Surface maps into orbifolds

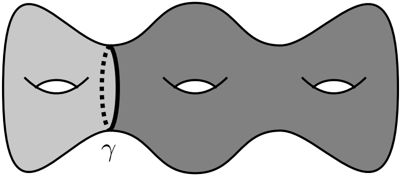
Alden Walker

October 22, 2013

Surface maps

Given a homologically trivial loop γ in a topological space X, there is a surface S with boundary and a map $f : S \to X$ such that $f(\partial S) = \gamma$.

Problem: Find such a surface with genus(S) minimal.



There are two obvious surface maps whose boundary goes to γ. One of them is better (genus 1) than the other (genus 2).

Efficient surface maps

Better problem: allow S to have multiple boundary components, where $f(\partial S)$ covers γ with arbitrary degree, and find a most *efficient* surface map.

Definition

A surface map $f : S \to X$ is *admissible* for the loop γ if

$$\begin{array}{ccc} S & \xrightarrow{f} & X \\ \uparrow & & \uparrow \gamma \\ \partial S & \xrightarrow[\partial f]{} & S^1 \end{array}$$

commutes. The total degree on the boundary is $n(S, f)$, i.e. $\partial f_*([\partial S]) = n(S, f)\left[S^1\right]$.

Then

$$\mathrm{scl}(\gamma) = \inf_{(S,f)} \frac{-\chi(S)}{2n(S, f)}$$

where the inf is taken over all admissible surface maps.

Efficient surface maps in groups

Definition 1: We conflate X and $\pi_1(X)$, so if $g \in G$, then $\mathrm{scl}(g) = \mathrm{scl}(\gamma)$ where γ represents g in $K(G,1)$.

Definition 2: For $g \in [G,G]$ we define the *commutator length* $\mathrm{cl}(g)$ to be the minimum number of commutators in a product which equals g. We define the *stable commutator length*

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

Lemma (Calegari)

These definitions are equivalent.

Things to notice about scl

Recall:

$$\mathrm{scl}(\gamma) = \inf_{(S,f)} \frac{-\chi(S)}{2n(S,f)}$$

Note:

- The inf might not be realized (there might not exist a surface map $f : S \to X$ such that $-\chi(S)/2n$ realizes the infimum). If there *is* such a surface map, it is *extremal*.
- scl seems difficult to compute.
- scl is inherently interesting, but it's also connected (dual) to bounded cohomology and quasimorphisms.
- γ need not be a single homologically trivial loop. It could be a collection of loops which are together homologically trivial.

Groups in which we can compute scl

1. Free groups (Calegari `scallop`)
2. Free products of free abelian groups (Calegari `sails`)
3. Free abelian groups amalgamated along $\mathbb{Z}^k$ subgroups (Tim Susse, algorithm descended from `sails`) (includes torus knot complements).
4. Baumslag-Solitar groups (some elements) (Matt Clay, Max Forester and Joel Louwsma)
5. Free products of cyclic groups (W. `scylla`, descended from `scallop`) (surfaces with boundaries and orbifold points).

Theoretically, (5) can be accomplished using (1), but this is infeasible and opaque. `scylla` is polynomial time and intuitive.

How to compute scl

All these scl algorithms do the following: they

1. Parameterize the space of surface maps admissible for a given loop γ
2. in such a way that $-\chi/2n$ can be computed.

How to compute scl

All these scl algorithms do the following: they

1. Parameterize the space of surface maps admissible for a given loop γ
2. in such a way that $-\chi/2n$ can be computed.

Example

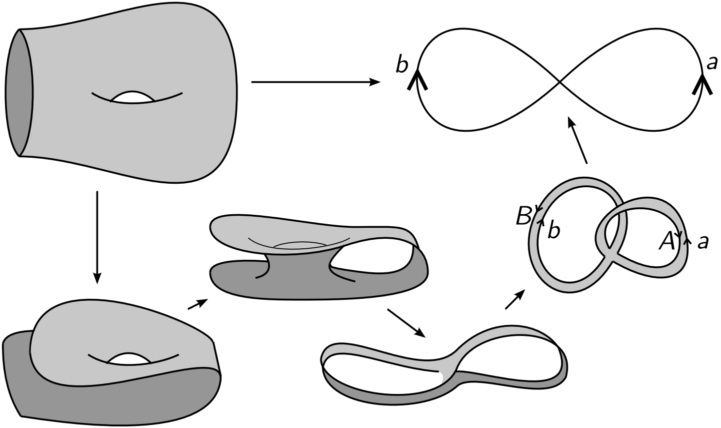
- For all the algorithms, the space of surface maps is parameterized as a polyhedron in a vector space, and the function $-\chi/2n$ is linear (`scallop`, `scylla`) or piecewise linear (`sails`).
- A consequence of these parameterizations is that scl is rational in these groups
- In `scallop` and `scylla`, the parameterizations prove that extremal surfaces exist in free groups and free products of cyclic groups.

Outline

1. Intro to scl computation (done)
2. Computing scl in free groups (`scallop`)
3. Computing scl in free products of cyclic groups (`scylla`)
4. Applications of `scylla`

Surface maps into free groups

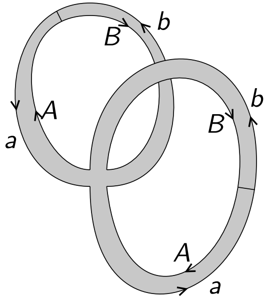
How can a surface map into a free group?



By making it skinny.

Labeled fatgraphs

The name for a skinny surface map is a *labeled fatgraph*.

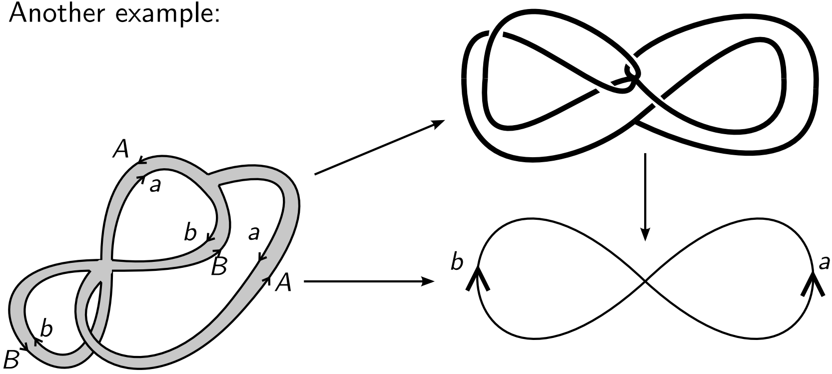


Technically, a labeled fatgraph is a graph in which every vertex has a cyclic order on the incoming edges, and every edge is labeled by a generator on one side and its inverse on the other ($A = a^{-1}$)

The combinatorial data of a labeled fatgraph induces the surface map.

Labeled fatgraphs

Another example:

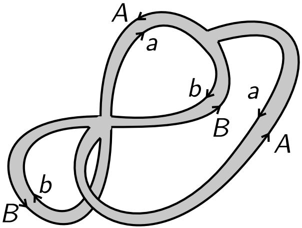


This surface map has boundary $abAABB + ab$.

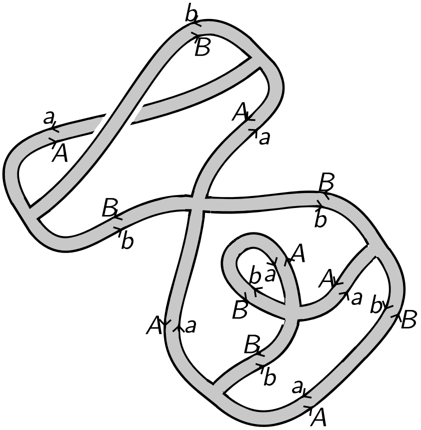
Lemma (Culler)

Every surface map factors through a labeled fatgraph (possibly after compression).

Comparing surface maps



$\chi = -2, n = 1, -\chi(S)/2n = 1$

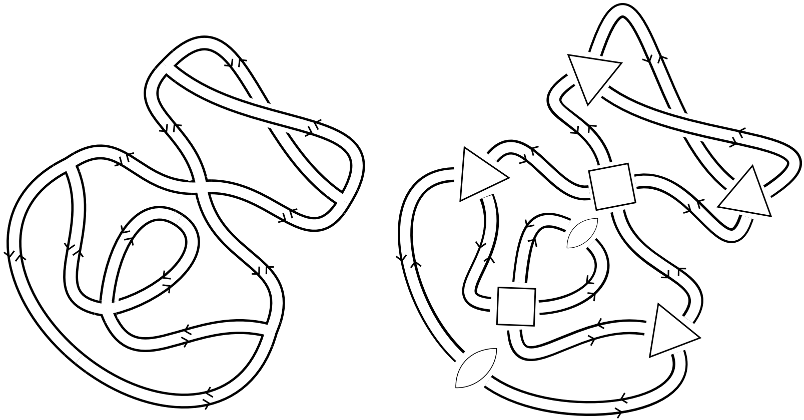


$\chi = -4, n = 3, -\chi(S)/2n = \frac{2}{3}$

Both surfaces are admissible for $abAABB + ab$. The right surface is more efficient (actually, it is extremal).

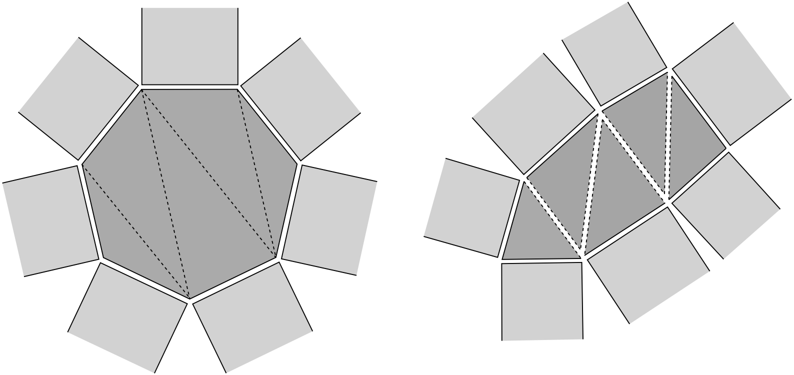
Parameterizing surface maps

Any labeled fatgraph (e.g. for $abAABB + ab$) can be broken into pieces: rectangles, and polygons:



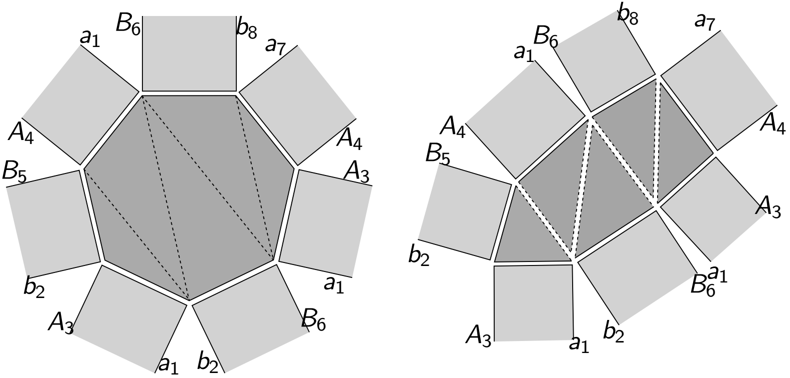
Every rectangle is labeled on each side by a particular letter from $abAABB + ab$, so there are only finitely many kinds of rectangles. But we need to understand the junctions, which we'll call *polygons*.

Polygons



Any polygon can be cut into triangles, and there are only finitely many kinds of triangles.

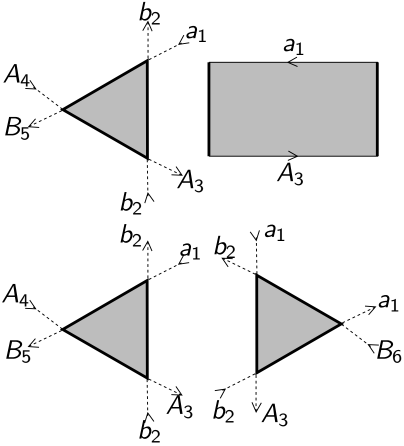
Polygons



Here is a hypothetical polygon in a fatgraph with boundary $a_1 b_2 A_3 A_4 B_5 B_6 + a_7 b_8$. 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 given 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

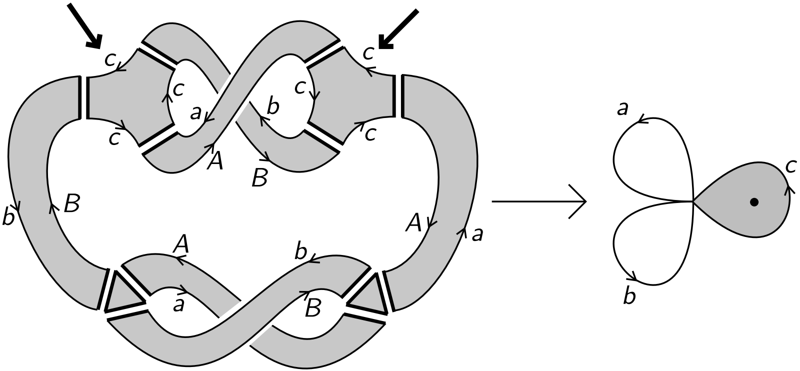
Let V be the vector space spanned by the set of all possible pieces for a boundary γ.

- Every fatgraph with boundary γ gives a vector in V.
- Given a vector in V satisfying the (linear) condition that it can be glued up, then *any* way of gluing it up has boundary γ.
- $n(S, f)$ and $-\chi(S)$ are *linear functions*

I.e. there is a polyhedron in V which parameterizes the space of admissible surface maps for γ, and computing scl reduces to a linear programming problem.

Parameterizing surface maps into orbifolds (`scylla`)

In the case of a free product of cyclic groups, we repeat the same idea, but we need a new kind of piece in our labeled fatgraphs: *group polygons*.



Here is a surface map into $\langle a \rangle * \langle b \rangle * \langle c \rangle / \langle 3c \rangle$ with boundary $acacBcAba + bBABcAcbc$. Note the boundaries of the group polygons map to c^3, so they bound disks.

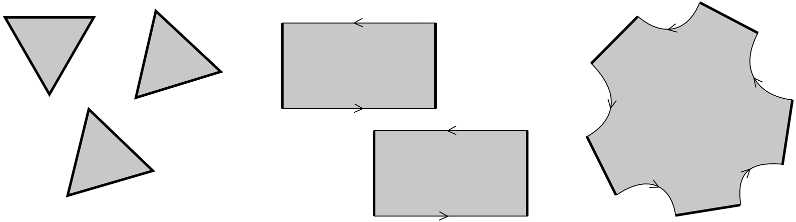
Parameterizing surface maps into orbifolds

Lemma (W.)

Every surface map into a free product of cyclic groups factors through a labeled fatgraph, possibly after compression.

Parameterizing surface maps into orbifolds

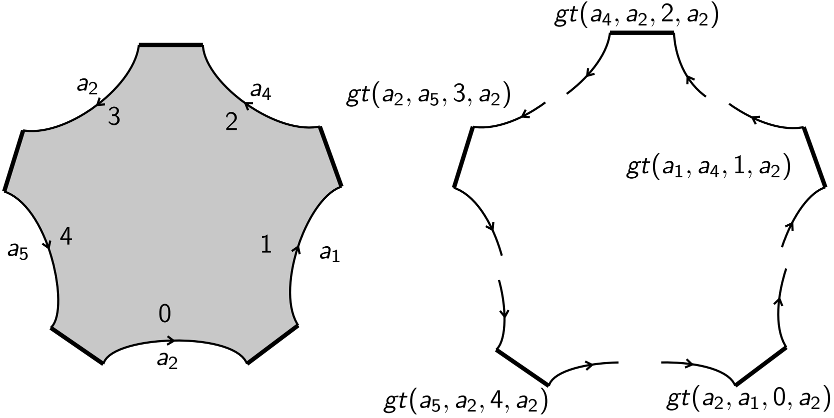
Given a loop γ, we could enumerate all triangles, rectangles, and group polygons and do linear programming in the vector space spanned by them.



But this is completely infeasible: the number of group polygons is exponential in the orders of the finite factors.

Group teeth

As a general principle, chopping a surface into smaller bits makes the linear programming problem smaller at the expense of making it trickier to reconstruct χ.



In our case, we can chop the group polygons into *group teeth*.

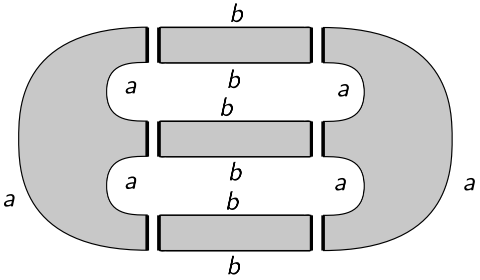
Group teeth

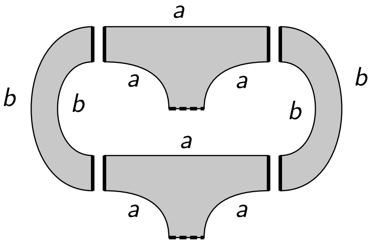
Each group tooth remembers the adjacent letters, where it is in the group polygon, and which letter is at the bottom.

The result is:

- There are at most $o|\gamma|^3$ group teeth (not exponentially many) in a finite factor with order o.
- There are linear constraints which ensure that a collection of group teeth can be glued up into group polygons.
- With K group teeth, we produce K/o group polygons (topologically, disks), so we can compute χ.
- Therefore, the space of admissible surfaces for γ is parameterized by a polyhedron in a vector space (spanned by triangles, rectangles, and group teeth), and scl can be computed by linear programming.

Some surfaces in orbifolds





These surfaces have boundary $6ab$ and $2abaab$ ($= 2abAB$) in $G = \langle a \rangle / \langle 3a \rangle * \langle b \rangle / \langle 2b \rangle$. They show that $\mathrm{scl}_G(ab) \leq 1/12$ and $\mathrm{scl}_G(abAB) \leq 0$, and they are actually both extremal.

Patterns

Let $G(n, m) = \langle a \rangle / \langle na \rangle * \langle b \rangle / \langle mb \rangle$, where $G(\infty, \infty) = \langle a \rangle * \langle b \rangle$ is a free group.

Given γ homologically trivial in $G(\infty, \infty)$, we can compute $\mathrm{scl}_{G(n,m)}(\gamma)$ for any n and m.

Lemma (W.)

$$\mathrm{scl}_{G(\infty,\infty)}(\gamma) - \frac{|\gamma|}{2(n+m)} \leq \mathrm{scl}_{G(n,m)}(\gamma) \leq \mathrm{scl}_{G(\infty,\infty)}(\gamma)$$

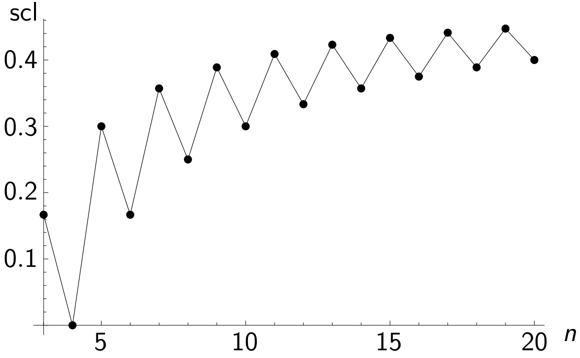
Proof.
`scylla` parameterization. □

In particular, as $n, m \to \infty$, $\mathrm{scl}_{G(n,m)}(\gamma) \to \mathrm{scl}_{G(\infty,\infty)}(\gamma)$.

Patterns

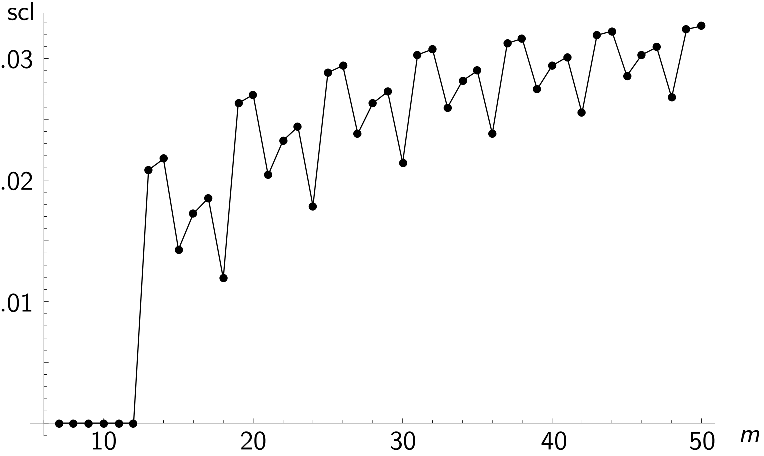
As $n, m \to \infty$, $\mathrm{scl}_{G(n,m)}(\gamma) \to \mathrm{scl}_{G(\infty,\infty)}(\gamma)$. But how does $\mathrm{scl}_{G(n,m)}(\gamma)$ vary in n and m?

Experimentally, with interesting periodic behavior!



The value of $\mathrm{scl}_{G(n,m)}(aabABAAbaB)$ as n varies (there is no dependence on m).

Patterns



The value of $\mathrm{scl}_{G(100,m)}(aba^2b^2a^3b^3A^6B^6)$ as m varies.

Immersions

The `scylla` parameterization is also useful for understanding immersions: if the target orbifold X is hyperbolic, then the `scylla` combinatorialization of a surface map $f : S \to X$ detects whether f is homotopic to an immersion with geodesic boundary.

For a loop $\gamma \in X$, we say it *virtually bounds an immersed surface* if there is an immersed surface with geodesic boundary γ^n.

Theorem (W.)

Let X be a hyperbolic orbifold with one boundary component b (and a condition on the gcd *of the orbifold point orders). If $w \in \pi_1(X)$ is any homologically trivial loop in X, then wb^N virtually bounds an immersed surface for any sufficiently large N.*

cf. Calegari and Calegari-Louwsma.