Bachelor Thesis

The forested graph complex

Jean-Luc Portner

supervised by
Prof. Dr. Thomas Willwacher, advisor
Dr. Michael Borinsky, co-advisor

Department of Mathematics
ETH Zürich
June 2022

1 Introduction

Since its beginning in the late 18th century, group theory has been one of the fundamental concepts of mathematics and groups have been studied intensely ever since. Free groups are the class of groups where no relations between the generators exist: The elements are words of generators and their inverses. The group relation is given by concatenating words and reducing them i.e. if two adjacent elements are inverse to each other they will be omitted. The free group on n generators will be denoted by F_n. The importance of free groups arises from the fact that every group is isomorphic to a quotient group of a free group. In particular, any finitely generated group is isomorphic to a quotient group of F_n for some n.

Another way to approach a group G is to characterize its automorphism group $\mathrm{Aut}(G)$, as it describes G's symmetries. $\mathrm{Aut}(G)$ can be separated into the inner automorphism group $\mathrm{Inn}(G)$, the group of automorphisms that arise from conjugation, and the outer automorphism group $\mathrm{Out}(G)$, the quotient of $\mathrm{Aut}(G)$ by $\mathrm{Inn}(G)$.

Combining these two aspects, it is only natural that mathematicians intensely study the automorphism group of F_n. The inner automorphism group is well understood and isomorphic to F_n itself. However the structure of the outer automorphism group $\mathrm{Out}(F_n)$ remains for the most part unknown. For the first free group we have that $F_1 \cong \mathbb{Z}$ and thus

$$\mathrm{Out}(F_1) = \mathrm{Out}(\mathbb{Z}) = \mathrm{GL}_1(\mathbb{Z}).$$

For F_2 Nielsen proved in [15] that $\mathrm{Out}(F_2) \cong \mathrm{GL}_2(\mathbb{Z})$. For higher groups, only the existence of a surjection $\mathrm{Out}(F_n) \to \mathrm{GL}_n(\mathbb{Z})$ can be guaranteed.

Historically, $\mathrm{GL}_n(\mathbb{Z})$ has been studied by its action on the symmetric space $\mathrm{SL}_n(\mathbb{R})/\,\mathrm{SO}_n(\mathbb{R})$. Due to the relation between $\mathrm{Out}(F_n)$ and $\mathrm{GL}_n(\mathbb{Z})$, early attempts at studying the outer automorphism group of F_n examined its action on $\mathrm{SL}_n(\mathbb{R})/\,\mathrm{SO}_n(\mathbb{R})$ induced by the above surjection. However, this action is not proper, that is the preimage of a compact set under the group action is not necessarily compact. Hence it behaves quite badly and a different approach was needed.

Therefore, Culler and Vogtmann introduced a new space $\mathcal{X}_n$ known as "Outer space" in [19]. To describe this space we first have to introduce some basic concepts of graph theory.

1.1 Graphs

Definition 1.1. A *graph* G is a finite 1-dimensional CW complex. The set of edges is denoted by $E(G)$, the set of vertices by $V(G)$. We call an edge having the same start and end vertex a *loop*.

We call a graph *connected* if the CW complex is connected in the topological sense. A graph is *n-edge-connected* if it remains connected after removing $n-1$ arbitrary edges.

For a vertex v of a graph G we call the number of incident edges *valency* or *degree* and denote it by $\deg(v)$. A graph is said to be *n-regular* if every vertex has valency n.

For a subset Φ of the edges of G we denote by G/Φ the *graph quotient*, which is the quotient space of the CW complex G over its topological subspace Φ.

Remark 1.2. Note that sometimes these types of graphs are called multigraphs, as they are allowed to have loops as well as multiple edges between vertices. The word graph then normally refers to simple graphs which do not allow multi-edges or loops.

In the context of algebraic topology however, multigraphs are needed and thus the word graph here denotes multigraphs.

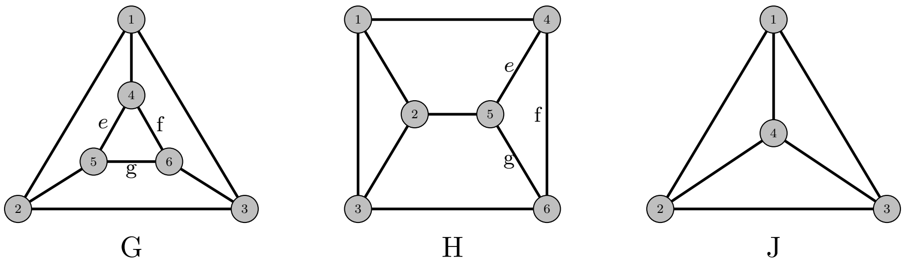
Definition 1.3. A *subgraph* G' of a graph G is a subcomplex of the CW complex G. Since subcomplexes are themselves CW complexes of dimension smaller or equal to the original complexes dimension, G' itself is a graph.

A *cycle* in a graph G is a subgraph that is homeomorphic to S^1. A *tree* is a connected graph containing no cycles. A *forest* is a collection of disjoint trees.

Definition 1.4. Let G, H be two graphs. A map $f : V(G) \to V(H)$ is said to be a *graph isomorphism* if f is a bijection such that

$$(u, v) \in E(G) \Leftrightarrow (f(u), f(v)) \in E(H).$$

Example 1.5. Consider the following graphs.



Then G is 3-connected and 3-regular. Moreover, G is isomorphic to H. An isomorphism between them is given by mapping the equally labelled vertices to each other. The graph quotient $G/\{e, f, g\}$ is given by J.

The following result gives the fundamental connection between graphs and free groups:

Theorem 1.6. *Let G be a graph. Then its fundamental group $\pi_1(G)$ is isomorphic to a free group.*

A proof of this theorem will be given later at the beginning of section 2.1.

Definition 1.7. By the theorem, it makes sense to define the *rank of a graph* $\mathrm{rank}(G)$ as the rank of its fundamental group.

Finally, a *metric graph* is a finite connected graph where each edge is assigned a positive real value, its length. Moreover, we endow the edges with the *path metric* i.e. the distance between two points is the length of the shortest path between them. Here the length of the path is the sum of the lengths of the edges that are (partially) traversed.

1.2 Outer space

We continue with the study of $\mathrm{Out}(F_n)$ and the aforementioned Outer space. For this we follow Vogtmann's survey article [17].

To define Outer space, we first consider the graph R_n given by one vertex and n edges each forming a loop. By Theorem 1.6 we can identify $\pi_1(R_n)$ with the free group F_n. We do this by orienting the edges of R_n and identifying them with the generators of F_n $x_1, \dots, x_n$. Then every reduced word $w \in F_n$ corresponds to a closed reduced walk in R_n, given by traversing the edges in the same order as their assigned generators appear in w. An automorphism ϕ on F_n corresponds to a homotopy equivalence sending the loop corresponding to x_i to the loop identified with $\phi(x_i)$.

Definition 1.8. Let M be the set of pairs (g, G) where G is a finite metric graph with vertex valency ≥ 3 and a total edge length of one. Moreover g is a homotopy equivalency $g : R_n \to G$ called the marking of G. Now we define *Outer space* $\mathcal{X}_n$ as the quotient space of M over the equivalence relation given by $(g, G) \sim (g', G')$ if and only if there exists an isometry $h : G \to G'$ such that $g \circ h$ is homotopic to g'.

A point (g, G) of $\mathcal{X}_n$ can be represented as follows: We choose a maximal forest T on G and orient all edges in $G \setminus T$. Moreover we label these edges with an element of F_n such that the labelling satisfies the following: If $f : G \to R_n$ is the map determined by the labels by sending T to the vertex of R_n and sending each edge in $G \setminus T$ to the corresponding loop in R_n, then f is a homotopy inverse to g. This is the case exactly when the words labelling the edges form a basis of F_n. An example is given in Figure 1, where the orange edges represent the forest T.

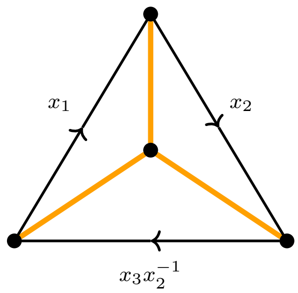
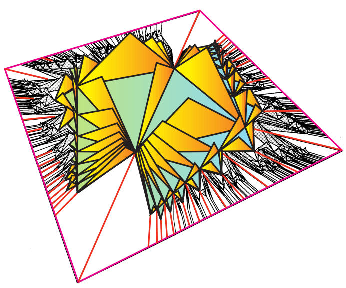


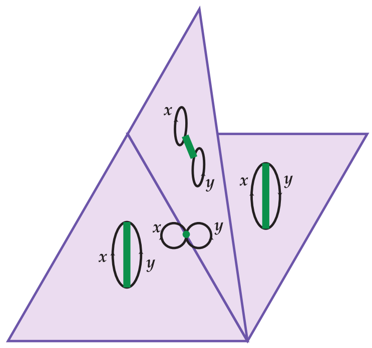
Figure 1: A point of Outer Space

To define a topology on $\mathcal{X}_n$ we consider the set $\mathcal{C}$ of conjugacy classes of F_n. These classes are the cyclically reduced words, that is words for which every cyclic permutation is reduced. Now we can define a map from Outer space $\mathcal{X}_n$ to the infinite projective space $\mathbb{RP}^{\mathcal{C}}$. For a fixed marked graph (g, G) the map assigns to each cyclically reduced word w the length of the unique cyclically reduced closed walk in G homotopic to $g(w)$. As this map is injective, we can view $\mathcal{X}_n$ as s subspace of $\mathbb{RP}^{\mathcal{C}}$ and thus give $\mathcal{X}_n$ the subspace topology.

This definition might seem quite ad hoc, however with it $\mathcal{X}_n$ decomposes nicely into a disjoint union of open simplices: Every marked graph (g, G) belongs to the open simplex containing all graphs that can be reached by varying the non-zero edge lengths such that the total edge length of G remains 1. The faces of the simplex are then the marked graphs where one edge of (g, G) has been fully contracted. For a marked graph with $k + 1$ edges, the corresponding simplex is k-dimensional. Of course, a contracted edge can again be extended in multiple ways thus connecting the different simplices. An example is given in the figure below[1] on the right. On the left, we see a depiction of $\mathcal{X}_2$.



(a) Outer space $\mathcal{X}_2$



(b) Simplices in $\mathcal{X}_2$

[1]Both figures are taken from [17]

Moreover, the identification works the other way round i.e. every open simplex in X_n is a face of a maximal simplex that corresponds to a trivalent marked graph. Taking the argument from the proof of Theorem 2.12, which will be presented in section 2.2, we see that the dimension of $\mathcal{X}_n$ is equal to $3n - 4$.

Now with the Outer space defined, we can consider the right group action of $\mathrm{Out}(F_n)$ on $\mathcal{X}_n$: Every $\phi \in \mathrm{Out}(F_n)$ induces a map $f : R_n \to R_n$ by mapping the edge labelled x to the edge labelled $\phi(x)$. Then the right group action is defined by $(g, G)\phi = (g \circ f, G)$.

An inconvenience that arises is that the quotient of $\mathcal{X}_n$ by $\mathrm{Out}(F_n)$ is not compact. Resolving this leads us to the next construction which comes in the next section.

1.3 The spine of Outer space

The beginning of this section follows Culler and Vogtmanns original definition from [19]. In the latter half, we adhere to [18]. We first need the following definition:

Definition 1.9. A subcomplex B of a simplicial complex A is called *full* if for every simplex $\Delta \in B$ whose vertices are contained in A Δ is also a simplex of A.

Now we can define a more convenient and simpler version of $\mathcal{X}_n$, *reduced Outer space*, and denote it by Y_n. The points in the subspace Y_n are the marked graphs (g, G) that do not contain any separating edges, i.e. no edges e such that $G \setminus e$ is disconnected. An equivariant deformation retraction from $\mathcal{X}_n$ to Y_n is given by shrinking the lengths of the separating edges to zero while uniformly extending the lengths of the other edges to preserve the total edge length of 1. Finally, we can define the spine K_n as follows:

Definition 1.10. Let $\widetilde{Y}_n$ be the barycentric subdivision of Y_n. Then *the spine of outer space* K_n is defined as the maximal full subcomplex of $\widetilde{Y}_n$ which is disjoint from the boundaries of the open simplices in Y_n. The vertices of K_n are the barycenters of simplices in Y_n i.e. the marked graphs whose edges are of equal length. The deformation retraction from Y_n to K_n is given by collapsing every simplex τ in $\widetilde{Y}_n$ to the face of τ contained in K_n. This can be done equivariantly. Therefore K_n can be thought of as ignoring the metric structure on Y_n and only focusing on its combinatorial structure.

Going the other way, consider two vertices (g, G) and (g', G') in K_n. Then the open simplex in $\mathcal{X}_n$ determined by (g, G) is a face of the one determined by (g', G') exactly when G is obtained from G' by collapsing a forest of edges in G' and g is homotopic to the composition of g' with the collapsing map. This collapsing is also called a *forest collapse*. It follows that K_n has the structure of a simplicial complex where a k-simplex is a chain of k forest collapses.

With this in mind we can determine the dimension of K_n: As collapsing an edge decreases the number of vertices by 1 a similar argument as in the proof of Theorem 2.12 shows that $\dim(K) = 2n - 3$. An example of a part of the spine of $\mathcal{X}_2$ is given in Figure 3[2].

Let us come back to the right action of $\mathrm{Out}(F_n)$ on Outer space: The action on $\mathcal{X}_n$ extends to a simplicial action on K_n. Culler and Vogtmann proved in [19] that this action has finite stabilizers. Thus the rational homology of $\mathrm{Out}(F_n)$ can be computed as the quotient of K_n by $\mathrm{Out}(F_n)$:

$$H_\bullet(\mathrm{Out}(F_n), \mathbb{Q}) \cong H_\bullet(K_n/\,\mathrm{Out}(F_n), \mathbb{Q}).$$

To calculate this homology, we turn K_n into a cube complex i.e. a CW complex where the cells are homeomorphic to Euclidean cubes and the attaching maps identify faces with lower-dimensional cubes via homeomorphisms. For a forest Φ in G with k edges we can now define

[2] This figure is taken from [17]

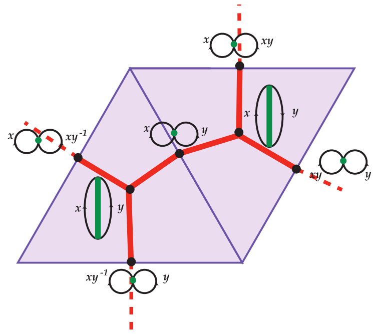


Figure 3: A section of the spine of Outer space $\mathcal{X}_2$.

its k-cube: From Φ we get a chain of k-forest collapses by collapsing each edge in Φ at a time. This yields a k-simplex. Collapsing the edges in another order yields another k-simplex. All these different k-simplices can now be fit together to triangulate a k-dimensional cube. Thus every k-cube is given by a graph G and a forest Φ of size k. The faces of dimension $k-1$ are the graphs obtained from G where one edge in Φ has been collapsed. An example is shown in Figure 4. Here the orange edges represent the edges that are being contracted along that face.

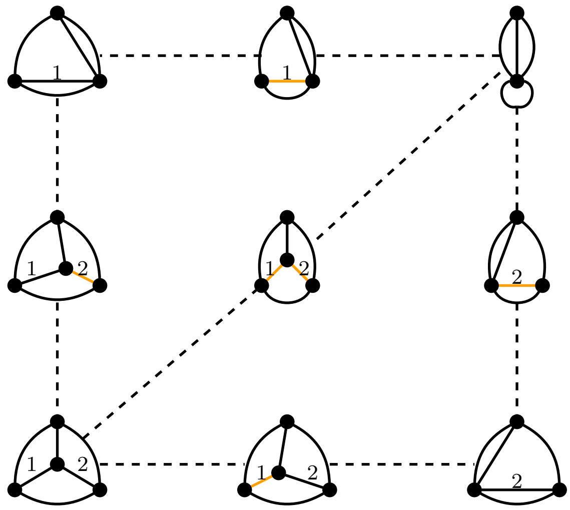


Figure 4: A cube in the spine K_3

Let us now view K_n as this cube complex with one cube for every tuple (g, G, Φ). We can define an orientation on the cube by ordering the edges of Φ such that odd permutations reverse the orientation. Then the rational homology of $K_n/\operatorname{Out}(F_n)$ can be computed from a chain complex with one generator for each pair (G, Φ) that has no orientation-reversing automorphism.

1.4 The Forested graph complex

The chain complex of pairs (G, Φ) computing $H_k(K_n/\operatorname{Out}(F_n); \mathbb{Q})$ is generally known as the forested graph complex and has been introduced by Conant and Vogtmann in [6]. In this introductory section, we will describe the original construction. In the later section 2.2, we are going to introduce a more practical hands-on but equivalent definition from [4].

Definition 1.11. A *forested graph* is a pair (G, Φ) of a finite connected trivalent graph G and an oriented forest Φ containing all vertices of G. The orientation on the forest is given by an ordering of its edges where interchanging any two edges reverses the orientation.

We now denote by $\widehat{f\mathcal{G}}_k$ the vector space spanned by forested graphs with forest size k modulo the relations $(G, \Phi) = -(G, -\Phi)$.

If we consider a forested graph (G, Φ) and we collapse an edge e in Φ then the obtained graph (G_e, Φ_e) has exactly one 4-valent vertex. As the image below shows, there are exactly two other graphs whose edge collapse leads to the graph (G_e, Φ_e).

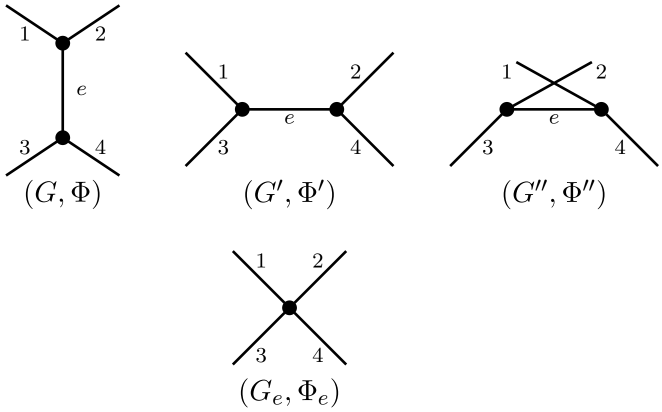


Figure 5: The numbers represent the four parts of the graph that get connected at e. The graphs G, G', G'' then represent the three different ways how this can happen so that collapsing e leads to G_e.

If we denote those two by (G', Φ') and (G'', Φ'') then we call the vector

$$(G, \Phi) + (G', \Phi') + (G'', \Phi'').$$

the *basic IHX relator associated to* (G, Φ, e). We denote by IHX_k the subspace of $\widehat{f\mathcal{G}}_k$ spanned by all basic IHX relators and define $f\mathcal{G}_k$ as the quotient space $\widehat{f\mathcal{G}}_k / IHX_k$.

Finally, we can define a boundary map $\partial : f\mathcal{G}_k \to f\mathcal{G}_{k+1}$ induced by the map on $\widehat{f\mathcal{G}}_k$ given by

$$\partial_E(G, \Phi) = \sum (G, \Phi \cup e)$$

where we sum over all edges in $G \setminus \Phi$ such that $\Phi \cup e$ is still a forest and e gets the label $k+1$ in the orientation.

One can check that ∂_E is a boundary map that is $\partial_E^2 = 0$. With this shown, we get a chain complex $f\mathcal{G}_\bullet$ with boundary map ∂_E. The rational homology of this complex computes the rational homology of $\mathrm{Out}(F_n)$ as explained at the end of the previous section.

1.5 The known homology of $\mathrm{Out}(F_n)$

To conclude this introduction we try to summarize the known rational homology of $\mathrm{Out}(F_n)$. As we only consider the rational homology, we will omit the field $\mathbb{Q}$. We first need to introduce two concepts:

Definition 1.12. If a group G contains a torsion-free subgroup K of finite index, then the *virtual cohomological dimension* $\mathrm{vcd}(G)$ is the cohomological dimension of K, i.e. the smallest number $n \in \mathbb{N}$ such that $H^k(K) = 0$ for all $k > n$. It is independent of the choice of the subgroup.

Moreover, a series of groups $G_1 \subseteq G_2 \subseteq \ldots$ is called *homologically stable* if for every k there exists a N such that

$$H_k(G_n) \cong H_k(G_{n+1}) \quad \text{for all} \quad n \geq N$$

that is for n large enough the homology is independent of n.

Culler and Vogtmann showed in [19] that $H_k(\mathrm{Out}(F_n))$ is finitely generated and vanishes for k greater than $\mathrm{vcd}(\mathrm{Out}(F_n))$, which is $2n-3$. This gives us an upper bound on homology.

Furthermore, they showed that the spine K_n is path-connected from which we conclude that $H_0(\mathrm{Out}(F_n)) = \mathbb{Q}$.

On the other hand, homological stability was proven for $\mathrm{Out}(F_n)$ for $n \geq 5(k+1)/4$ by Hatcher and Vogtmann in [11, 12]. Galatius proved in [7] that these groups are in fact zero, giving us a lower bound.

Only five of the non-trivial homology groups are explicitly known. For the groups $H_4(\mathrm{Out}(F_4))$, $H_8(\mathrm{Out}(F_6))$ and $H_{12}(\mathrm{Out}(F_8))$ it is known that the Morita classes do not vanish and in fact, as these homology groups have dimension 1, they are generated by the Morita classes. We will expand on this in section 3. Ohashi showed in [16] that all other groups for $n \leq 6$ are trivial. The only other known groups were determined by Bartholdi in [2] and are $H_8(\mathrm{Out}(F_7))$ and $H_{11}(\mathrm{Out}(F_7))$. He also showed that all other homologies for $n = 7$ are trivial. Figure 6 has been adapted from Figure 1 in [3] and summarizes these results.

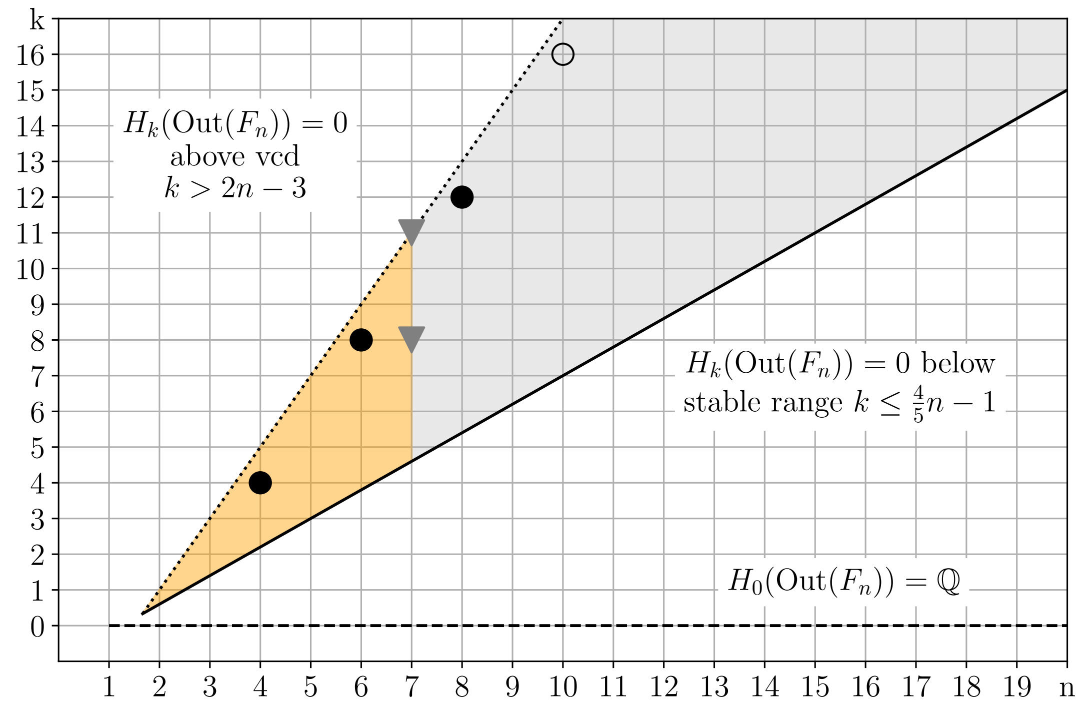


Figure 6: Homology classes of $\mathrm{Out}(F_n)$. The circles represent the Morita classes and are filled in if they are known to be non-trivial. The triangles are Bartholdi's non-trivial classes. The orange shaded region represents the area where the homology is known to be trivial. The grey shaded area is the unknown homology.

The Euler characteristic of a topological space is the alternating sum over the rank of the homology groups. Computations of it for $\mathrm{Out}(F_n)$ were done by Morita, Sakasai and Suzuki in [13, 14] for $n \leq 11$ and are shown in the table below.

n	3	4	5	6	7	8	9	10	11	12
$\chi(\mathrm{Out}(F_n))$	1	2	1	2	1	1	-21	-124	-1202	?

If the trend of rapidly increasing negativity starting at $n = 9$ continues, it would imply that there are a lot of odd-dimensional homology classes for $n \geq 9$. The only known odd-dimensional homology class however is Bartholdi's $H_{11}(\mathrm{Out}(F_7))$. Thus we see that the homology of $\mathrm{Out}(F_n)$ still remains mostly unknown.

2 The forested graph complex

Before we can introduce the forested graph complex we are going to proof a few more results about graphs.

2.1 Further results on graphs

We begin this section by proving Theorem 1.6 which is stated again below. Afterwards, we expand on the concept of the rank and the degree of a graph and prove several identities.

Theorem 1.6. *Let G be a graph. Then its fundamental group $\pi_1(G)$ is isomorphic to a free group.*

Proof. The following proof is from [10, p. 43f.]. Let G be a graph. W.l.o.g. G is connected as else we consider each connected component separately. Let T be a spanning tree on G i.e. T is a tree containing every vertex of G. Now choose for every $e_\alpha \in E(G) \setminus E(T)$ an open neighbourhood A_α of $E(T) \cup e_\alpha$, that deformation retracts onto $T \cup e_\alpha$. The intersection of such A_α deformation retracts onto T and is thus contractible. Moreover, as G is connected as a graph, A_α and T are path-connected. Now the A_α form an open cover of G and as T is simply connected by Van Kampen's theorem, we get that $\pi_1(G) = *_\alpha \pi_1(A_\alpha)$. Finally, every A_α deformation retracts onto S^1 and thus $\pi_1(A_\alpha) = \mathbb{Z}$. Now there are exactly $|E(G)| - |E(T)|$ many A_α, which as T is a spanning tree and hence $|E(T)| = |V(G)| - 1$ results in $\pi_1(G)$ being free on $|E(G)| - |V(G)| + 1$ generators. □

To understand the rank better we will need the following definitions:

Definition 2.1. For a finite CW complex X the *Euler characteristic* is defined as the alternating sum

$$\chi(X) = k_0 - k_1 + k_2 - \dots$$

where k_i denotes the number of cells of dimension i in the CW complex X.

As graphs are 1-dimensional we get $\chi(G) = k_0 - k_1$ which is equal to $\chi(G) = |V(G)| - |E(G)|$.

Definition 2.2. Let G be a graph. Then its *cycle space* is the set of even-degree subgraphs of G i.e. the subgraphs of G whose vertices have even degree. This space forms a vector space over $\mathbb{F}_2$ where the vector addition is given by the symmetric difference of two or more subgraphs. A basis of this space is called *cycle basis* and two cycles are *independent* if they are linearly independent in the vector space.

Remark 2.3. The cycle space is equal to the first homology group of G with coefficients in $\mathbb{F}_2$ i.e. $H_1(G, \mathbb{F}_2)$.

The following proposition relates the rank to different invariants and gives an easy way to compute it:

Proposition 2.4. *Let G be a connected graph. Then the following are equal:*

1. *The rank of G.*
2. *The number of independent cycles in G i.e. the size of the cycle basis of G.*
3. *The first Betti number i.e. the rank of $H_1(G)$.*
4. $1 - \chi(G) = |E(G)| - |V(G)| + 1$.

For the proof of this Proposition we will need the following lemma:

Lemma 2.5. *Let A be a set. Then the abelianization of the free group on A is isomorphic to the free abelian group on A.*

Proof. Consider the space $X = \bigvee_{a\in A} S^1$. By Van Kampen's theorem $\pi_1(X) \cong *_{a\in A}\mathbb{Z}$, that is $\pi_1(X)$ is isomorphic to the free group on A. By the relative homeomorphism theorem we have $H_1(X) \cong \bigoplus_{a\in A} H_1(S_1) \cong \bigoplus_{a\in A}\mathbb{Z}$, that is $H_1(X)$ is isomorphic to the free abelian group on A. Using Hurewicz theorem we get that the abelianization of $\pi_1(X)$ is isomorphic to $H_1(X)$ and thus the desired statement. □

Proof of Proposition 2.4. (1) = (4): This was shown in the proof of Theorem 1.6.
(1) = (3): From Hurewicz Theorem we get that the abelianization of $\pi_1(G)$ is equal to $H_1(G)$ and thus by the previous Lemma that the rank of $\pi_1(G)$ is equal to the rank of $H_1(G)$ which is the first Betti number.
(4) = (2)[3]: Let T be a spanning tree on G. Consider the sets A_α given by $e_\alpha \cup T$ for $e_\alpha \in E(G) \setminus E(T)$. Each of them is a deformation retraction onto a cycle in G. Let $Z(T)$ be the set of cycles obtained in this way. Then $Z(T)$ is independent as each cycle in $Z(T)$ contains an edge not contained in any other cycle of $Z(T)$. Moreover, every cycle Z in G can be written as the symmetric difference over the cycles corresponding to the edges in $(E(G) \setminus T) \cap Z$. Thus $Z(T)$ spans the cycle space and consequently is a cycle basis. Now the size of $Z(T)$ is given by $|E(G)| - |V(G)| + 1$ and thus we conclude the proof. □

Finally, we introduce the notion of the degree of a graph, also sometimes called excess.

Definition 2.6. Let G be a connected graph of rank n with vertex-valency ≥ 3. Its *degree* is defined by

$$\deg(G) := \sum_{v\in V(G)} (\deg(v) - 3).$$

Proposition 2.7. *Let $G = (V, E)$ be a graph of rank n. Then we have the following identities:*

1. $\deg(G) = 2|E| - 3|V|$
2. $|V| = 2n - 2 - \deg(G)$
3. $|E| = 3n - 3 - \deg(G)$
4. *G is 3-regular* $\Leftrightarrow \deg(G) = 0$.

Proof. By counting half edges we get $2|E| = \sum_{v\in V} \deg(V)$. Combining this with the definition of the degree we get the first identity:

$$\deg(G) = \sum_{v\in V(G)} (\deg(v) - 3) = \sum_{v\in V} \deg(v) - 3|V| = 2|E| - 3|V|.$$

Using Proposition 2.4 and the first identity we have

$$2n - 2 - \deg(G) = 2|E| - 2|V| + 2 - 2 - 2|E| + 3|V| = |V|$$
$$3n - 3 - \deg(G) = 3|E| - 3|V| + 3 - 3 - 2|E| + 3|V| = |E|$$

which proves the second and the third identity. The last statement follows, as every element in the sum of the degree is non-negative, as every vertex has valency ≥ 3. Thus $\deg(G) = 0$ if and only if every term is 0 and therefore if and only if every vertex has valency 3. □

[3]This proof is based on Harary's proof in [9, p. 37-40].

2.2 Forested graphs and the boundary map

As stated in the introduction, the forested graph complex has first been introduced by Conant and Vogtmann in [6]. Here, however, we will introduce the simplified construction and definition of the forested graph complex given by Conant and Vogtmann in [4]. Very useful in the general understanding of what a graph complex is and how the boundary map acts was Bar-Natan's and McKay's draft [1]. Inspired by this, similar examples for the forested graph complex are presented.

Let us denote by $\mathbb{S}_n$ the symmetric group of degree n.

Definition 2.8. An *admissible graph of rank* n is a 2-edge-connected graph G with vertex-valency ≥ 3 whose fundamental group is isomorphic to F_n.

We often abbreviate an admissible graph of rank n by a graph.

Definition 2.9. Let $G = (V, E)$ be a graph. An *ordering* on its edges is a bijective function σ from E to $\{1, \dots, |E|\}$. Notice that $\mathbb{S}_{|E|}$ acts on σ by $\pi \circ \sigma$ for $\pi \in \mathbb{S}_n$. We call the tuple (G, σ) an ordered graph and note that $\mathbb{S}_{|E|}$ acts on (G, σ) by $\pi(G, \sigma) = (G, \pi\sigma)$ for $\pi \in \mathbb{S}_{|E|}$.

A *forested graph* is a triple (G, Φ, σ) where G is an admissible graph, Φ is a subset of edges that spans a forest on G and σ is an ordering on Φ i.e. $\sigma : \Phi \to \{1, \dots, |\Phi|\}$.

A map f between two forested graphs $(G, \Phi, \sigma) \to (H, \Psi, \tau)$ is said to be a *forested graph isomorphism* if f is a graph isomorphism on G, $f(\Phi) = \Psi$ and $\sigma = \tau \circ f$

We now want to construct the forested graph complex. For this we remember the notion of a graded vector space:

Definition 2.10. A graded vector space is a vector space V with a decomposition $(V_k)_{k=0}^{\infty}$ such that

$$V = \bigoplus_{k=0}^{\infty} V_k.$$

We now consider the $\mathbb{Q}$-vector space C spanned by isomorphism classes of forested graphs, subject to the relation

$$(G, \Phi, \pi \circ \sigma) = \operatorname{sgn} \pi \cdot (G, \Phi, \sigma) \quad \text{for all} \quad \pi \in \mathbb{S}_{|\Phi|}.$$

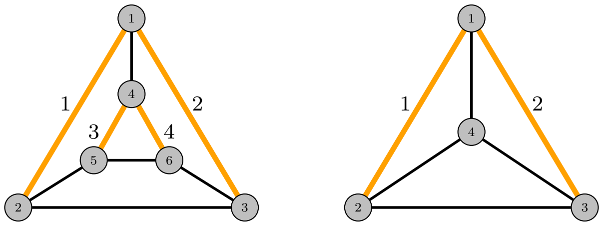
Under this relation we call σ an *orientation.* Observe that if $(G, \Phi, \sigma) \simeq (G, \Phi, \pi \circ \sigma)$ for an odd permutation π then $(G, \Phi, \sigma) \simeq (G, \Phi, \pi \circ \sigma) = -(G, \Phi, \sigma)$ and thus $(G, \Phi, \sigma) = 0$ in C.

We can define the following three gradings on C:

- Let $C^n \subseteq C$ be the subspace spanned by forested graphs of rank n. Then clearly $C^n \cap C^m = \varnothing$ for $n \neq m$ and as every graph has a rank, we get that the C^n form a grading on C.
- Let $C_k \subseteq C$ be the subspace spanned by forested graphs (G, Φ, σ) with $|\Phi| = k$. Clearly, this also yields a decomposition of C into a direct sum and thus gives another grading on C.
- Let $C_d \subseteq C$ be the subspace spanned by forested graphs of degree d. Once again this yields a grading on C.

In the following, we will mostly be concerned with the first two gradings. In particular, we will consider the double-grading C_k^n, where k denotes the forest size and n the rank.

Example 2.11. Consider the graphs G, J from Example 1.5. Then G and J are admissible graphs of rank 4 and 3. Thus if we equip them with ordered forests $(\Phi, \sigma), (\Psi, \tau)$ as below (where the orange edges represent the forest and the numbers the orientation), we get forested graphs in C_4^4 and C_2^3 respectively.



Observe, that $(J, \Psi, \tau) = 0$ in C_2^3, since (12) is an odd permutation and $(12)(J, \Psi, \tau)$ is isomorphic to (J, Ψ, τ) via the isomorphism mirroring vertices along the vertical line passing through the vertices labelled 1 and 4.

(G, Φ, σ) however is not trivial as the automorphism group is given by the identity, mirroring along the vertical, exchanging inner and outer vertices and their composition. None of these automorphisms induce an odd permutation and hence (G, Φ, σ) does not vanish.

Before we construct the chain complex we show that the C^n are finitely generated and thus so are the C_k^n.

Theorem 2.12. *For all n, C^n is finitely generated and $C_k^n = 0$ for all $k > 2n - 3$.*

To prove this theorem we first need the following lemma:

Lemma 2.13. *For $n, m \in \mathbb{N}$ There are only finitely many admissible graphs $G = (V, E)$ with $|V| \leq n$ vertices and $|E| \leq m$ edges.*

Proof. Every graph on n vertices can be written as a $n \times n$-incidence matrix and each entry is $\leq m$ if the graph has maximally m edges. Thus there are maximally m^{n^2} many different incidence matrices for graphs with n vertices and maximally m edges. As every graph corresponds to an incidence matrix, this also gives an upper bound on the number of different graphs with n vertices and maximally m edges.

Thus the maximal possible number of admissible graphs with $\leq n$ vertices and $\leq m$ edges is bounded by

$$\sum_{k=1}^{n} m^{k^2}$$

which is finite. □

Proof of Theorem 2.12. By counting half edges we have

$$2|E| = \sum_{v \in V} \deg v.$$

Using that admissible graphs have vertex-valency ≥ 3 and rearranging gives $|E| \geq \frac{3}{2}|V|$. From Proposition 2.4 we get that $|E| = |V| + n - 1$. Combining the two yields

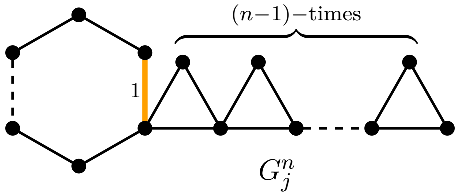
$$|V| + n - 1 \geq \frac{3}{2}|V| \Leftrightarrow 2(n-1) \geq |V|$$

and plugging in the result in the identity from Proposition 2.4 results in $|E| \leq 3(n-1)$.

Thus by the above lemma, we have that there are only finitely many graphs $G = (V, E)$ with $|V| \leq 2(n-1)$ and $|E| \leq 3(n-1)$. As each graph only has a finite number of different forests we get that C^n is finitely generated.

That $C_k^n = 0 \; \forall k > 2n - 3$ follows from the bound on the number of vertices and the fact that a forest in a graph has maximally $|V| - 1$ edges. □

Remark 2.14. Notice that for C^n to be finitely generated the constraint of vertex-valency ≥ 3 in the definition of admissible graphs is necessary. As else we can consider the following family of graphs G_j^n where the leftmost polygon contains $j \geq 2$ vertices:



By Proposition 2.4 we see that each G_j^n has rank n. Moreover the G_j^n are 2-edge connected and do not vanish, as no automorphism except the identity exists for the given forest. Finally, for $i \neq j$ G_i^n and G_j^n are not isomorphic as they have different numbers of vertices/edges. Thus for every n we find infinitely many different non-zero graphs in C^n.

Remark 2.15. The bound on the C_k^n can also not be improved as the graph G from Example 2.11 with the tree extended by the edge between the vertex labelled 1 and 4 has rank 4 and tree size 5 which equals $2 \cdot 4 - 3$. Moreover G has again no odd automorphisms and thus does not vanish.

To construct the forested graph complex we fix the rank n and define the differential as follows:

Definition 2.16. Let $(G, \Phi, \sigma) = (G, \{e_1, \dots, e_k\}, \sigma)$ be a forested graph. Then let $\partial_C, \partial_R : C_k^n \to C_{k-1}^n$ be given by

$$\partial_C(G, \Phi, \sigma) := \sum_{i=1}^{k} (-1)^i (G/e_i, \Phi \setminus \{e_i\}, \sigma_{e_i}),$$
$$\partial_R(G, \Phi, \sigma) := \sum_{i=1}^{k} (-1)^i (G, \Phi \setminus \{e_i\}, \sigma_{e_i})$$

where $\sigma_{e_i} : \Phi \setminus \{e_i\} \to \{1, \dots, k-1\}$ is given by

$$\sigma_{e_i}(e) := \begin{cases} \sigma(e) & \text{if } \sigma(e) < i \\ \sigma(e) - 1 & \text{if } \sigma(e) > i \end{cases}.$$

Notice that the case $\sigma(e) = i$ cannot happen as e_i is not contained in $\Phi \setminus \{e_i\}$. Finally define the boundary map $\partial := \partial_C - \partial_R$.

Proposition 2.17. *∂ is well-defined and $\partial^2 = 0$.*

Proof. For better readability, we will omit the orientation σ in the proof. We prove the result in three steps:

Step 1: Contracting an edge of a graph does not change the Euler characteristic as both the vertex number and the edge number decrease by one. Thus ∂_C preserves the rank of the graph. Moreover, the vertex-valency stays ≥ 3 and the graph continues to be 2-edge-connected. Hence it is admissible. Furthermore, ∂_C as well as ∂_R remove one edge from each forest, thus decreasing k by 1. Moreover, both maps are compatible with the quotient in C^n and are thus well-defined from C_k^n to C_{k-1}^n and so is ∂.

Let $(G, \Phi) \in C_k^n$ be a forested graph with $\Phi = \{e_1, \dots, e_k\}$ and denote the edges in $\Phi \setminus \{e_i\}$ by $\{e'_1, \dots, e'_{k-1}\}$, where $e'_j = e_j$ for $j < i$ and $e'_j = e_{j+1}$ for $j > i$. For the consecutive steps we need the following observations:

$$(G/e_i)/e'_j = \begin{cases} (G/e_j)/e'_{i-1} & \text{if } i > j \\ (G/e_{j+1})/e'_i & \text{if } i \leq j \end{cases} \quad \text{and} \quad (\Phi \setminus \{e_i\}) \setminus \{e'_j\} = \begin{cases} (\Phi \setminus \{e_j\}) \setminus \{e'_{i-1}\} & \text{if } i > j \\ (\Phi \setminus \{e_{j+i}\}) \setminus \{e'_i\} & \text{if } i \leq j \end{cases}$$

Step 2: *Claim:* $\partial_C^2 = 0$ and $\partial_R^2 = 0$

We compute:

$$\begin{aligned}\partial_C^2 &= \partial_C \sum_{i=1}^{k} (-1)^i (G/e_i, \Phi \setminus \{e_i\}) = \sum_{i=1}^{k} \sum_{j=1}^{k-1} (-1)^{i+j} ((G/e_i)/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \\ &= \sum_{j<i} (-1)^{i+j} ((G/e_i)/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) + \sum_{i \le j} (-1)^{i+j} ((G/e_i)/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \quad (\star)\end{aligned}$$

We claim that the two sums cancel. For this, first apply the observations above to the first sum and then change variables by setting $l = j$ and $m = i - 1$ to obtain:

$$\begin{aligned}\sum_{j<i} (-1)^{i+j} ((G/e_i)/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) &= \sum_{j<i} (-1)^{i+j} ((G/e_j)/e'_{i-1}, (\Phi \setminus \{e_j\}) \setminus \{e'_{i-1}\}) \\ &= \sum_{l \le m} (-1)^{l+m+1} ((G/e_l)/e'_m, (\Phi \setminus \{e_l\}) \setminus \{e'_m\})\end{aligned}$$

This last expression is the same as the second sum in $(\star)$ but with opposite sign. Thus they cancel and we have shown $\partial_C^2 = 0$. The same argument shows that $\partial_R^2 = 0$.

Step 3: *Claim:* $\partial_C \partial_R - \partial_R \partial_C = 0$

For the mixed terms, we compute

$$\begin{aligned}\partial_R \partial_C &= \sum_{i=1}^{k} \sum_{j=1}^{k-1} (-1)^{i+j} (G/e_i, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \\ &= \sum_{j<i} (-1)^{i+j} (G/e_i, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) + \sum_{i \le j} (-1)^{i+j} (G/e_i, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \qquad (*)\end{aligned}$$

and

$$\begin{aligned}\partial_C \partial_R &= \sum_{i=1}^{k} \sum_{j=1}^{k-1} (-1)^{i+j} (G/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \\ &= \sum_{j<i} (-1)^{i+j} (G/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) + \sum_{i \le j} (-1)^{i+j} (G/e'_j, (\Phi \setminus \{e_i\}) \setminus \{e'_j\}) \\ &\overset{(\heartsuit)}{=} \sum_{j<i} (-1)^{i+j} (G/e_j, (\Phi \setminus \{e_j\}) \setminus \{e'_{i-1}\}) + \sum_{i \le j} (-1)^{i+j} (G/e_{j+1}, (\Phi \setminus \{e_{j+1}\}) \setminus \{e'_i\}) \\ &\overset{(\dagger)}{=} \sum_{l \le m} (-1)^{m+l+1} (G/e_l, (\Phi \setminus \{e_l\}) \setminus \{e'_m\}) + \sum_{m<l} (-1)^{m+l-1} (G/e_l, (\Phi \setminus \{e_l\}) \setminus \{e'_m\}) \quad (**)\end{aligned}$$

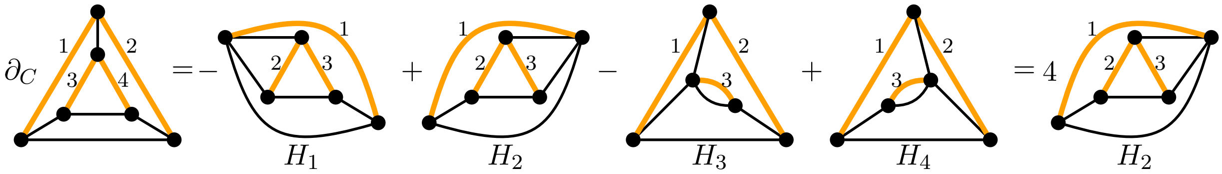
Where in $(\heartsuit)$ we used that if $j < i$ then $e'_j = e_j$ and if $i \le j$ then $e'_j = e_{j+1}$, as well as the observation above. In $(\dagger)$ we used the substitution $m = i - 1$, $l = j$ on the left and $m = i$, $l = j + 1$ on the right sum. Comparing the sums in $(*)$ and $(**)$ we see that they differ by a sign and thus cancel. Hence $\partial_C \partial_R - \partial_R \partial_C = 0$.

Combining step 2 and 3 we get:

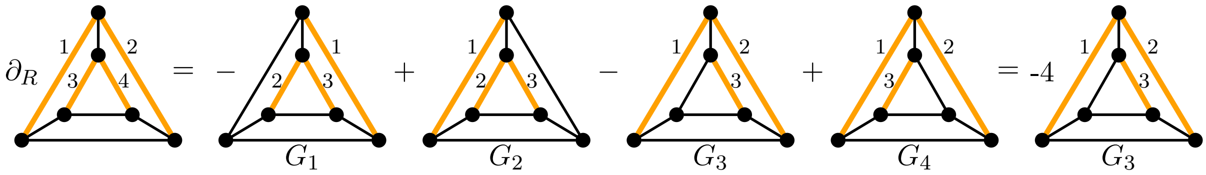
$$\partial^2 = (\partial_C - \partial_R)^2 = \partial_C^2 - (\partial_C \partial_R + \partial_R \partial_C) + \partial_R^2 = 0 \qquad \square$$

Thus the spaces $(C^n_\bullet)$ with the differential $\partial_\bullet$ form a chain complex.

Example 2.18. Once again we consider the graph (G, Φ, σ) from Example 2.11 and calculate its boundary:



Where we used that $-H_1$ is equal to H_2 by mirroring along the vertical and applying (23), $-H_3$ is equal to H_2 by exchanging inner and outer vertices, mirroring along the vertical and applying (13) and H_4 is equal to H_2 by exchanging inner and outer vertices and applying (13)(23). And for ∂_R we get:



Where we have used that $-G_1$ is equal to $-G_3$ by exchanging inner and outer vertices and applying (13)(12), G_2 is equal to $-G_3$ by exchanging, mirroring along the vertical and applying (13) and G_4 is equal to $-G_3$ by mirroring along the vertical and applying (12).

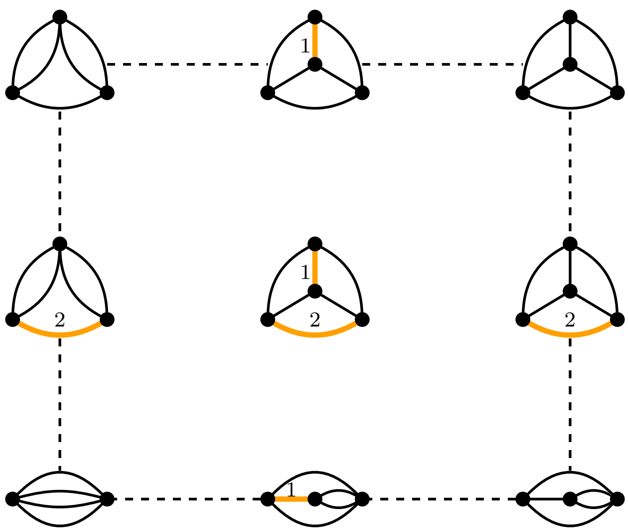
Thus we can conclude that $\partial G = 4H_2 - 4G_3$. Moreover, we have that $H_2 - G_3 \in \operatorname{Im} \partial$ and as $\partial^2 = 0$ we also have $H_2 - G_3 \in \operatorname{Ker} \partial$

Remark 2.19. As the example above shows, the calculation of the boundary, especially identifying isomorphic graphs and finding odd automorphisms, quickly becomes quite tedious. Therefore it is best to leave this to the computer and a python implementation can be found in the appendix.

Coming back to the forested graph complex C^n, it can be viewed as a cubical complex in a similar way to the spine of Outer space in section 1.3: The k-cubes are given by graphs (G, Φ, σ) with forest size k and the faces are $k-1$-cubes obtained from collapsing an edge in Φ or removing an edge from the forest Φ. Collapsing the edge $e \in \Phi$ is on the opposite side of removing e from Φ on the k-cube. An orientation on the cube is induced by the signs from the boundary maps ∂_C and ∂_R.

To visualize this construction we consider the following example:

Example 2.20. Consider the graph J from Example 1.5 with the forest Φ given by an edge between the top and middle vertex and between the left and right vertex. Then its 2-dimensional cube is given as below:



3 Morita cycles

In this section we are going to show that a cycle exists in every C^m_{2m-4} for $m \geq 4$. For this we define the Morita graphs and show that there exists a chain of these graphs that vanishes under the boundary ∂.

Definition 3.1. A *Morita graph* $M_n(\sigma)$ for $n \in \mathbb{N}_{\geq 3}$, $\sigma \in \mathbb{S}_n$ is a forested graph (G, Φ, τ) defined as follows: G consists of two polygons with n vertices, each of which is connected to precisely one vertex of the other polygon. The vertices are labelled from 1 to n on one polygon and $n+1$ to $2n$ on the other, then the connecting edges between them are given by $(i, \sigma(i) + n)$ i.e. σ describes how the edges connecting the two polygons are permuted.

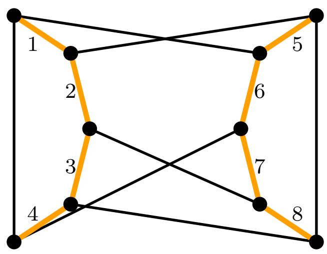
The forest Φ is given by the edges $(i, i+1)$ for $i \in \{1, \dots, n-1, n+1, \dots, 2n-1\}$ i.e. the forest consists of two linear trees of size $n-1$ each of which is obtained by removing one edge from the corresponding polygon. Finally, the ordering τ is given by

$$\tau((i, i+1)) = \begin{cases} i & \text{for } 1 \leq i \leq n-1 \\ i-1 & \text{for } n+1 \leq i \leq 2n-1 \end{cases}.$$

Notice that for all even n the graph $M_n(\sigma)$ has an odd automorphism which is given by exchanging the two polygons and applying the permutation $(1\ n)\dots(n-1\ 2n-1)$ to the orientation. Thus those Morita graphs vanish. As $M_n(\sigma)$ has $2n$ vertices and $3n$ edges by Proposition 2.4 it has rank $n+1$. Furthermore, $M_n(\sigma)$ is clearly an admissible graph and thus $M_n(\sigma) \in C^{n+1}$. Moreover as the forest size is $2n-2$ $M_n(\sigma)$ is in C^{n+1}_{2n-2}

To give a better understanding of this definition we give the following example:

Example 3.2. Below is a Morita graph of order 5 given by the permutation (12)(345).



The following result proves that there exists a cycle in every C^n for $n \geq 4$.

Theorem 3.3. *For odd $n \in \mathbb{N}_{\geq 3}$ let*

$$Z_n := \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) M_n(\sigma).$$

Then $\partial^2(Z_n) = 0$ and we call Z_n a Morita cycle.

We prove this statement in two parts first for ∂_C and then for ∂_R, from which the final result follows.

Proof for ∂_C. Let (H, Ψ, η) be an element of the chain $\partial_C Z_n$. Then as it is an element of the boundary of some Morita graph it has to have precisely one vertex of degree 4. W.l.o.g. we can assume that this vertex is in the polygon containing the edge with the label 1, as else we can apply the even permutation $(1 \dots 2n-2)^{n-1}$ to the orientation.

Let k be the label of the degree 4 vertex. An example is shown in Figure 7 on the left for $k = 2$. Then there exist exactly two Morita graphs in whose boundary H lies and whose induced vertex labeling corresponds to the one of H. One graph is obtained by splitting the vertex k into two vertices labeled k and $k+1$. This is done so that each vertex connects to one edge from the forest and to one edge from the other polygon. Moreover, the two new vertices get connected by an edge which is also added to the forest and given the number k in the ordering. All other edges with ordering number $> k$ get their ordering number increased by one. Let us denote this Morita graph by $M_n(\sigma)$. The other Morita graph is obtained in the same way, however the two edges connecting to the other polygon are permuted i.e. this graph equals $M_n(\sigma(k\ k+1))$. Examples for both are also shown in Figure 7 in the middle and the right respectively.

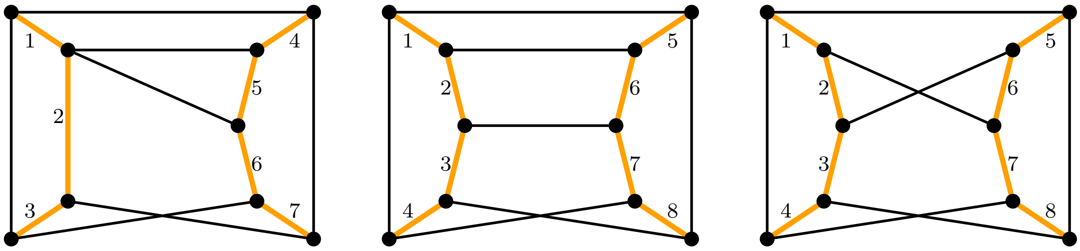


Figure 7: An element of the sum Z_n with its two possible pre-images under ∂_C.

We see that the two permutations σ and $\sigma(k\ \ k+1)$ have opposite parity and thus $M_n(\sigma)$ and $M_n(\sigma(k\ \ k+1))$ have opposite signs in Z_n. Finally we take the boundary and get for the elements corresponding to H that the one in $\partial_C(M_n(\sigma))$ is H with sign $(-1)^k$ and the element in $\partial_C(M_n(\sigma(k\ k+1))$ is also H with sign $(-1)^k$. Thus the opposite signs of $M_n(\sigma)$ and $M_n(\sigma(k\ k+1))$ in Z_n carry over and hence cancel each other.

As H was an arbitrary element of Z_n we get that every summand has coefficient 0 and that Z_n vanishes. □

Proof for ∂_R. Let (H, Ψ, η) be an element of the chain $\partial_R Z_n$. Then H is a Morita graph whose forest Ψ is missing one edge in one of its trees. We denote by σ the edge permutation of H between its two polygons as given in the definition of Morita graphs. W.l.o.g. we can assume that the tree with the missing edge contains the edge with the label 1 as else we can apply the even permutation $(1 \dots 2n-3)^{n-2}$ to the orientation.

Let $(j, j+1)$ denote the extra edge missing from Φ. An example is shown in Figure 8 on the left for $j = 2$. Then there are exactly two Morita graphs in the sum which have H in their boundary and whose induced vertex labelling corresponds to the one of H. We get one graph where the edge $(j, j+1)$ has been added to the forest with ordering number j. All other edges with ordering number $> j$ get their ordering number increased by one. Thus this this graph equals $M_n(\sigma)$. For the other Morita graph the edge $(1, n-1)$ has been added to the forest and the ordering is given by

$$\tau((i,i+1)) = \begin{cases} i-j & \text{for } j+1 \le i \le n-1 \\ n-j+i & \text{for } 1 \le i \le j-1 \\ i-1 & \text{for } n+1 \le i \le 2n-1 \end{cases} \qquad \text{and} \quad \tau(1,n-1) = n-j$$

i.e. the numbering starts with 1 at the edge $(j+1, j)$ and increases until it reaches $n-1$ on the edge $(j-1, j)$. The ordering on the other polygon has not been altered. Examples for both are also shown in Figure 8 in the middle and the right respectively.

Notice that the second graph is not part of Z_n as the vertex labelling and the forest ordering do not correspond anymore i.e. the edge with label k is not given by $(k\ \ k+1)$. To resolve this we have to change the vertex labelling which induces the permutation $(1 \dots n)^j$ on σ. Thus the element in the sum corresponding to this graph is given by $M_n(\tau) := M_n(\sigma(1 \dots n)^j)$. As

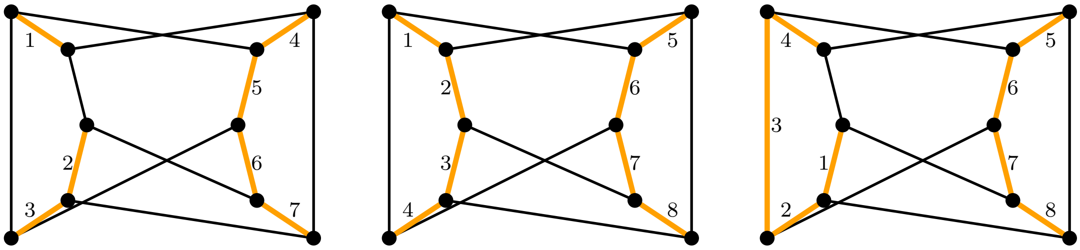


Figure 8: An element of the sum Z_n with its two possible pre-images under ∂_R.

$(1 \dots n)$ for n odd has even parity τ and σ have the same parity and thus $M_n(\tau)$ and $M_n(\sigma)$ have the same sign in the sum.

Finally, taking the boundary, the elements corresponding to H have different orientations. The element in $\partial_R(M_n(\sigma))$ is exactly H with sign $(-1)^j$. The element in $\partial_R(M_n(\tau))$ has sign $(-1)^{n-j}$ however, it differs from H by the permutation $(1 \dots n-2)^{j-1}$ with even parity as $n-2$ is odd. Now for n odd if j is even $n-j$ is odd and vice versa. Thus the elements corresponding to H have opposite signs and cancel.

As H was an arbitrary element of Z_n we get that every summand has coefficient 0 and that Z_n vanishes.

Thus we have shown the result for both ∂_C and ∂_R. As $\partial = \partial_C - \partial_R$ it also follows that $\partial Z_n = 0$. □

Remark 3.4. The Morita graphs are also included in the python implementation in the appendix and the above formula can be checked for small n. For $n = 3$ and $n = 5$ explicit isomorphism-reduced representations of the Morita cycles Z_n can be seen in Figure 9 and 10 respectively. For larger n these calculations very quickly become difficult as the number of Morita graphs is proportional to $n!$.

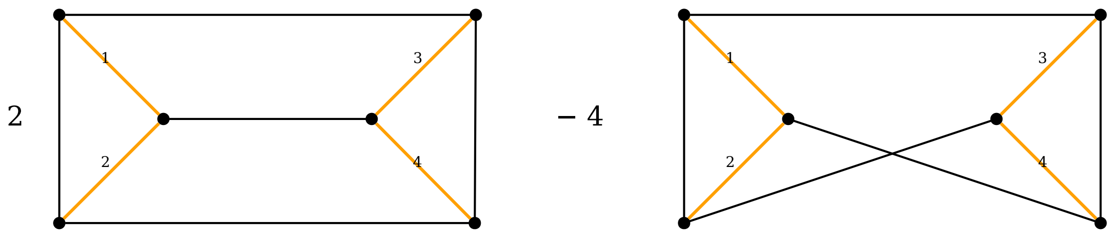


Figure 9: Isomorphism reduced version of Z_3

A more general result, showing that a similar sum vanishes for m polygons instead of just two, has been shown by Conant and Vogtmann in [4].

As we have shown that the Morita cycles vanish under the boundary the question arises if they are non-trivial in homology i.e. if they lie in the image of ∂ or not. Morita showed himself that the first cycle ($n = 3$) is non-trivial and conjectured that all of his classes are non-trivial. Conant and Vogtmann showed in [5] that the second cycle ($n = 5$) is also non-trivial. Gray extended this to show in [8] that the third cycle ($n = 7$) is non-trivial. These calculations relied partly on computer calculations whose runtime increases extremely quickly with n and are therefore not practically viable for higher n. Thus Morita's conjecture remains a challenging open problem to this day.

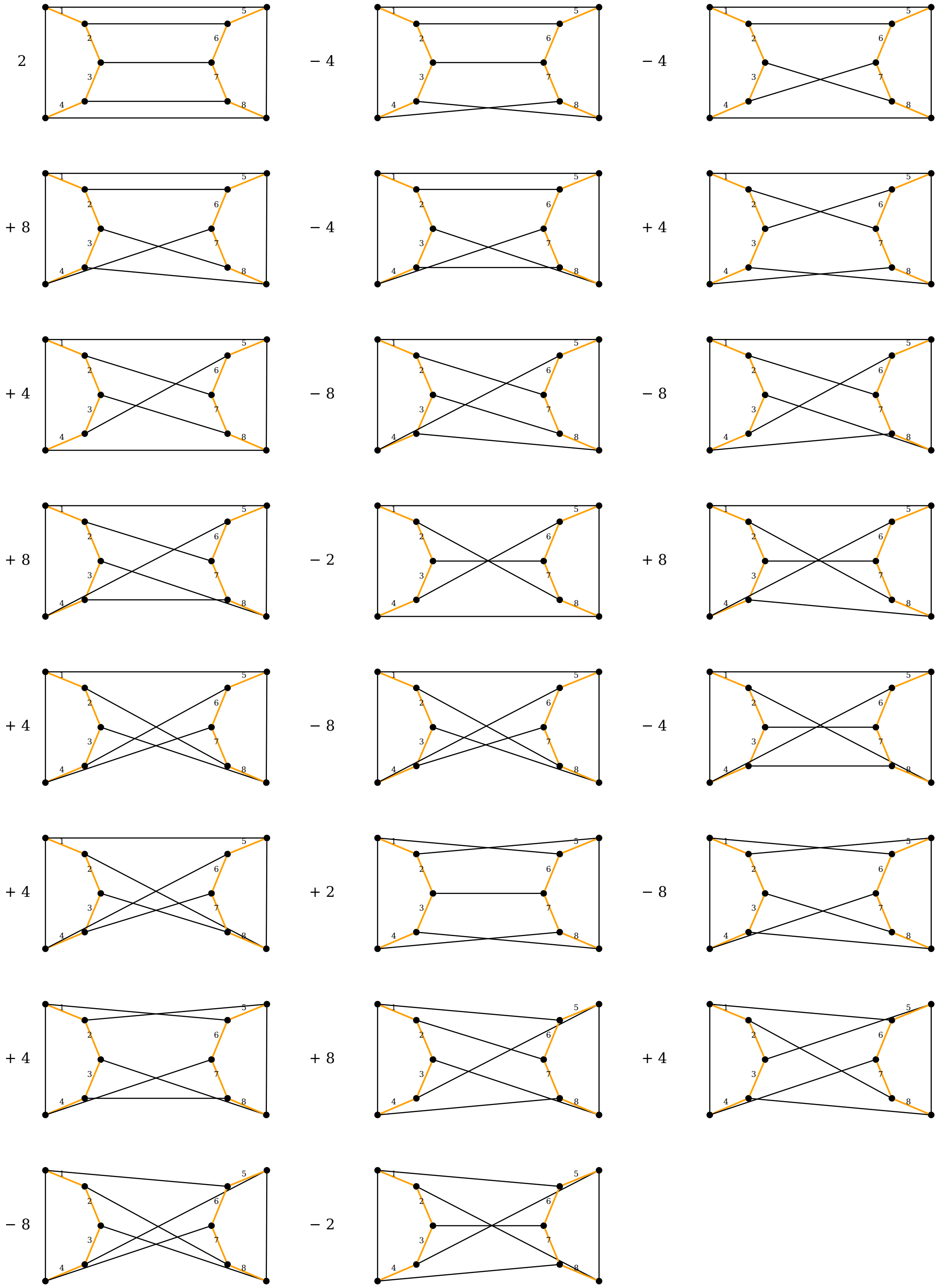


Figure 10: Isomorphism reduced version of Z_5

References

[1] Dror Bar-Natan and Brendan McKay. "Graph Cohomology - An Overview and Some Computations". unpublished preprint. `www.math.toronto.edu/drorbn/papers/GCOC/GCOC.ps`.

[2] Laurent Bartholdi. "The rational homology of the outer automorphism group of F_7". In: *New York Journal of Mathematics* 22 (2016), pp. 191–197.

[3] James Conant, Allen Hatcher, Martin Kassabov, and Karen Vogtmann. "Assembling homology classes in automorphism groups of free groups". In: *Commentarii Mathematici Helvetici* 91.4 (2016), pp. 751–806.

[4] James Conant and Karen Vogtmann. "Morita classes in the homology of $\mathrm{Aut}(F_n)$ vanish after one stabilization". In: *Groups, Geometry, and Dynamics* 2.1 (2008), pp. 121–138.

[5] James Conant and Karen Vogtmann. "Morita classes in the homology of automorphism groups of free groups". In: *Geometry & Topology* 8.3 (2004), pp. 1471–1499.

[6] James Conant and Karen Vogtmann. "On a theorem of Kontsevich". In: *Algebraic & Geometric Topology* 3.2 (2003), pp. 1167–1224.

[7] Søren Galatius. "Stable homology of automorphism groups of free groups". In: *Annals of mathematics* (2011), pp. 705–768.

[8] Jonathan Nathan Gray. "On the homology of automorphism groups of free groups." PhD thesis. University of Tennessee, 2011.

[9] Frank Harary. *Graph Theory.* Reading Massachusetts: Addison-Wesley, 1969.

[10] Allen Hatcher. *Algebraic Topology.* Cambridge: Cambridge University Press, 2001.

[11] Allen Hatcher and Karen Vogtmann. "Homology stability for outer automorphism groups of free groups". In: *Algebraic & Geometric Topology* 4.2 (2004), pp. 1253–1272.

[12] Allen Hatcher and Karen Vogtmann. "Rational homology of $\mathrm{Aut}(F_n)$". In: *Mathematical Research Letters* 5.6 (1998), pp. 759–780.

[13] Shigeyuki Morita, Takuya Sakasai, and Masaaki Suzuki. "Computations in formal symplectic geometry and characteristic classes of moduli spaces". In: *Quantum Topology* 6.1 (2015), pp. 139–182.

[14] Shigeyuki Morita, Takuya Sakasai, and Masaaki Suzuki. "Integral Euler characteristic of $\mathrm{Out}(F_{11})$". In: *Experimental Mathematics* 24.1 (2015), pp. 93–97.

[15] Jakob Nielsen. "Die Isomorphismen der allgemeinen, unendlichen Gruppe mit zwei Erzeugenden". In: *Mathematische Annalen* 78.1 (1917), pp. 385–397.

[16] Ryo Ohashi. "The Rational Homology Group of $\mathrm{Out}(F_n)$ for $n \leq 6$". In: *Experimental Mathematics* 17.2 (2008), pp. 167–179.

[17] Karen Vogtmann. "Automorphisms of Free Groups and Outer Space". In: *Geometriae Dedicata* 94.1 (2002), pp. 1–31.

[18] Karen Vogtmann. "The topology and geometry of automorphism groups of free groups". In: *Proceedings of the 2016 European Congress of Mathematicians* (2018), pp. 181–202.

[19] Karen Vogtmann and Marc Culler. "Moduli of graphs and automorphisms of free groups." In: *Inventiones mathematicae* 84 (1986), pp. 91–120.

A Python implementation

To run the ensuing code graphviz, a version of python 3 as well as the following packages are needed:

- numpy
- sympy
- pydot
- networkx

The code consists of four files. In the `boundaries.py` file ∂_C,∂_R, ∂ as well as the reduction up to isomorphism and the vanishing of graphs with odd automorphisms is implemented. The file `plotting.py` includes functions for plotting a single forested graph as well as a chain. In `MoritaCycles.py` the generation of Morita graphs $M_n(\sigma)$ as well as Morita cycles Z_n and producing positions for plotting is realized. Finally, in `main.py` a small working example reducing Z_n up to isomorphism and plotting it as well as calculating its boundary has been implemented.

A.1 main.py

```
from boundaries import *
from MoritaCycles import *
from plotting import *

n=3
path = "MCCycle" + str(n)
C = createAllMCOf(n)
C = resolveIsos(C)
C = resolveVanishing(C)
pltChain(C,lambda x: createMCPos(n,center=x),path,lineWidth=3)

dC = delta(C)
print(dC)
```

A.2 boundaries.py

```
import numpy as np
import networkx as nx
from networkx.algorithms import isomorphism as nxiso
from sympy.combinatorics import Permutation

def getForest(G):
    o = nx.get_edge_attributes(G, 'order')
    return {key: val for key, val in o.items() if val != -1}


def contractEdge(G, e):
    H = G.copy()
    H.remove_edges_from([e])
    return nx.contracted_nodes(H, e[0], e[1])
```

```python
edgeEquality = nx.isomorphism.categorical_multiedge_match("forest",
    False)


def deltaC(C):
    dC = []
    for k,G in C:
        F = getForest(G)
        dG = []
        for e, i in F.items():
            H = contractEdge(G, e)
            FH = getForest(H)
            for a, j in FH.items():
                if j > i:
                    H.edges[a]['order'] = j - 1
            dG.append([(-1) ** i * k, H])
        dC += dG
    return dC


def deltaR(C):
    dC = []
    for k,G in C:
        F = getForest(G)
        dG = []
        for e, i in F.items():
            H = G.copy()
            H.edges[e]['forest'] = False
            H.edges[e]['order'] = -1
            FH = getForest(H)
            for a, j in FH.items():
                if j > i:
                    H.edges[a]['order'] = j - 1
            dG.append([(-1) ** i * k, H])
        dC += dG
    return dC


def getForestPerm(F1, F2, g):
    F1 = {(k[0], k[1]): v for k, v in F1.items()}
    F2 = {(k[0], k[1]): v for k, v in F2.items()}
    sig = np.zeros(len(F1))
    for e, i in F1.items():
        se = (g[e[0]], g[e[1]])
        if se in F2:
            sig[i - 1] = F2[se]
        else:
            sig[i - 1] = F2[(se[1], se[0])]
    perm = Permutation(sig - 1.0)
    return (-1) ** perm.parity()
```

```python
def resolveIsos(dG):
    i = 0
    n = len(dG)
    while i < n - 1:
        H1 = dG[i]
        j = i + 1
        while j < n:
            H2 = dG[j]
            GM = nxiso.GraphMatcher(H1[1], H2[1],
   edge_match=edgeEquality)
            if GM.is_isomorphic():
                sgn = getForestPerm(getForest(H1[1]),
   getForest(H2[1]), GM.mapping)
                H1[0] += sgn * H2[0]
                del dG[j]
                n += -1
            else:
                j += 1
        i += 1

    # remove 0s
    result = [G for G in dG if G[0] != 0]
    return result

def resolveVanishing(dG):
    def hasOddAuto(H):
        GM = nxiso.GraphMatcher(H, H, edge_match=edgeEquality)
        for g in GM.isomorphisms_iter():
            F = getForest(H)
            if getForestPerm(F, F, g) == -1:
                return 1
        return 0

    result = [G for G in dG if not hasOddAuto(G[1])]
    return result


def delta(G):
    dGC = deltaC(G)
    dGR = deltaR(G)
    dGC = resolveIsos(dGC)
    dGR = resolveIsos(dGR)

    dGR = np.array(dGR, dtype=object)
    dGR[:, 0] *= -1
    dG = dGC + dGR.tolist()
    return resolveVanishing(dG)
```

A.3 plotting.py

```
import os
import networkx as nx
from boundaries import *

def pltFG(H, pos,path = "out"):
    G = H.copy()
    forest = getForest(G)
    nx.set_edge_attributes(G,{key: {"color": "red", "label": val}
  for key, val in forest.items()})
    for n, npos in pos.items():
        if(G.has_node(n)):
            G.nodes[n]['pos'] = '"%d,%d"'%(npos[0],npos[1])
            G.nodes[n]['shape'] = "point"
    p = nx.drawing.nx_pydot.to_pydot(G)
    p.write(path + ".dot")
    os.system("neato -n2 -Tpng " + path + ".dot -o " + path + ".png")

def pltChain(C,posFunction,path="out",lineWidth=5):
    masterG = nx.MultiGraph()
    hPos = 0
    for j, (count,H) in enumerate(C):
        if j!= 0 and j % lineWidth == 0:
            hPos -= 3.0
        vPos = 6 * (j % lineWidth)
        pos = posFunction((vPos, hPos))
        G = H.copy()
        nx.set_edge_attributes(G,2.0,name="penwidth")
        forest = getForest(G)
        nx.set_edge_attributes(G, {key: {"color": "#ffa000",
  "penwidth": 3, "label": val} for key, val in forest.items()})

        for node, nodePos in pos.items():
            if(G.has_node(node)):
                G.nodes[node]['pos'] = '"%d,%d"' % (nodePos[0],
  nodePos[1])
                G.nodes[node]['shape'] = "point"
                G.nodes[node]['width'] = "0.15pt"
        masterG = nx.disjoint_union(masterG,G)
        cof = str(j) + "coef"
        masterG.add_node(cof)
        if j == 0:
            masterG.nodes[cof]['label'] = "  " + str(count)
        elif count >= 0:
            masterG.nodes[cof]['label'] = "+ " + str(count)
        else:
            masterG.nodes[cof]['label'] = "\u2212 " + str(abs(count))
        masterG.nodes[cof]['shape'] = "plaintext"
        masterG.nodes[cof]['fontsize'] = "26pt"
```

```
        if j % lineWidth == 0:
            masterG.nodes[cof]['pos'] = '"%d,%d"' % (100 * (-2.5 +
    vPos), hPos * 100)
        else:
            masterG.nodes[cof]['pos'] = '"%d,%d"' % (100 * (-3.0 +
    vPos), hPos * 100)


    p = nx.drawing.nx_pydot.to_pydot(masterG)

    p.write(path + ".dot")
    os.system("neato -n2 -Tpdf " + path + ".dot -o " + path + ".pdf")
```

A.4 MoritaCycles.py

```
import numpy as np
import networkx as nx
from itertools import permutations
from sympy.combinatorics import Permutation

def createMC(n, i=0, j=0, perm = 0):
    H1 = nx.cycle_graph(np.arange(n))
    H2 = nx.cycle_graph(np.arange(n, 2 * n))
    G = nx.MultiGraph(nx.compose(H1, H2))
    if perm == 0:
        G.add_edges_from(np.arange(2 * n).reshape((2, -1)).T)
    else:
        G.add_edges_from(np.vstack((np.arange(n),perm)).T)
    nx.set_edge_attributes(G, False, 'forest')
    nx.set_edge_attributes(G, -1, 'order')

    forest = np.hstack((np.vstack((np.vstack((np.arange(n - 1),
    np.arange(1, n))).T,
                        np.vstack((np.arange(n, 2 * n - 1),
    np.arange(n + 1, 2 * n))).T)),np.zeros((2*n-2,1))))
    if i != n-1:
        forest[i] = [0, n - 1,0]
    if j != n-1:
        forest[n-1 + j] = [n,2*n-1,0]

    for i, e in enumerate(forest):
        G.edges[e]['forest'] = True
        G.edges[e]['order'] = i + 1

    return G

def createAllMCOf(n):
    dG = []
    for perm in permutations(np.arange(n,2*n)):
        G = createMC(n,n-1,n-1,perm)
        perm = Permutation(np.array(perm)-n)
```

```
            dG += [[(-1) ** perm.parity(),G]]
        return dG

def createMCPos(n,center=(0,0)):
    scale = 100.0
    lAng = np.linspace(np.pi / 2, -np.pi / 2, n)
    lPoints = scale*np.vstack((np.cos(lAng) - 2.0 + center[0],
    np.sin(lAng) + center[1])).T
    lPos = dict(enumerate(lPoints.tolist(), 0))

    rAng = np.linspace(np.pi / 2, 3 * np.pi / 2, n)
    rPoints = scale*np.vstack((np.cos(rAng) + 2.0 + center[0],
    np.sin(rAng) + center[1])).T
    rPos = dict(enumerate(rPoints.tolist(), n))
    return lPos | rPos

def rotateMCPos(pos,r):
    n = int(len(pos)/2)
    pos = np.array(list(pos.values()))
    lPos = np.roll(pos[:n],r,axis=0)
    rPos = np.roll(pos[n:], r, axis=0)
    return dict(enumerate(lPos.tolist(), 0)) |
    dict(enumerate(rPos.tolist(), n))
```