Groups Geom. Dyn. 18 (2024), 191–202
DOI 10.4171/GGD/747

On commutator length in free groups

Laurent Bartholdi, Sergei O. Ivanov, and Danil Fialkovski

Abstract. Let F be a free group. We present for arbitrary $g \in \mathbb{N}$ a LogSpace (and thus polynomial time) algorithm that determines whether a given $w \in F$ is a product of at most g commutators; and more generally, an algorithm that determines, given $w \in F$, the minimal g such that w may be written as a product of g commutators (and returns ∞ if no such g exists). This algorithm also returns words $x_1, y_1, \dots, x_g, y_g$ such that $w = [x_1, y_1] \dots [x_g, y_g]$. These algorithms are also efficient in practice. Using them, we produce the first example of a word in the free group whose commutator length *decreases* under taking a square. This disproves in a very strong sense a conjecture by Bardakov.

1. Introduction

Let F be a free group and $[F, F]$ its derived subgroup; so every element $w \in [F, F]$ is a product of commutators $[u, v] = u^{-1}v^{-1}uv$. The minimal number of terms in such a product is called the *commutator length* of w. This "norm" $\| \cdot \|$ on $[F, F]$ was the subject of much investigation, already by Burnside [2, §238, p. 319, Example 7], but is still poorly understood, in particular in relation to the usual word length $|w|$. One surprising phenomenon is that $\|w^m\|$ can be smaller than $m \cdot \|w\|$; for $F = \langle x, y \rangle$, we have

$$\|[x, y]^3\| \leq 2 \quad \text{since } [x, y]^3 = [x^{-1}yx, x^{-2}yxy^{-1}] \cdot [yxy^{-1}, y^2]$$

(and in fact equals 2; more generally, $\|[x, y]^m\| = \lfloor \frac{m}{2} \rfloor + 1$, see [5, Example 2.6]. In contrast, *stable commutator length*, the limit $\operatorname{scl}(w) = \lim_{m\to\infty} \frac{\|w^m\|}{m}$, is much better understood, see [3]).

1.1. Algorithms

The first algorithm for computing commutator length was constructed by Goldstein and Turner [7]; see also [5, 12]. The method is fundamentally topological: construct a graph with w labeled along a cycle, and determine the minimal genus of a topological surface on which the graph embeds. Given a graph, it is straightforward to compute the minimal genus by linear algebra, but a large number of graphs need to be considered.

2020 Mathematics Subject Classification. Primary 20E05; Secondary 20F12.
Keywords. Commutator length, equations in free groups.

Bardakov suggests a more algebraic algorithm in [1], see Section 2.1, which translates the problem into a calculation in the symmetric group $S_{|w|}$. He proves that $\|w^m\|$ increases at least linearly with m, giving a lower bound based on a quasi-homomorphism, and conjectures $\|w^m\| \geq \frac{m+1}{2}\|w\|$ for all w, m. Note that from [14], we have $\|w^m\| \geq 2$ for all $m \geq 2$ and $w \neq 1$.

Yet a different algorithm is proposed by Calegari [3, p. 96sqq], based on linear programming. In fact, *stable* commutator length can be computed as the solution of a linear program of polynomial size in $|w|$, and integer solutions lead to commutator length. Integer linear programming is much more computationally intensive than real linear programming, but is nevertheless feasible, and has been implemented by Walker, as the program SCALLOP [15].

Writing group conjugates as $w^g = g^{-1}wg$, our main result is the following.

Theorem A. *Consider a non-trivial word $w \in [F,F]$ in a free group $F = \langle S \rangle$. Then there exists a factorization of w without cancellations,*

$$w = w_1 a^{-1} w_2 b^{-1} w_3 a w_4 b w_5,$$

with $a, b \in S \cup S^{-1}$ and $\|w_1 w_4 w_3 w_2 w_5\| = \|w\| - 1$. Furthermore, we have

$$w = [(w_4 w_3 a)^{w_1^{-1}}, (b w_2^{-1} w_3^{-1})^{w_4^{-1} w_1^{-1}}] \cdot (w_1 w_4 w_3 w_2 w_5).$$

Recall that a decision problem is in LogSpace if it can be solved by a Turing machine with read-only input and one auxiliary read-write tape initially empty, with the guarantee that its read-write head remains within $\mathcal{O}(\log n)$ steps of the origin, for an input word of length n. Its number of total configurations is bounded by a polynomial in n, so such a machine stops after polynomial time if it ever stops. Lipton and Zalcstein prove in [11] that the word problem in free groups is in LogSpace. From Theorem A, we deduce the following corollary.

Corollary B. *Let a free group F and an integer $g \in \mathbb{N}$ be fixed. Then the problem "Given $w \in F$, is $\|w\| \leq g$?" is in* LogSpace.

Let F be a free group. We call *F-RAM machines* the extension of the computational model of RAM machines with finitely many registers holding elements of F, which can be left- and right-multiplied by generators and tested on their left-most and right-most letter in constant time.

Corollary C. *Let a free group F be fixed. Then there is an algorithm for an F-RAM machine that, given $w \in F$, determines $\|w\|$ in time $\mathcal{O}(|w|^{4\|w\|})$. Furthermore, this algorithm returns a representation of w as a product of $\|w\|$ commutators.*

An algorithm by Wicks [16] determines whether a word w is a commutator by the criterion: "some cyclic permutation of w must have the form $w_1^{-1} w_2^{-1} w_3^{-1} w_1 w_2 w_3$ as a product without cancellation". This leads to an $\mathcal{O}(|w|^3)$-time algorithm by searching for the possible starting positions of w_1, w_2, w_3. We give, in Section 2.4, an algorithm that, assuming constant-time arithmetic operations on integers, improves the average time complexity to $\mathcal{O}(|w|^2)$.

Given a solution $w = [u_1, v_1] \dots [u_g, v_g]$ to the problem of expressing w as a product of g commutators, numerous other solutions may be derived by elementary transformations, such as "replace u_i by $v_i u_i$". Viewing a solution as a homomorphism $\Sigma_{g,1} \to F$ with $\Sigma_{g,1} = \langle u_1, v_1, \dots, u_g, v_g, c \mid [u_1, v_1] \dots [u_g, v_g] = c \rangle$ the fundamental group of a surface of genus g with one boundary component, we see that the mapping class group of that surface (which coincides with the outer automorphism group of $\Sigma_{g,1}$) naturally acts by precomposition on the space of solutions. Culler proves in [5, Theorem 4.1] that there are finitely many orbits of solutions under the mapping class group action. We believe that our algorithm produces at least one solution in every orbit.

For example, running our algorithm on $w = x^{-1}y^{-1}x^2yx^{-1}$ with $g = 1$ produces, with the convention $x^{-1} = X$ and $y^{-1} = Y$, the solutions

$$w = [x, yX] = [xx, yX] = [Yx, YX] = [yx, XX] = [yx, X]$$

while running it on $w = x^{-1}x^{-1}y^{-1}x^{-1}yxy^{-1}x^2y$ with $g = 2$ produces the solutions

$$\begin{aligned}
w &= [YYXyx, YxYxyy] \cdot [y, xy] = [YYXyx, YxYxyy] \cdot [y, Yxy] \\
&= [YxYXyx, YxxYxyXy] \cdot [Xy, y] = [YxYXyx, YxxYxyXy] \cdot [YXy, y] \\
&= [Yxyx, YxxyxYXy] \cdot [y, X] = [x, Yxyx] \cdot [x, y] = [YXyxYx, xyXYxy] \cdot [y, x] \\
&= [yxYx, xyxyXY] \cdot [Y, X] = [xx, xyXYxy] \cdot [y, x] = [XyxYxx, yyXYx] \cdot [x, Y] \\
&= [xx, yXYxy] \cdot [y, x] = [XYxyxx, XYxyx] \cdot [x, y] = [x, XYxyx] \cdot [x, y] \\
&= [XXyxYxxx, XyXyXYxx] \cdot [x, Yx] = [XXyxYxxx, XyXyXYxx] \cdot [x, XYx] \\
&= [xx, XyXYxy] \cdot [y, x] = [XXyxx, x] \cdot [Yx, xy] = [XXyxx, x] \cdot [xx, xy] \\
&= [XXyxx, x] \cdot [xx, y] = [XXyxx, x] \cdot [xx, Xy] \\
&= [XXYxyxx, XXYxx] \cdot [Yx, xy] = [XXYxyxx, XXYxx] \cdot [xx, xy] \\
&= [XXYxyxx, XXYxx] \cdot [xx, y] = [XXYxyxx, XXYxx] \cdot [xx, Xy] \\
&= [XXYxxyxx, XXYxyXYXyxx] \cdot [x, yx] \\
&= [XXYxxyxx, XXYxyXYXyxx] \cdot [x, Xyx] \\
&= [XXYxxyxx, XXXYXyxx] \cdot [x, yx] = [XXYxxyxx, XXXYXyxx] \cdot [x, Xyx].
\end{aligned}$$

1.2. Non-monotonicity

We already noted that $\|w^m\| < m \cdot \|w\|$ can occur, for example with $w = [x, y]$. Bardakov's conjecture "$\|w^m\| \geq \frac{m+1}{2}\|w\|$" is refuted from general results on solutions of equations in free groups: Kharlampovich and Myasnikov deduce [9, Theorem 3] the existence of a sequence (w_n) in $[F, F]$ with $\|w_n\| \to \infty$ and $\|w_n^2\|$ bounded; equivalently, the infinite product F^∞ contains an element of order 2 in its abelianization.

These results are however fundamentally not constructive, and explicit elements with $\|w^2\| < \|w\|$ had eluded discovery; see [6, Question 2].

Theorem D. *There exists an element $w \in [F, F]$ of length* 64*, see* (3)*, with $\|w\| = 3$ and $\|w^2\| = 2$.*

We obtain this result by running our algorithm on an enumeration of the solutions to a quadratic equation, see Section 3. Some tricks and techniques were necessary to render such a computation (and discovery) feasible, which we record in Section 2.4.

2. A commutator length algorithm

We begin by recalling standard notation. Fix a free group $F = \langle x_1, \dots, x_r\rangle$. Its elements are represented as *words* over the *letters* $x_1, \dots, x_r, x_1^{-1}, \dots, x_r^{-1}$; words may be simplified by removing a pair of contiguous letters aa^{-1} as often as possible, and two words are deemed equal if they simplify to the same word, called *freely reduced*. A word is called *cyclically reduced* if furthermore its first and last letters do not cancel. We denote by $w[i]$ the ith letter of a word w, numbered from 1 to $|w|$; and for $j \geq i - 1$, we denote by $w[i : j]$ the possibly-empty subword $w[i]w[i+1]\dots w[j]$. The following is our main algorithm computing commutator length; some improvements will be given in Section 2.4.

Algorithm 1: Test recursively whether $\|w\| \leq g$

```
cyclically reduce $w$;
if $g = 0$ then
    return $w = 1$
for $1 \leq i < j < k < \ell \leq |w|$ do
    if $w[i] \neq w[k]^{-1}$ or $w[j] \neq w[\ell]^{-1}$ then
        continue
    if $\|w[1 : i-1]w[k+1 : \ell-1]w[j+1 : k-1]w[i+1 : j-1]$
                          $w[\ell+1 : |w|]\| \leq g - 1$ then
        return true
return false
```

2.1. Bardakov's theorem

Let us denote by S_n the symmetric group on n elements. We compose elements of S_n right-to-left, as functions. Every $\sigma \in S_n$ induces a partition of $\{1, \dots, n\}$ into orbits. Let $(1, \dots, n)$ denote the cyclic permutation of $\{1, \dots, n\}$, and for a permutation $\pi \in S_n$ write

$$v(\pi) := \text{number of orbits of } (1, \dots, n)\pi.$$

Let now $w = w_1 \dots w_n$ be a word (not necessarily reduced) over an alphabet $\{x_1^{\pm 1}, \dots, x_r^{\pm 1}\}$. A *pairing* on the word w is an involution $\pi \in S_n$ such that $a_{\pi(i)} = a_i^{-1}$ for all i. Note that a pairing exists if and only if w represents an element of the commutator subgroup.

Theorem 2.1 ([1, Theorem 1]). *Consider a word w representing a non-trivial element also written w of the derived subgroup. Then*

$$\|w\| = \min_{\pi \text{ pairing on } w} \Big(\frac{1 - v(\pi)}{2} + \frac{n}{4} \Big).$$

Remark 2.2. Bardakov proves his result only in the case of cyclically reduced words, but does not use this in the proof. He also overlooks the restriction that w be non-trivial.

2.2. Induced permutations

Let us consider more generally the symmetric group S_X on a set X, and an injective map $\alpha: Y \to X$. There is a *renormalization map* $S_X \to S_Y$ induced by α, defined as follows: for $\sigma \in S_X$ and $y \in Y$, let $m(y) \geq 1$ be the least positive integer such that $\sigma^{m(y)}(\alpha(y)) \in \alpha(Y)$, and set

$$\sigma_\alpha(y) := \alpha^{-1}(\sigma^{m(y)}(\alpha(y))).$$

This is the permutation of Y obtained from the disjoint cycle representation of σ by erasing all elements of $X \setminus \alpha(Y)$ and replacing every element of $\alpha(Y)$ by its α-preimage.

Note that the renormalization map is not quite a homomorphism; nevertheless, we have the following assertion.

Lemma 2.3. *Let $\alpha: Y \to X$ be an injective map, and let $\tau \in S_X$ be such that $\tau(\alpha(Y)) = \alpha(Y)$, so $\tau = \tau'\tau''$ with τ' fixing $\alpha(Y)$ and τ'' fixing $X \setminus \alpha(Y)$. Then for all $\sigma \in S_X$, we have*

$$(\sigma\tau)_\alpha = (\sigma\tau')_\alpha \tau_\alpha.$$

Proof. We first note $\tau''_\alpha = \alpha^{-1}\tau\alpha = \tau_\alpha$. Consider then $y \in Y$, write $x := \alpha(y)$, and let $m > 0$ be minimal such that $(\sigma\tau)^m(x) \in \alpha(Y)$; thus we have $(\sigma\tau)_\alpha(y) = \alpha((\sigma\tau)^m(x))$. Now $\tau(x) = \tau''(x)$ by assumption, while $z_i := (\sigma\tau)^i(x) \notin \alpha(Y)$ for $i < m$, so $\tau(z_i) = \tau'(z_i)$; thus $(\sigma\tau)^i(x) = (\sigma\tau')^i(\tau''(x))$, and m is also minimal such that

$$(\sigma\tau')^m(\tau''(x)) \in \alpha(Y). \qquad \blacksquare$$

Lemma 2.4. *Let $\alpha: Y \to X$ be an injective map and consider $\sigma \in S_X$. Then the map $O \mapsto \alpha^{-1}(O)$ is a bijection between the orbits of σ on X that intersect $\alpha(Y)$ and those of σ_α on Y. In particular, if $\alpha(Y)$ intersects every orbit of σ, then σ and σ_α have the same number of orbits.*

Proof. Let O be a σ-orbit of X, which by assumption contains $\alpha(y) =: x$ for some $y \in Y$. It suffices to prove that $\alpha^{-1}(O)$ is a σ_α-orbit. Now $O = \{\sigma^i(x) \mid i \geq 0\}$; let $0 = m_0 < m_1 < \cdots$ be all the indices m_i such that $\sigma^{m_i}(x) \in \alpha(Y)$, so $O \cap \alpha(Y) = \{\sigma^{m_i}(x) \mid i \geq 0\}$, and $\sigma^{m_i}(x) = \alpha(\sigma_\alpha^i(y))$, so $\alpha^{-1}(O) = \{\sigma_\alpha^i(y) \mid i \geq 0\}$. ■

Consider now $1 \leq i < j < k < \ell \leq n$, and denote by α the injective map

$$\{1, \dots, n-4\} \to \{1, \dots, n\}$$

defined by

$$\begin{aligned}1, \dots, n-4 \mapsto 1, \dots, i-1, k+1, \dots, \ell-1, j+1, \dots, k-1, i+1, \dots, j-1, \\ \ell+1, \dots, n.\end{aligned}$$

Proposition 2.5. *Assume that $n \geq 5$ and let $\pi \in S_n$ map i, j, k, ℓ to k, ℓ, i, j, namely its cycle decomposition contains the cycles (i,k) and (j,ℓ). Then $v(\pi) = v(\pi_\alpha)$.*

Proof. Write $\rho = (1, \dots, n)(i,k)(j,\ell)$ and $\pi'' = (i,k)(j,\ell)\pi$, the restriction of π to the range of α, so $\rho\pi'' = (1, \dots, n)\pi$. An easy calculation checks $\rho_\alpha = (1, \dots, n-4)$. We note that the image of α intersects all orbits of $(1, \dots, n)\pi$: indeed, it suffices to show that no orbit is contained in $\{i, j, k, \ell\}$. Now the images of i, j, k, ℓ are $k+1, \ell+1, i+1, j+1 \pmod n$, respectively, so this may happen only if $i+1 = j$, $j+1 = k$, $k+1 = \ell$, $\ell+1 = i-n$ and hence $n = 4$. We thus have

$$\begin{aligned} v(\pi) &= \text{number of orbits of } (1, \dots, n)\pi \overset{\text{Lemma 2.4}}{=} \text{number of orbits of } ((1, \dots, n)\pi)_\alpha \\ &= \text{number of orbits of } (\rho\pi'')_\alpha \overset{\text{Lemma 2.3}}{=} \text{number of orbits of } \rho_\alpha\pi_\alpha \\ &= \text{number of orbits of } (1, \dots, n-4)\pi_\alpha = v(\pi_\alpha). \end{aligned}$$ ∎

2.3. Proof of Theorem A

We recall our main result.

Theorem 2.6. *Consider a non-trivial word $w \in [F, F]$. Then there exists a factorization of w without cancellations,*

$$w = w_1 a^{-1} w_2 b^{-1} w_3 a w_4 b w_5, \quad \textit{with } a, b \textit{ generators or their inverses}$$

and $\|w_1 w_4 w_3 w_2 w_5\| = \|w\| - 1$. Furthermore, we have

$$w = [(w_4 w_3 a)^{w_1^{-1}}, (b w_2^{-1} w_3^{-1})^{w_4^{-1} w_1^{-1}}] \cdot (w_1 w_4 w_3 w_2 w_5).$$

Proof. The last equality is directly checked by expanding the commutator; therefore,

$$\|w_1 w_4 w_3 w_2 w_5\| \geq \|w\| - 1,$$

and it remains to prove the reverse inequality. Write $n = |w|$, and let $\pi \in S_n$ be a pairing on w maximizing $v(\pi)$. If $n = 4$, then $w_1 = w_2 = w_3 = w_4 = w_5 = 1$ and the result follows, so we may assume $n \geq 5$. Say two transpositions (i,k), (j,ℓ) with $i < k$, $j < \ell$ are *linked* if the intervals $[i,k]$ and $[j,\ell]$ are neither nested nor disjoint. If all transpositions in the cycle decomposition of π are unlinked, then w freely reduces to the trivial word;

we may thus assume that there are linked transpositions (i,k), (j,ℓ) in π, ordered so that $i<j<k<\ell$. By Proposition 2.5 and twice by Theorem 2.1, we have

$$\begin{aligned}\|w_1w_4w_3w_2w_5\| &\le \frac{2-2v(\pi_\alpha)+n-4}{4} = \frac{2-2v(\pi)+n-4}{4}\\ &= \frac{2-2v(\pi)+n}{4}-1=\|w\|-1.\end{aligned}$$ ■

Proof of Corollary B. We unroll g times the recursion in Algorithm 1, arriving at an iterative algorithm with $4g$ nested loops. We implement it using a Turing machine whose read-write storage contains $4g$ pointers in the range $\{1,\dots,n\}$. Using these $4g$ pointers $i_1,j_1,k_1,\ell_1,\dots,i_g,j_g,k_g,\ell_g$, it is possible with $\mathcal{O}(\log n)$ memory to compute the function mapping $i\in\{1,\dots,n\}$ to the ith letter of the word obtained at the last stage of the recursion. We then apply the Lipton–Zalcstein algorithm to check in LogSpace whether this word is trivial. ■

Algorithm 2: factorization(w,g)

```
Data: A word $w$ and an integer $g \geq 0$
Result: A list of pairs of words expressing $w$ as a product of $g$ commutators, or
        fail if no such factorization exists
cyclically reduce $w$;
if $g = 0$ then
    return fail if $w \neq 1$, else ()
for $1 \leq i < j < k < \ell \leq |w|$ do
    if $w[i] \neq w[k]^{-1}$ or $w[j] \neq w[\ell]^{-1}$ then
        continue
    $(w_1, w_2, w_3, w_4, w_5) \leftarrow (w[1 : i-1], w[i+1 : j-1], w[j+1 : k-1],$
                                  $w[k+1 : \ell-1], w[\ell+1 : |w|])$;
    $F \leftarrow$ factorization$(w_1 w_4 w_3 w_2 w_5, g-1)$;
    if $F \neq$ fail then
        return $((w_1 w_4 w_3 w[k] w_1^{-1}, w_1 w_4 w[\ell] w_2^{-1} w_3^{-1} w_4^{-1} w_1^{-1}), F)$
return fail
```

Proof of Corollary C. To compute the commutator length of w, we apply Algorithm 1 first with $g=0$, then $g=1$, then $g=2$ etc. till it succeeds. In the RAM model, it is possible to keep track of $w_1w_4w_3w_2w_5$ as a linked list of letters, and at each elementary step of the algorithm (increase i,j,k or ℓ) a bounded number of operations need be executed to adjust the product $w_1w_4w_3w_2w_5$, including its free cancellations. Therefore, the complexity of the algorithm, at each step of the recursion, is controlled by the loop over $i<j<k<\ell$, with time complexity $\mathcal{O}(|w|^4)$.

Furthermore, Algorithm 1 may be modified in a straightforward manner to return an expression of w as a product of commutators, see Algorithm 2. ■

2.4. Implementation

It is possible to speed up the algorithm a little bit, and this was crucial for the computations in the next section. Let us concentrate on the case $\|w\| = 1$, namely determine whether w is itself a commutator. A naive implementation, looping over all positions of a^{-1}, b^{-1}, a, b and testing whether $w_1w_4w_3w_2w_5$ freely reduces to the trivial word, requires $\mathcal{O}(|w|^5)$ steps: four factors $|w|$ for the loop and one for the reduction of $w_1w_4w_3w_2w_5$.

This time can be reduced to $\mathcal{O}(|w|^3)$ assuming large machine integers, and even to $\mathcal{O}(|w|^2)$ on average, as follows: firstly, rather than keeping track of words w_1, w_2, w_3, w_4, w_5, we store 2×2 integer matrices faithfully representing them. If F has rank 2, we can even choose as images of generators the transvections $\left(\begin{smallmatrix}1&2\\0&1\end{smallmatrix}\right)$ and $\left(\begin{smallmatrix}1&0\\2&1\end{smallmatrix}\right)$. We update the words w_i at each step of the algorithm by elementary row and column operations. In this manner, words of length up to 64 may faithfully be stored into four 64-bit integers. Next, we note that we may loop first on the positions of a^{-1} and a; and that, if w is a commutator, then the words $w_3^{-1}bw_2^{-1}$ and $w_4bw_5w_1$ are conjugates of each other. Knowing their matrix representations, we can immediately rule out a pair of a-positions if these matrices have different traces, a constant-time test that eliminates almost all candidates (note that integer overflow is not an issue here, since we are looking for cheap ways of ruling out some candidates).

For those that survive the test, rather than considering the $\mathcal{O}(|w|^2)$ pairs of b-positions, we could cyclically reduce these words and check whether they are cyclic permutations of each other, using for example the Knuth–Morris–Pratt algorithm [10]; but this case occurs so seldom in practice that it was not worth implementing.

In even more detail: our implementation loops over i, k, j, ℓ in that order, and maintains matrices M_{23451}, M_{23}, M_{451}, M_{32}, M_{514} representing the respective products $w_2b^{-1}w_3aw_4bw_5w_1$, $w_2b^{-1}w_3$, $w_4bw_5w_1$, w_3w_2, $w_5w_1w_4$. At each elementary step (increase one of i, j, k, ℓ), one of these matrices is to be multiplied on the left or on the right by a generator. For each choice of i, k, the implementation checks whether M_{451} and M_{23} have same trace (recall that in $\mathrm{SL}_2(\mathbb{Z})$ the trace of a matrix equals the trace of its inverse), before looping on j and ℓ.

3. Quadratic equations and Theorem D

Consider the equation $x_1^2x_2^2x_3^2x_4^2 = 1$ in a free group F. A solution (x_1, x_2, x_3, x_4) is the same thing as a homomorphism $\phi\colon S \to F$ with $S = \langle x_1, x_2, x_3, x_4 \mid x_1^2x_2^2x_3^2x_4^2 = 1\rangle$. According to [13], the set Φ of solutions ϕ to the equation is characterized as follows: let F_2 denote the free group of rank 2. A *literal* solution $\lambda\colon S \to F_2$ is a homomorphism sending each generator to a word of length ≤ 1. The *mapping class group* of S is the group of automorphisms of S that preserve $\{x_1\}^S \cup \dots \cup \{x_4\}^S$; equivalently, automorphisms of the free group $\langle x_1, \dots, x_4\rangle$ that preserve the conjugacy class of $x_1^2x_2^2x_3^2x_4^2$. Then

$$\Phi = \{\eta \circ \lambda \circ \alpha \mid \alpha \in \mathrm{MCG}(S),\ \lambda \text{ literal},\ \eta\colon F_2 \to F \text{ any homomorphism}\}. \tag{1}$$

3.1. The mapping class group of the non-orientable surface of genus 2

In order to explore the solutions of the equation

$$x_1^2 x_2^2 x_3^2 x_4^2 = 1,$$

it is helpful to also consider a different presentation of S, in which generators of $\mathrm{MCG}(S)$ are easier to write. We use

$$S = \langle s_1, t_1, s_2, t_2 \mid s_1 t_1 s_1^{-1} t_1^{-1} s_2 t_2 s_2^{-1} t_2 \rangle.$$

The isomorphism is given for instance by

$$(s_1, t_1, s_2, t_2) = (x_2 x_3^2 x_4, x_3^{-1} x_2^{-1}, x_3 x_4 x_2, x_3 x_4 x_1 x_2).$$

Using these generators, we have the following mapping classes, in which unlisted generators are fixed:

$$\alpha_1\colon t_1 \mapsto t_1 s_1^{-1}, \quad \beta_1\colon s_1 \mapsto s_1 t_1, \quad \beta_2\colon s_2 \mapsto s_2 t_2,$$

$$\gamma_1\colon \begin{cases} s_1 \mapsto s_1 u, \\ t_1 \mapsto u^{-1} t_1 u, \\ s_2 \mapsto u^{-1} s_2 \end{cases} \quad \text{for } u = s_2 t_2^{-1} s_2^{-1} t_1,$$

$$\eta_3\colon \begin{cases} s_1 \mapsto s_1 v, \\ t_1 \mapsto v^{-1} t_1 v, \\ s_2 \mapsto v^{-1} t_1 s_2, \\ t_2 \mapsto t_1^{-1} s_2 t_2 s_2^{-2} t_1 s_2 \end{cases} \quad \text{for } v = s_2^2 t_2^{-1} s_2 t_2^{-1} s_2^{-1} t_1,$$

and by [4, Appendix], these elements generate $\mathrm{MCG}(S)$.

We add for good measure the automorphism

$$\delta\colon x_1 \mapsto x_2 \mapsto x_3 \mapsto x_4 \mapsto x_1$$

in the original generators, and thus consider as generating set for $\mathrm{MCG}(S)$ the collection

$$\Sigma = \{\alpha_1^{\pm 1}, \beta_1^{\pm 1}, \beta_2^{\pm 1}, \gamma_1^{\pm 1}, \eta_3^{\pm 1}, \delta, \delta^2, \delta^3\}. \tag{2}$$

In the generating set $\{s_1, t_1, s_2, t_2\}$, there are three maximal literal solutions τ_{-1}, τ_0, τ_1 given for $i = -1, 0, 1$ by

$$\tau_i\colon \begin{cases} s_1 \mapsto x, \\ t_1 \mapsto x^i, \\ s_2 \mapsto y, \\ t_2 \mapsto 1. \end{cases}$$

3.2. Proof of Theorem D

The connection to Theorem D is the following: given a solution $(x_1, \dots, x_4) \in F^4$ to $x_1^2 x_2^2 x_3^2 x_4^2 = 1$, consider the element $w = x_1 x_2 x_3 x_4$. Then

$$\begin{aligned} w^2 &= (x_1 x_2 x_3 x_4)^2 (x_1^2 x_2^2 x_3^2 x_4^2)^{-1} \\ &= ([x_4^{-1} x_3^{-1}, x_3^{-1} x_2^{-1} x_1^{-1}] \cdot [x_2^{-1} x_1^{-1}, x_2])^{x_2^{-1} x_1^{-1}}, \end{aligned}$$

so for every solution (x_1, x_2, x_3, x_4), the element $w^2 = (x_1 x_2 x_3 x_4)^2$ has commutator length at most 2. On the other hand, the first claim of Theorem D will follow by constructing a sufficiently complicated solution.

We computed a finite set of solutions from the set Φ defined in (1) by restricting to α that are products of at most 5 of the generators from Σ given in (2), to $\lambda \in \{\tau_{-1}, \tau_0, \tau_1\}$, to $F = F_2$ and to $\eta = 1$. We computed, for each solution, the image w of $x_1 x_2 x_3 x_4$ (which is conjugate to t_2), and sorted them in increasing order of length. We finally computed the commutator length of each of these solutions (see the next section), and after examining about 2500 candidates arrived at a solution

$$\begin{aligned} w =& x^{-2} y x y^2 x^{-2} y x y x^{-1} y^{-1} x y^2 x^{-1} y^2 x y^{-1} x^{-1} y^{-1} x^2 y^{-2} x^{-1} y^{-1} \\ & \cdot x y^{-2} x y^{-2} x^{-1} y^{-1} x y^{-2} x^{-1} y x y^{-1} x^{-1} y^{-1} x^2 y^{-1} x^{-1} y^{-1} x y^2 \\ & \cdot x^{-1} y x y^2 x^{-2} y x y \end{aligned} \tag{3}$$

with commutator length 3. It was produced by $\lambda = \tau_1$ and $\alpha = \eta_3^{-1} \circ \alpha_1 \circ \eta_3^{-1} \circ \delta^2 \circ \eta_3^{-1}$. Dozens of other solutions appeared, but no shorter one.

Algorithm 1, with the improvements described in Section 2.4, could test words of length comparable to w in about a second. Words of length above 100 could be routinely tested. For extra safety and to protect against hidden bugs, the commutator length of w was also computed using Walker's program SCALLOP [15]. With the options "-CYCLIC-C" and the GUROBI solver [8] as backend, it certified $\|w\| = 3$ for the word w from (3) in a bit more than one hour.

3.3. Open problems

There does not seem to be much hope to obtain, by brute force, words w with $\|w\| \geq 4$ and $\|w^2\| = 2$. However, the construction of w in (3) in the form $\tau_1(\alpha(x_1 x_2 x_3 x_4))$ for a mapping class α that is a product of 5 generators raises the following problem.

Question 3.1. Does there exist a mapping class $\alpha \in \mathrm{MCG}(S)$ with

$$\|\tau_1(\alpha^n(x_1 x_2 x_3 x_4))\| \geq n$$

for all $n \in \mathbb{N}$?

This would provide a systematic collection of solutions, and remove much of the non-constructivity of the implicit function theorem in free groups.

It is as of yet unknown whether commutator length can decrease upon taking *cubes*; more importantly, whether there exists a sequence (w_n) in F with $\|w_n\| \to \infty$ and $\|w_n^3\|$ bounded, or equivalently whether there is 3-torsion in the abelianization of F^∞. Following the same idea, one would want arbitrarily complicated solutions to the equation $x_1^3 x_2^3 x_3^3 x_4^3 = 1$, a goal that seems out of reach now.

Danny Calegari suggested a variant of the question that may be more tractable: "can one find w with $\|3w\| < \|w\|$?". Here $\|m \cdot w\|$ is the minimal number of commutators required to express a product of m conjugates of w, and is trivially at most $\|w^m\|$. A sequence (w_n) with $\|w_n\| \to \infty$ and $\|3w_n\|$ bounded would likewise imply the existence of 3-torsion in $H_1(F^\infty)$.

Funding. This work is supported by the Ministry of Science and Higher Education of the Russian Federation, agreement 075-15-2019-1619.

References

[1] V. G. Bardakov, Computation of commutator length in free groups. *Algebra Logic* **39** (2000), no. 4, 224–251 Zbl 0960.20019 MR 1803583

[2] W. Burnside, *Theory of groups of finite order*. 2nd edn., Dover Publications, New York, 2017 Zbl 1375.20001 MR 69818

[3] D. Calegari, *scl*. MSJ Mem. 20, Mathematical Society of Japan, Tokyo, 2009 Zbl 1187.20035 MR 2527432

[4] J. A. Comerford and Y. Lee, Product of two commutators as a square in a free group. *Canad. Math. Bull.* **33** (1990), no. 2, 190–196 Zbl 0654.20030 MR 1060374

[5] M. Culler, Using surfaces to solve equations in free groups. *Topology* **20** (1981), no. 2, 133–145 Zbl 0452.20038 MR 605653

[6] C. C. Edmunds and G. Rosenberger, Powers of genus two in free groups. *Canad. Math. Bull.* **33** (1990), no. 3, 342–344 Zbl 0681.20022 MR 1077108

[7] R. Z. Goldstein and E. C. Turner, Applications of topological graph theory to group theory. *Math. Z.* **165** (1979), no. 1, 1–10 Zbl 0377.20027 MR 521516

[8] Gurobi Optimization, LLC, Gurobi optimizer reference manual. 2021, https://www.gurobi.com, visited on 3 October 2023

[9] O. Kharlampovich and A. Myasnikov, Implicit function theorem over free groups and genus problem. In *Knots, braids, and mapping class groups – papers dedicated to Joan S. Birman (New York, 1998)*, pp. 77–83, AMS/IP Stud. Adv. Math. 24, American Mathematical Society, Providence, RI, 2001 Zbl 1007.20020 MR 1873109

[10] D. E. Knuth, J. H. Morris, Jr., and V. R. Pratt, Fast pattern matching in strings. *SIAM J. Comput.* **6** (1977), no. 2, 323–350 Zbl 0372.68005 MR 451916

[11] R. J. Lipton and Y. Zalcstein, Word problems solvable in logspace. *J. Assoc. Comput. Mach.* **24** (1977), no. 3, 522–526 Zbl 0359.68049 MR 445901

[12] A. Yu. Ol'shanskii, Homomorphism diagrams of surface groups. *Sib. Math. J.* **30** (1989), no. 6, 961–979 Zbl 0792.20040 MR 1043443

[13] D. Piollet, Solutions d'une équation quadratique dans le groupe libre. *Discrete Math.* **59** (1986), no. 1–2, 115–123 Zbl 0599.20035 MR 837961

[14] M.-P. Schützenberger, Sur l'équation $a^{2+n} = b^{2+m}c^{2+p}$ dans un groupe libre. *C. R. Acad. Sci. Paris* **248** (1959), 2435–2436 Zbl 0090.24403 MR 103219

[15] A. Walker, Scallop. 2021, https://github.com/aldenwalker/scallop, visited on 3 October 2023

[16] N. J. Wicks, Commutators in free products. *J. Lond. Math. Soc.* **37** (1962), 433–444 Zbl 0107.01802 MR 142610

Received 20 October 2021.

Laurent Bartholdi
Mathematisches Institut, Universität des Saarlandes, Campus E2.4, 66123 Saarbrücken, Germany;
laurent.bartholdi@gmail.com

Sergei O. Ivanov
Laboratory of Modern Algebra and Applications, St. Petersburg State University, 29b 14th Line,
199178 Saint Petersburg, Russia; Yanqi Lake Beijing Institute of Mathematical Sciences and
Applications (BIMSA), No. 544, Hefangkou Village, Huaibei Town, Huairou District,
101408 Beijing, P. R. China; ivanov.s.o.1986@gmail.com

Danil Fialkovski
Laboratory of Modern Algebra and Applications, St. Petersburg State University, 29b 14th Line,
199178 Saint Petersburg, Russia; 19fdr97@gmail.com