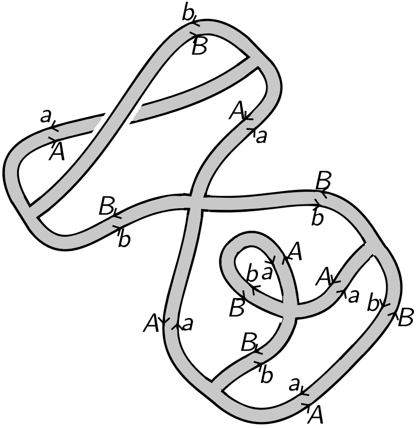
$\chi(S) = -10$, $n(S) = 3$

$\frac{-\chi(S)}{n(S)} = \frac{10}{6} < 3$

Efficient surfaces

Example: The best surface which wraps around $abAABB + ab$ once has $-\chi(S)/2 = 1$. The most *efficient* surface wraps $n = 3$ times around, and has $-\chi(S)/(2n(S)) = 2/3$.





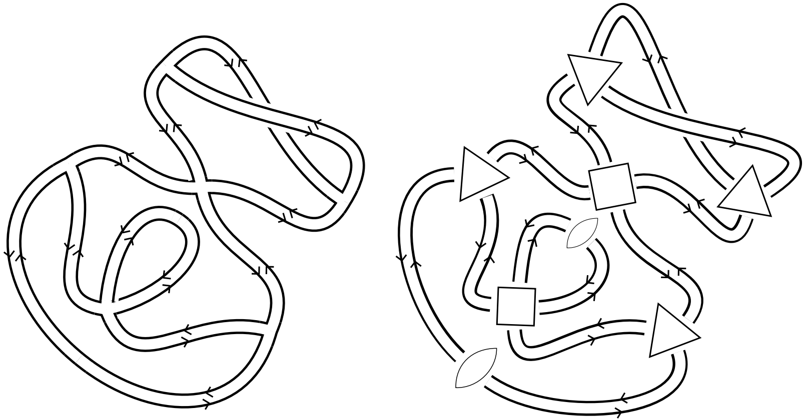
Finding efficient surfaces

Like the algorithm to compute cl, we'll build efficient surfaces out of pieces. It is more complicated this time; we need to record not just the rectangles, but how they attach together.

(If we have more than one copy of a rectangle, it's no longer completely determined how to glue them up).

Finding efficient surfaces

Any labeled fatgraph for $abAABB + ab$ can be broken into pieces:



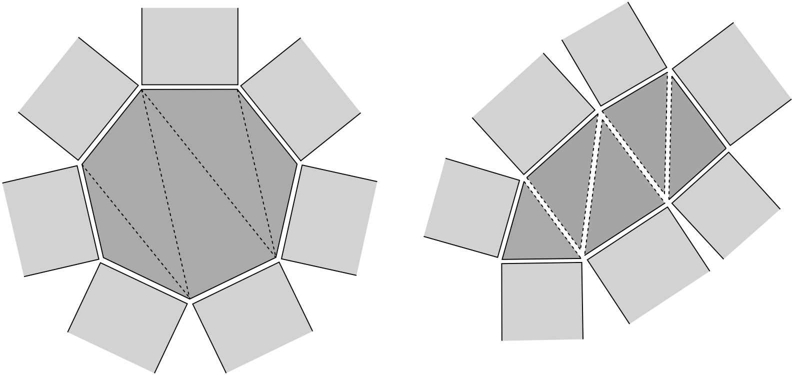
Every rectangle is one of the finitely many possibilities. But we need to understand the junctions, which we'll call *polygons*.

Polygons

The problem: we no longer have a bound on the number of rectangles, so we don't have a bound on the number of sides of each polygon. Hence, there are infinitely many possible types of polygons!

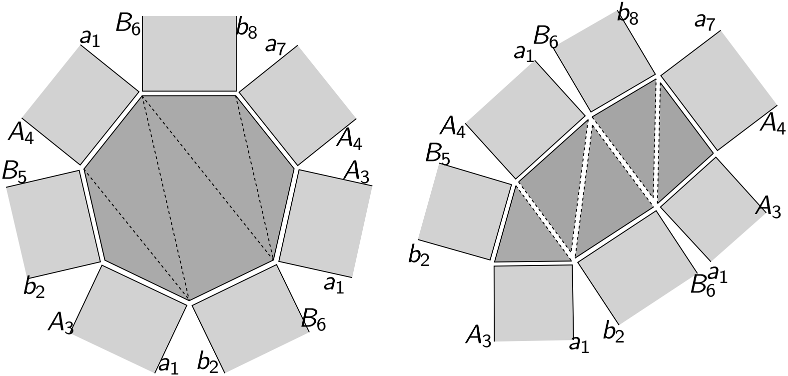
This is bad, because it seems like we don't have a nice finite collection of pieces with which to build surfaces. But there is a trick.

Polygons



The trick: any polygon can be cut into triangles, and there are only finitely many kinds of triangles.

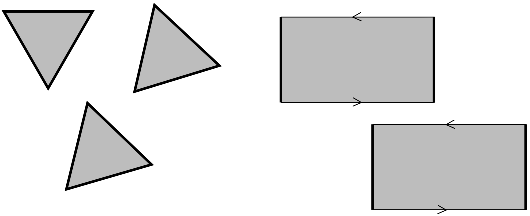
Polygons



Here is a hypothetical polygon in a fatgraph with boundary $a_1 b_2 A_3 A_4 B_5 B_6 + a_7 b_8$. There are many ways to cut it into triangles; we show one of them here. We can record the triangles by recording the incoming and outgoing labels. For example, the lower left triangle could be denoted $((a_1, b_2), (A_4, B_5), (b_2, A_3))$.

Building efficient surfaces

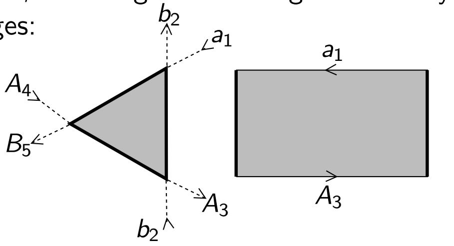
For a specified boundary, the set of all *pieces* (all possible rectangles and all possible triangles in a fatgraph with the given boundary) is finite. Every rectangle has two *edges* to glue, and every triangle has three.

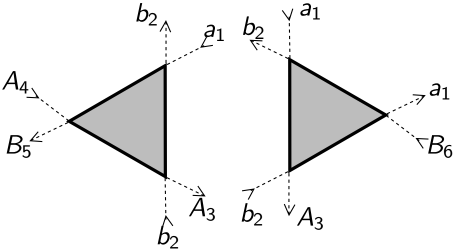


Given two pieces, we can glue them together if they have compatible edges.

Building efficient surfaces

Given two pieces, we can glue them together if they have compatible edges:





E.g. we are allowed to glue rectangle (A_3, a_1) to triangle $((a_1, b_2), (A_4, B_5), (b_2, A_3))$, and we're allowed to glue triangle $((a_1, b_2), (A_4, B_5), (b_2, A_3))$ to triangle $((A_4, B_5), (a_1, b_2), (B_6, a_1))$.

Fatgraphs as vectors

We can denote the edges as pairs of incoming and outgoing labels e.g. $((a_1, b_2), (A_4, B_5))$. Each edge has a matching edge to which it is glued, namely the reverse: we glue $((a_1, b_2), (A_4, B_5))$ to $((A_4, B_5), (a_1, b_2))$.

Given any collection of pieces, if we can glue them up (every edge appears as many times as its mate), then however we glue them up, we'll get a fatgraph with the desired boundary.

We can determine whether the edges match up by looking at linear constraints.

Fatgraphs as vectors

Given a desired boundary x, suppose there are N possible pieces. We record how many of each piece we have in a vector $X \in \mathbb{R}^N$, so X_i denotes how many copies of piece i we have.

Suppose there are K pairs of edges. Write e_k for edge k, and $-e_k$ for its mate (its reverse). (Which particular one gets to be positive vs negative doesn't matter). E.g.
$-((a_1, b_2), (A_4, B_5)) = ((A_4, B_5), (a_1, b_2))$.
For each k, define a vector E_k, where

$$(E_k)_i = \begin{cases} 1 & \text{if edge } e_k \text{ is in piece } i \\ -1 & \text{if edge } -e_k \text{ is in piece } i \\ 0 & \text{if neither edge is in piece } i \end{cases}$$

Fatgraphs as vectors

Suppose $X \cdot E_k = 0$ (dot product). This means

$$\# \text{ of times the edge } e_k \text{ appears in the pieces in } X$$

$$=$$

$$\# \text{ of times the edge } -e_k \text{ appears in the pieces in } X$$

so if $X \cdot E_k = 0$ for all k, then every edge appears the same time as its mate, so the collection of pieces represented by X can be glued up into a fatgraph.

Fatgraphs as vectors

Define another vector H, where $H_i = 0$ if piece i is a triangles, and if piece i is a rectangle, then:

$$H_i = \begin{cases} 1 & \text{if the first letter of } x \text{ is in piece } i \\ 0 & \text{otherwise} \end{cases}$$

Notice that $X \cdot H$ gives the total number of copies of x in *any* fatgraph built from the pieces in X. I.e. $X \cdot H = n(S)$.

Computing χ

Recall $\chi = J - R$ for a fatgraph. If the fatgraph is built from triangles and rectangles, it turns out that $\chi = -T/2$. Define a vector we'll denote by χ, where

$$\chi_i = \begin{cases} \frac{-1}{2} & \text{if piece } i \text{ is a triangle} \\ 0 & \text{otherwise} \end{cases}$$

Notice that $X \cdot \chi = \chi(S)$, where S is any fatgraph assembled from the pieces in X.

Finding efficient surfaces

We claim that finding a most-efficient surface is the same thing as solving the problem:

Find X to minimize $-(X \cdot \chi)/2$, subject to the restriction that $X_i \geq 0$ and the matrix equation:

$$\begin{bmatrix} \text{---} & E_1 & \text{---} \\ \text{---} & E_2 & \text{---} \\ \text{---} & E_3 & \text{---} \\ & \vdots & \\ \text{---} & E_K & \text{---} \\ \text{---} & H & \text{---} \end{bmatrix} \begin{bmatrix} | \\ X \\ | \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Because $X \in \mathbb{R}^n$, it can be scaled so $X \cdot H = 1$, so this is just a normalization, not an actual restriction.

Finding efficient surfaces

Given a fatgraph S with x as boundary, record the pieces in a vector X, scale X so that $X \cdot H = 1$; now we have a vector which satisfies the matrix equation, and $X \cdot \chi = \chi(S)/n(S)$.

Conversely, given a vector $X \in \mathbb{Q}^N$ which satisfies the restrictions, we can scale it so it has integral entries, and build a surface S, and $\chi(S)/n(S) = X \cdot \chi$.

Thus, if we can minimize $-(X \cdot \chi)/2$ subject to the restrictions, we produce a vector which, when glued up, gives a most-efficient surface.

Finding efficient surfaces

Minimizing a linear function subject to linear inequalities is called *linear programming*, and there are good algorithms for it, hence, given a loop (or loops) x, we can:

1. Enumerate all the pieces that could appear in a fatgraph with boundary x.
2. Build the constraint matrix of vectors E_k and H
3. Solve the linear programming problem
4. Glue up an output vector to get a most-efficient surface

Key fact: the minimizing vector will always be rational.

Linear programming example

Here is a big surface that is most-efficient for its boundary: