ALGORITHMIC COMPUTATION OF REIDEMEISTER TORSION

FABIO NEUGEBAUER

Abstract. We develop an algorithm that computes the Reidemeister Torsion of the geometric realization of a simplicial set X with respect to a provided representation of X's fundamental group.

The SageMath implementation of our algorithm enabled us to automatically compute the Reidemeister Torsions of the Poincaré Homology 3-Sphere with respect to all irreducible representations of its fundamental group.

Contents

1. Introduction

Reidemeister Torsion is a invariant under homeomorphisms defined for compact simplicial complexes or, more generally, CW-complexes with finitely many cells. An important application of Reidemeister torsion is to distinguish homotopy equivalent spaces which are not homeomorphic. For exactly that purpose, Kurt Reidemeister introduced Reidemeister torsion in his 1938 paper [Rei35], where he classified 3-dimensional lens spaces. Reidemeister Torsion has been extensively studied since. As explained in [Tur01] and [Coh12], Reidemeister torsion is closely linked to Whitehead torsion, which classifies simple homotopy types. Further, Reidemeister torsion has applications in knot theory, see [Mil62], [Tur01] and [Mil66]. Moreover, the equality of Reidemeister torsion and analytic torsion, proven by Cheeger in [Che77], makes Reidemeister torsion a relevant tool for studying compact Riemannian manifolds. Somewhat surprisingly, no algorithm has ever been published or implemented that automates the calculation of Reidemeister torsions.

Date: July 27, 2025.

For a compact CW-complex X and a complex representation $\rho : \pi_1(X) \to \mathrm{GL}(\mathbb{C}^n)$ of its fundamental group the Reidemeister torsion $\tau(X, \rho)$ is represented by a complex number. Our algorithm enables researchers to compute this number even if the size of X or $\pi_1(X)$ is too large to do that by hand. For example, the Poincaré homology 3-sphere is a 3-dimensional manifold whose fundamental group, the binary icosahedral group, has order 120 and consists of 9 conjugacy classes. Further, the smallest known simplicial-complex-triangulation of the Poincaré homology 3-sphere, as presented in [BL00], uses 392 simplices. While a by hand computations seems unrealistic, with our algorithm the Reidemeister torsions of the Poincaré homology 3-sphere can be computed automatically, both numerically, as well as, symbolically.

In order to create an implementable algorithm, we had to agree on a data structure that encodes the input space X from which the user wants to compute the Reidemeister torsion of. We decided that the input space X should be given as a simplicial set with finitely many non-degenerate simplices, so that our algorithm also works for simplicial and Δ-complexes that have finitely many simplices. Moreover, the triangulation as a simplicial set often gets by with fewer simplices, as with Δ- or simplicial complexes. Further, in the category of simplicial sets it is more convenient for us to construct spaces X, as it admits more morphisms then the category of Δ-complexes.

1.1. **Our Contribution.** Recently, the authors of the Paper [MBRD23] implemented the construction of the universal cover in the SageMath programming language [The23] for connected simplicial sets X with finitely many nondegenerate simplices, provided that the fundamental group π of the simplicial set X is finite and that we can solve the word problem in that particular case. In the case of an infinite fundamental group, the cellular chain complex of the universal cover can still be constructed, namely, as a chain complex of free modules over the integral group ring $\mathbb{Z}[\pi]$. Readers can find an algorithmic solution to this in Section 4 of [EK21], and an alternative description of the algorithm, better suited to the data structures in the current paper, is also provided, here, in Section 3.

Our main contributions, see section 4, is the design of algorithms for symbolic and numerical computation of combinatorial torsion of chain complexes over fields.

Combining these efforts results in an algorithm for calculating Reidemeister torsion. A SageMath implementation of the algorithms introduced in this paper is available here.

1.2. **Acknowledgments.** This project summarizes the content of my bachelor thesis. The thesis was supervised by Ulrich Bauer and Nico Stucki. All the results were achieved with their help and consultations.

2. Preliminaries

2.1. **Definition of Reidemeister Torsion.** The goal of this subsection is to recall the definition of Reidemeister torsion. But we will also hint on an outline for the structure of our algorithm. The notation introduced in this paragraph will be used throughout the paper.

We put ourselves in the following situation: We are given a simplicial set X, that has finitely many non-degenerate simplices. The geometric realization of X as a CW-complex is the space from which we want to compute the Reidemeister torsion of. For that purpose we assume that X is connected and that we are given a representation $\rho : \pi \to \mathrm{GL}(\mathbb{C}^n)$ of the fundamental group π of the geometric realization $|X|$ of X at some base-point. We assume that the base-point is the geometric realization of some vertex $x_0 \in X_0$ of X. Further, we view $\mathrm{GL}(\mathbb{C}^n)$ as the group of units of the ring $\mathrm{Mat}_{n\times n}(\mathbb{C})$ of $n \times n$ matrices with complex coefficients, so that $\rho(g)$ is a invertible matrix for all $g \in \pi$.

Recall that the *universal cover* $\tilde{X}$ is defined as a simplicial set $\tilde{X}$ together with a morphism of simplicial sets $p : \tilde{X} \to X$, such that the geometric realization of the morphism p is a universal covering projection of topological spaces. The existence of $\tilde{X}$ is justified by Theorem 2.1.1.

To compute the Reidemeister torsion $\tau(X, \rho)$ of X with respect to the representation ρ our first step will be

1) Computing the normalized chain complex $C(\tilde{X})$ of the universal cover $\tilde{X}$ of X. (This will be discussed in section 3.)

The *normalized chain complex* of a simplicial set is defined in [Lur23, Construction 00QH] and agrees with the cellular chain complex of the geometric realization of the simplicial set as CW-complex. We will now explain how the normalized chain complex $C(\tilde{X})$ of $\tilde{X}$ with coefficients in $\mathbb{Z}$ admits a left $\mathbb{Z}[\pi]$-module structure, where $\mathbb{Z}[\pi]$ is the integral group ring over the fundamental group π of X:
For that purpose we choose a vertex $\tilde{x}_0 \in \tilde{X}_0$ of the universal cover, which lies in the fiber of x_0. Recall that a deck-transformation is a morphism of simplicial sets $\phi : \tilde{X} \to \tilde{X}$ satisfying $p\phi = p$. Very importantly, it doesn't matter in that definition if we interchange the category of simplicial sets with the category of topological spaces in the following sense:

Theorem 2.1.1. [GZ12, Append. I] *Let $Cov(X)$ denote the full subcategory of the slice category of simplicial sets over X, consisting of morphisms of simplicial sets $q : R \to X$, whose geometric realization is a covering projection of topological spaces. Let $Cov(|X|)$ denote the full subcategory of the slice category of topological spaces over $|X|$, consisting of covering projections. Then, geometric realization induces an equivalence of categories $Cov(X) \to Cov(|X|)$.*

Let $g \in \pi$. For every loop $\gamma \in g$ there is a unique path $\tilde{\gamma}$ in the geometric realization of $\tilde{X}$ that lifts γ and starts at $\tilde{x}_0$. The endpoint of $\tilde{\gamma}$, constructed with that recipe, is the same for all $\gamma \in g$. Moreover, there is a unique deck-transformation ϕ_g that sends $\tilde{x}_0$ to the endpoint of $\tilde{\gamma}$. Using Theorem 2.1.1 to translate to the category of simplicial sets, these facts are proven in [Hat02, Sec.1.3] together with the following proposition:

Proposition 2.1.2. [Hat02, Prop. 1.39] *The assignment $g \mapsto \phi_g$ is a group-isomorphism from π to the subgroup of the automorphism group of $\tilde{X}$ that consists of deck-transformations.*

A left $\mathbb{Z}[\pi]$-module structure on $C(\tilde{X})$ is given by $\mathbb{Z}$-linearly extending the operation $g \cdot \tilde{\sigma} = \phi_g(\tilde{\sigma})$ that is initially given for $g \in \pi_1$ and simplices $\tilde{\sigma}$ of $\tilde{X}$. Because the ϕ_g are morphisms of simplicial sets, the boundary operators of $C(\tilde{X})$ are $\mathbb{Z}[\pi]$-linear. The next lemma will be used to give $C(\tilde{X})$ even more structure: Every chain group of $C(\tilde{X})$ will be equipped with a finite $\mathbb{Z}[\pi]$-module basis.

Lemma 2.1.3. Let $k \geq 0$ and let $\tilde{\sigma}^{(k)}, \tilde{\tau}^{(k)} \in \tilde{X}_k$ be two k-dimensional simplices of the universal cover of X, which lift the same simplex under the covering projection p, i.e. $p(\tilde{\sigma}^{(k)}) = p(\tilde{\tau}^{(k)})$. Then there exists a unique deck-transformation ϕ_g such that $\tilde{\tau}^{(k)} = \phi_g(\tilde{\sigma}^{(k)})$.

Proof. By [Hat02, Sec. 1.3] the action of π on the fiber $p^{-1}(y_0)$ via $g \cdot \tilde{y}_0 = \phi_g(\tilde{y}_0)$ is transitive and free for all vertices y_0 of X. So the lemma will be implied by the following theorem, which we cite. □

Theorem 2.1.4. [GZ12, Append. I] *Let $p : Z \to X$ be a morphism of simplicial sets. The following are equivalent:*

a) The geometric realization of the morphism p is a covering projection $|p| : |Z| \to |X|$ of topological spaces.

b) *For any simplex σ of X and vertex w of Z such that $p(w)$ is the i^{th}-vertex of σ for some i, there is a unique simplex $\tilde{\sigma}$ of Z with $p(\tilde{\sigma}) = \sigma$ such that the i-th vertex of $\tilde{\sigma}$ equals w.*

Construction 2.1.5. For any $k \geq 0$ and any k-dimensional simplex $\sigma^{(k)}$ of X choose a fixed simplex $\tilde{\sigma}^{(k)}$ of the universal cover $\tilde{X}$, such that $\tilde{\sigma}^{(k)}$ lifts $\sigma^{(k)}$, i.e. $p(\tilde{\sigma}^{(k)}) = \sigma^{(k)}$. By Lemma 2.1.3

$$\{\tilde{\sigma}^{(k)} : \sigma^{(k)} \text{ is a nondegenerate } k\text{-simplex of } X\}$$

constitutes a left $\mathbb{Z}[\pi]$-module basis of the k^{th} chain group $C_k(\tilde{X})$ of the normalized chain complex $C(\tilde{X})$.

Concretely, step 1) of computing the Reidemeister torsion $\tau(X,\rho)$ is to compute the matrices of the boundary operators of $C(\tilde{X})$ with respect to $\mathbb{Z}[\pi]$-bases, which are constructed as in 2.1.5.[1] The second step will be

2) Changing the coefficients from $\mathbb{Z}[\pi]$ to $\mathbb{C}$ to bring Reidemeister torsion in the realm of computability.

For that purpose, we make $\mathbb{C}^n$ a right $\mathbb{Z}[\pi]$-module by $\mathbb{Z}$-linearly extending the left-action of π on $\mathbb{C}^n$ via $v \cdot g = \rho(g)^{-1}v$. This allows us to define the *twist complex* $C(X,\rho) := \mathbb{C}^n \otimes_{\mathbb{Z}[\pi]} C(\tilde{X})$, which we view as a chain-complex of $\mathbb{C}$-vector spaces.

Definition 2.1.6. A *based chain-complex* C is a chain complex of finite dimensional vector spaces over a field $\mathbb{F}$ together with a fixed vector space basis of every chain group of the vector space. We refer to these bases as the distinguished bases. Further, we assume that there exists some $d \in \mathbb{N}$ such that $C_k = 0$ for $k \notin \{0, 1, \dots, d\}$.

We can promote the twist complex $C(X,\rho)$ to a based chain-complex by using a $\mathbb{Z}[\pi]$-basis of $C(\tilde{X})$ as constructed in 2.1.5:
If $\{b_i^{(k)}\}$ is a module basis of the left $\mathbb{Z}[\pi]$-module $C_k(\tilde{X})$ and $e_1, \dots, e_n$ is the standard basis of $\mathbb{C}^n$, then $\{e_s \otimes b_i^{(k)}\}_{i,s}$ constitutes a $\mathbb{C}$-vector space basis of

$$C_k(X,\rho) = \mathbb{C}^n \otimes_{\mathbb{Z}[\pi]} C_k(\tilde{X}).$$

To prove this, recall that tensor products commute with direct sums and that, for any right $\mathbb{Z}[\pi]$-module M, the module $M \otimes_{\mathbb{Z}[\pi]} \mathbb{Z}[\pi]$ is canonically isomorphic to M. Then use these canonical isomorphisms to write $C_k(X,\rho)$ as a direct sum of $\mathbb{C}^n$'s.

Convention 2.1.7. We will refer to the basis $\{e_s \otimes b_i^{(k)}\}_{i,s}$ as the *tensor product basis* obtained from the basis $\{b_i^{(k)}\}$ and the standard basis of $\mathbb{C}^n$.

The last step of computing the Reidemeister torsion $\tau(X,\rho)$ of X with respect to the representation ρ is

3) Computing the combinatorial torsion of the twist complex $C(X,\rho)$.

The *combinatorial torsion* τ of a based chain complex C over a field $\mathbb{F}$ is defined as follows: If C is not exact, τ is defined to be $0 \in \mathbb{F}$. Otherwise, let $\{c_i^{(k)}\}_{1 \leq i \leq m_k}$ be the distinguished basis

[1]Note that if the matrix $M^{(k)}$ represent $\partial_k : C_k(\tilde{X}) \to C_{k-1}(\tilde{X})$ then, because we work with left-modules, the $(i,j) - th$ entry of the matrix representing the composition $\partial_k \circ \partial_{k+1}$ is $\sum_m M^{(k+1)}_{m,j} M^{(k)}_{i,m}$ not $\sum_m M^{(k)}_{i,m} M^{(k+1)}_{m,j}$.

of C_k for $0 \leq k \leq d$ and we assumed $C_k = 0$ for all other k. For all $0 < k \leq d+1$ we choose a basis $\{a_j^{(k-1)}\}_{1\leq j\leq r_k}$ of $\operatorname{im}(\partial_k) \subseteq C_{k-1}$, where $r_k \in \mathbb{N}_0$. For every $0 < k \leq d$ the sequence

$$0 \to \operatorname{im}(\partial_{k+1}) \to C_k \xrightarrow{\partial_k} \operatorname{im}(\partial_k) \to 0 \tag{1}$$

is split exact, so there exists a set of vectors $\{a_{i+r_{k+1}}^{(k)}\}_{1\leq i\leq r_k}$ in C_k such that $\partial_k a_{i+r_{k+1}}^{(k)} = a_i^{(k-1)}$ for $1 \leq i \leq r_k$ and such that $\{a_i^{(k)}\}_{1\leq i\leq m_k}$ is a basis of C_k. We denote by $B^{(k)} \in \operatorname{Mat}_{m_k,m_k}(\mathbb{F})$ the base-change matrix, i.e. $a_j^{(k)} = \sum_{i=1}^{m_k} B_{i,j}^{(k)} c_i^{(k)}$ for all $1 \leq j \leq m_k$. The *combinatorial torsion* of the based chain complex C is defined as

$$\tau(C) := \prod_{k=0}^{d} \det(B^{(k)})^{(-1)^k} \in \mathbb{F}^*.$$

It is proven in [Tur01, Chap. 1, 1] that $\tau(C)$ is independent of the choices involved in the construction of the $a_i^{(k)}$'s. We present algorithms that compute the combinatorial torsion of based chain complexes in section 4.

Definition 2.1.8. The *Reidemeister torsion* $\tau(X,\rho)$ of X with respect to the representation $\rho : \pi \to \operatorname{Gl}(\mathbb{C}^n)$ is defined as $0 \in \mathbb{C}$ if the twist complex $C(X,\rho)$ is not exact and

$$\tau(X,\rho) := \tau(C(X,\rho)) \in \mathbb{C}^*/\pm\{\det(\rho(g)) : g \in \pi\}, \tag{2}$$

else. Here $C_k(X,\rho)$ is based by the tensor product basis obtained from a basis as chosen in 2.1.5 and the standard basis of $\mathbb{C}^n$.

Why is the Reidemeister torsion well-defined in Equation 2? Because choosing a different basis of $C(\tilde{X})$ in Construction 2.1.5 will have the same result as multiplying the torsion of the twist complex $C(X,\rho)$ by $\det(\rho(g))$ for some $g \in \pi$. Interchanging the order of such a basis, can only possibly change the sign of $\tau(C(X,\rho))$.

2.2. **Encoding Simplicial Sets.** The following data structure encoding simplicial sets is implemented in SageMath [The23, The Sage Developers]. We give a short review in preparation for the next section.

A simplicial set X is encoded by the set of its nondegenerate simplices $\{\sigma_j^{(k)}\}_{j,k}$, where we impose the condition on X that this set is finite. For every nondegenerate k-simplex $\sigma_j^{(k)} \in X_k$ of the simplicial set X the following data is stored:

(1) Its dimension $k \in \mathbb{N}_0$.
(2) The list of its $(k+1)$-faces.

A face of $\sigma_j^{(k)}$ is specified in this data structure by a nondegenerate simplex $\sigma_j^{(\lambda)}$ of X together with integers $k-1 \geq i_1 > i_2 > \cdots > i_\zeta \geq 0$, where $\zeta = k-1-\lambda$.
The corresponding face of $\sigma_j^{(k)}$ is then the $(k-1)$-simplex of X given by

$$s_{i_1} \circ s_{i_2} \circ \cdots \circ s_{i_\zeta}(\sigma_j^{(\lambda)}). \tag{3}$$

Recall that the i^{th}-degeneracy map of the simplicial set X is denoted by s_i. By [Lur23, Proposition 0014] every $(k-1)$-dimensional simplex of X can uniquely be written as in Equation 3, where we used the simplicial identity

$$s_i s_j = s_{j+1} s_i \qquad \text{if } , i \leq j, \tag{4}$$

to ensure $i_1 > i_2 > \cdots > i_\zeta$.

3. Computing the Twisted Chain Complex of the Universal Cover

First, we need to specify how we handle the fundamental group π of the geometric realization $|X|$ of X at some base-point. For that purpose we choose/compute, once and for all, a spanning tree Γ of the undirected multi-graph determined by the 1-skeleton of X. Note that the 1-skeleton of X can be viewed as a connected graph by [Lur23, Subsection 00G5]. Let $\epsilon_1, \dots, \epsilon_\nu \in X_1$ denote the nondegenerate 1-simplices of X, which are not contained in the edges of the spanning tree Γ. Let us also choose, once and for all, a base-point for the fundamental group. We let $x_0 \in X_0$ be a vertex and set the base-point of the fundamental group π to be its geometric realization $|x_0| \in |X|$. For every ϵ_μ we choose a loop γ_μ in $|X|$ based at $|x_0|$, which is contained in $|\Gamma| \cup |\epsilon_\mu|$ and traverses $|\epsilon_\mu|$ exactly once and from $|\epsilon_\mu(0)|$ to $|\epsilon_\mu(1)|$. Here $|\Gamma| \subseteq |X|$ denotes the geometric realization of the spanning tree Γ and $\epsilon_\mu(i)$ is the i^{th}-vertex of the edge $\epsilon_\mu \in X_1$.

Proposition 3.0.1 (Presentation of the fundamental group)**.** [Lee10, Chap. 10] *The map from the free group $F(\epsilon_1, \dots, \epsilon_\nu)$ on $\{\epsilon_1, \dots, \epsilon_\nu\}$ to π induced by $\epsilon_\mu \mapsto [\gamma_\mu]$ is surjective with kernel the normal closure of*

$$\left\{ z_0 \cdot z_1^{-1} \cdot z_2 \;\middle|\; \begin{array}{c} \sigma \text{ a nondegenerate 2-simplex of } X \text{ and } z_i = d_i(\sigma) \\ \text{when } d_i(\sigma) \in \{\varepsilon_1, \dots, \varepsilon_\nu\} \text{ and } z_i = 1 \text{ else.} \end{array} \right\}$$

Proposition 3.0.1 enables us to compute a group presentation of π that will be used to resemble π on the computer.

Once we specified one initial lift $\tilde{v}$ of every vertex v of X to $\tilde{X}$, we can find one normalized lift of every higher simplex σ of X, namely the unique lift of σ with 0^{th} vertex being in the set of initial lifts of vertices of X. Relative to these normalized lifts the structure of $\tilde{X}$ will be encoded. We construct the initial lifts of the vertices in a way that works well with the chosen spanning tree Γ.

Construction 3.0.2 (Initial lifts for the vertices of X)**.** We construct a set function $X_0 \to \tilde{X}_0,\ v \mapsto \tilde{v}$ with the following three properties:

a) *Lifts with respect to p:* For all vertices $y_0 \in X_0$ of X the vertex $\tilde{y} \in \tilde{X}_0$ lies in the fiber $p^{-1}(y_0)$.
b) *Base-point preserving:* $\tilde{x}_0$ is the lift of the base-point x_0 which is used in proposition 2.1.2 to construct the group-isomorphism from π to the group of deck-transformation.
c) *Induced lift of the spanning tree:* Let $\sigma^{(1)} \in X_1$ be a nondegenerate edge of X, which is contained in the edges of Γ. Let $v \in X_0$ be the 0^{th}-vertex of $\sigma^{(1)}$ and let $w \in X_0$ be the 1^{st}-vertex of $\sigma^{(1)}$. By Theorem 2.1.4 there is a unique edge $\tilde{\sigma}^{(1)} \in \tilde{X}_1$ of $\tilde{X}$ that lifts $\sigma^{(1)}$ and has 0^{th} vertex $\tilde{v}$. We require for the constructed function $X_0 \to \tilde{X}_0$ that $\tilde{\sigma}^{(1)}$ has 1^{st}-vertex equal to $\tilde{w}$.

Let us consider the geometric realization of the inclusion of Γ into X, which is a continuous map $\iota : |\Gamma| \to |X|$ from a simply-connected space $|\Gamma|$ into $|X|$. By the lifting criterion [Hat02, Prop. 1.33], there exists a (unique) continuous map $\tilde{\iota} : |\Gamma| \to |\tilde{X}|$ with $|p| \circ \tilde{\iota} = \iota$ and $\tilde{\iota}(|x_0|) = |\tilde{x}_0|$. We define a function $X_0 \to \tilde{X}_0$ by sending any vertex y_0 of X to the unique vertex $\tilde{y}_0 \in \tilde{X}_0$, which satisfies $\tilde{\iota}(|y_0|) = |\tilde{y}_0|$. This set-function $X_0 \to \tilde{X}_0$ satisfies conditions a) and b) because $\tilde{\iota}$ does. Let $\sigma^{(1)} \in X_1$ be as in condition c). Then $|\tilde{\sigma}^{(1)}|$ equals $\tilde{\iota}(|\sigma^{(1)}|)$ as map $|\Delta^1| \to \tilde{X}$ by the unique lifting criterion [Hat02, Prop. 1.34]. In particular, the point $\tilde{\iota}(|w|)$ equals the geometric realization of the 1^{st}-vertex of $\tilde{\sigma}^{(1)}$. Hence, by the definition of $w \mapsto \tilde{w}$, the vertex $\tilde{w}$ equals the 1^{st}-vertex of $\tilde{\sigma}^{(1)}$.

Convention 3.0.3 (Normalized lifts of the simplices of X)**.** From now on, let $X_0 \to \tilde{X}_0,\ v \mapsto \tilde{v}$ be as constructed in Construction 3.0.2. For every k-dimension simplex $\sigma^{(k)}$ of X, let $\tilde{\sigma}^{(k)}$ denote the unique simplex of $\tilde{X}$ having the following two properties: If $\sigma^{(k)}$ has 0^{th}-vertex y_0 then $\tilde{\sigma}^{(k)}$

has 0^{th}-vertex $\tilde{y}_0$ and, secondly, $p(\tilde{\sigma}^{(k)}) = \sigma^{(k)}$. Existence and uniqueness of $\tilde{\sigma}^{(k)}$ follows from theorem 2.1.4.

For all $k \geq 0$, the set of *normalized lifts*

$$\{\tilde{\sigma}^{(k)} : \sigma^{(k)} \text{ is a nondegenerate } k\text{-simplex of } X\} \tag{5}$$

is exactly as constructed in 2.1.5 and therefore constitutes a left $\mathbb{Z}[\pi]$-module basis of the k^{th} chain group $C_k(\tilde{X})$ of the normalized chain complex of the universal cover. The following proposition allows us to construct the matrices $M^{(1)}, \ldots, M^{(\dim(X))}$ with coefficients in $\mathbb{Z}[\pi]$, such that $M^{(k)}$ represents the boundary operator $\partial : C_k(\tilde{X}) \to C_{k-1}(\tilde{X})$ with respect to these bases.

Proposition 3.0.4 (The attaching maps of the universal cover)**.** *Let $\sigma^{(k)}$ be a k-simplex of X for $k \geq 1$ and let $\sigma^{(k-1)} := d_i(\sigma^{(k)})$ for some $i = 0, \ldots, k$. If $i > 0$, then the equation*

$$d_i(\tilde{\sigma}^{(k)}) = \tilde{\sigma}^{(k-1)} \tag{6}$$

holds in $\tilde{X}$.
Now let us consider the case $i = 0$, i.e. $\sigma^{(k-1)} = d_0\sigma^{(k)}$. Let $\sigma^{(1)} \in X_1$ be the edge of $\sigma^{(k)}$ that connects the 0^{th}-vertex v of $\sigma^{(k)}$ to the 1^{st}-vertex w of $\sigma^{(k)}$.

a) If $\sigma^{(1)}$ is degenerate or contained in the edges of the spanning tree Γ, then the equation

$$d_0(\tilde{\sigma}^{(k)}) = \tilde{\sigma}^{(k-1)} \text{ holds in } \tilde{X}. \tag{7}$$

b) If $\sigma^{(1)}$ equals one of the edges ϵ_μ not contained in the spanning tree Γ, then

$$d_0(\tilde{\sigma}^{(k)}) = \phi_g(\tilde{\sigma}^{(k-1)}), \tag{8}$$

for $g := [\gamma_\mu] \in \pi$ defined as before Proposition 3.0.1 and ϕ_g being the deck-transformation corresponding to g under the isomorphism in Proposition 2.1.2.

Proof. As our normalized lifts $\tilde{\sigma}^{(k)}$ are specified by their 0^{th} vertex being the initial lift and the d_i's preserve 0^{th} vertices for $i > 0$, equation 6 follows. Comparing 0^{th} vertices, theorem 2.1.4 implies

$$d_0(\tilde{\sigma}^{(k)}) = \phi(\tilde{\sigma}^{(k-1)}) \text{ for } \phi \text{ the unique deck-transformation with } d_0\tilde{\sigma}^{(1)} = \phi(\tilde{w}). \tag{9}$$

a) Recall that the deck-transformation ϕ is determined by its free action on the fiber of w. Thus, if $\sigma^{(1)}$ is degenerate or contained in the edges of the spanning tree Γ, then ϕ must be the identity, where in the latter case we employ Property c) in Construction 3.0.2.

b) Let us assume that $\sigma^{(1)}$ equals one of the nondegenerate edges ϵ_μ of X, which are not contained in the spanning tree Γ.
Let $W = (e_1, \ldots, e_{s-1})$ be the unique simple path in Γ from x_0 to v. We denote the corresponding sequence of vertices by $(v_1 = x_0, v_2, \ldots, v_s = v)$. By Condition 3.0.2 c) in the construction of the initial lifts $X_0 \to \tilde{X}_0$, the sequence of vertices corresponding to the walk $\tilde{W} := (\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_{s-1})$ in the graph determined by the 1-skeleton of $\tilde{X}$ is given by $(\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_s)$. We can build a continuous path

$$\tilde{\gamma}_1 := |\tilde{e}_1|^{\pm 1} \cdot |\tilde{e}_2|^{\pm 1} \cdot \cdots \cdot |\tilde{e}_{s-1}|^{\pm 1} : [0,1] \to |\tilde{X}|$$

by concatenating the geometric realizations of the $\tilde{e}_i$'s, where we choose the direction to traverse $|\tilde{e}_i|$ according to the direction of the simple path $\tilde{W}$ from $\tilde{x}_0$ to $\tilde{v}$. Here, by raising a path to the $(\cdot)^{-1}$ we will mean that we traverse the path in inverse direction. When $s = 1$, we define $\tilde{\gamma}_1$ to be constant path at $|\tilde{x}_0|$. Clearly, the following two properties hold:

(i) The continuous path $\gamma_1 := |p| \circ \tilde{\gamma}_1 : [0,1] \to |X|$ has starting point $\gamma_1(0) = |x_0|$ and endpoint $\gamma_1(1) = |v|$. Further, the image of γ_1 is contained in the geometric realization $|\Gamma|$ of the tree Γ.
(ii) The path $\tilde{\gamma}_1$ has starting point $\tilde{\gamma}_1(0) = |\tilde{x}_0|$ and endpoint the geometric realization of $\tilde{v}$.

By exactly the same argument, there exists a continuous path $\tilde{\gamma}_2 : [0,1] \to |\tilde{X}|$ such that the following two properties hold:

(i) The continuous path $\gamma_2 := |p| \circ \tilde{\gamma}_2 : [0,1] \to |X|$ has starting point $\gamma_2(0) = |x_0|$ and endpoint $\gamma_2(1) = |w|$. Further, the image of γ_2 is contained in the geometric realization $|\Gamma|$ of the tree Γ.
(ii) The path $\tilde{\gamma}_2$ has starting point $\tilde{\gamma}_2(0) = |\tilde{x}_0|$ and endpoint the geometric realization of $\tilde{w}$.

Recall that $\phi : \tilde{X} \to \tilde{X}$ is the unique deck-transformation that sends $\tilde{w}$ to the 1^{st}-vertex of $\sigma^{(1)}$. Because the endpoint of $\tilde{\gamma}_2$ equals the geometric realization of $\tilde{w}$, the concatenation

$$\tilde{\gamma} := \tilde{\gamma}_1 \cdot |\tilde{\sigma}^{(1)}| \cdot (|\phi| \circ \tilde{\gamma}_2^{-1}) : [0,1] \to \tilde{X}$$

defines a continuous path, where by $\tilde{\gamma}_2^{-1}$ the path that traverses $\tilde{\gamma}_2$ backwards is meant. By definition of $\tilde{\gamma}$, we have $\tilde{\gamma}(0) = \tilde{\gamma}_1(0) = |\tilde{x}_0|$ and $\tilde{\gamma}(1) = \phi(\tilde{\gamma}_2(0)) = \phi(|\tilde{x}_0|)$. So, the deck-transformation ϕ corresponds to $[|p| \circ \tilde{\gamma}] \in \pi$ under the isomorphism in Proposition 2.1.2. The elements $[|p| \circ \tilde{\gamma}]$ and $[\gamma_\mu]$ are equal in the fundamental group π, because both $|p| \circ \tilde{\gamma}$ and γ_μ are loops in $|X|$ based at $|x_0|$, which are contained in $|\Gamma| \cup |\epsilon_\mu|$ and traverses $|\epsilon_\mu|$ exactly once and from $|\epsilon_\mu(0)|$ to $|\epsilon_\mu(1)|$. In summary, the deck-transformations ϕ_g and ϕ agree and therefore the equation $d_0(\tilde{\sigma}^{(k)}) = \phi(\tilde{\sigma}^{(k-1)}) = \phi_g(\tilde{\sigma}^{(k-1)})$follows by the already proven claim in 9. □

We now want to adapt the previous result in Equation 8 more to the data structure introduced in Section 2.2.

Corollary 3.0.5. *Let $k \geq 2$ and $\sigma^{(k)} \in X_k$ be a k-simplex of X. Let $\sigma^{(k-1)} := d_0\sigma^{(k)}$ denote $\sigma^{(k)}$'s 0^{th}-face. Let $\tau^{(\lambda)}$ be a nondegenerate simplex of X and let $i_1 > i_2 > \cdots > i_\zeta \geq 0$ be integers such that*

$$d_k(\sigma^{(k)}) = s_{i_1} \circ s_{i_2} \circ \cdots \circ s_{i_\zeta}(\sigma^{(\lambda)}).$$

Let ϕ be the unique deck-transformation such that $d_0(\tilde{\sigma}^{(\lambda)}) = \phi(\tilde{\sigma}^{(\lambda-1)})$, where $\sigma^{(\lambda-1)} := d_0(\sigma^{(\lambda)})$ denotes the 0^{th}-face of $\sigma^{(\lambda)}$. Then

$$d_0(\tilde{\sigma}^{(k)}) = \begin{cases} \tilde{\sigma}^{(k-1)}, & \text{if } d_k(\sigma^{(k)}) \text{ is degenerate and } i_\zeta = 0, \\ \phi(\tilde{\sigma}^{(k-1)}), & \text{else.} \end{cases} \tag{10}$$

Proof. Let $\sigma^{(1)} \in X_1$ denote the edge of $\sigma^{(k)}$ that connects the 0^{th}-vertex of $\sigma^{(k)}$ to the 1^{st}-vertex of $\sigma^{(k)}$. If $d_k(\sigma^{(k)})$ is degenerate and $i_\zeta = 0$, then $\sigma^{(1)}$ is degenerate, too, and the result follows from Proposition 3.0.4 a). Otherwise, $\sigma^{(1)}$ is the edge of $\sigma^{(\lambda)}$ that connects the 0^{th}-vertex of $\sigma^{(\lambda)}$ to the 1^{th}-vertex of $\sigma^{(\lambda)}$. So, the result follows from first applying Proposition 3.0.4 to $\sigma^{(\lambda)}$ and then to $\sigma^{(k)}$. □

We now explain how we can derive algorithms from Proposition 3.0.4 that calculate the chain complex of the universal cover, the twist complex and the Reidemeister torsion, respectively. For that purpose we assume that a connected simplicial set X with finitely many nondegenerate simplices is given via the data structure explained in Section 2.2. Further, for the latter two algorithms, we assume that we are given a representation $\rho : \pi \to \mathrm{GL}(\mathbb{C}^n)$ of the fundamental group π of X. For the time being, we take for granted that ρ is an oracle that takes a loop

$\gamma : [0,1] \to |X|$ based at $|x_0|$ and returns the $n \times n$-matrix $\rho([\gamma])$. Here $|x_0| \in |X|$ is the geometric realization of a vertex of X that we use as base point of the fundamental group π.

We start our algorithm by computing a spanning tree Γ of the undirected multigraph determined by the 1-skeleton of X. For a nondegenerate 1-simplex $\sigma^{(1)} \in X$ we can read off from Proposition 3.0.4 a representation of $\partial\sigma^{(1)} \in C_0(\tilde{X})$ in terms of the $\mathbb{Z}[\pi]$-basis of $C(\tilde{X})$, which we constructed in 5, i.e. we are able to construct the matrix $M^{(1)}$ with coefficients in $\mathbb{Z}[\pi]$ that represent the boundary operator $\partial : C_1(\tilde{X}) \to C_0(\tilde{X})$ with respect to the bases defined in 5. Further, we will initialize a dictionary D with keys being all nondegenerate simplices of X. We start by filling in values of D. For any nondegenerate 1-simplex $\sigma^{(1)}$ of X we set the value $D[\sigma^{(1)}]$ to be the unique deck-transformation ϕ such that $d_0(\tilde{\sigma}^{(1)}) = \phi(\tilde{w})$ for w being the 1^{th}-vertex of $\sigma^{(1)}$.
By induction on $k \geq 1$, we assume we have filled the dictionary D for all $\lambda < k$ with the following values. The value $D[\sigma^{(\lambda)}]$ in D corresponding to a nondegenerate simplex $\sigma^{(\lambda)}$ of X is the unique deck-transformation ϕ such that $d_0(\tilde{\sigma}^{(\lambda)}) = \phi(\tilde{\sigma}^{(\lambda-1)})$, where $\sigma^{(\lambda-1)}$ is the 0^{th}-face of $\sigma^{(\lambda)}$. Using the case distinction in Equation 10 we can extend the values of D for all nondegenerate k-simplices of X, invoking the induction hypothesis.

For any nondegenerate k-simplex $\sigma^{(k)}$ we can read off a representation of $\partial(\tilde{\sigma}^{(k)})$ in terms of our $\mathbb{Z}[\pi]$-basis of $C_{k-1}(\tilde{X})$ from the completely filled dictionary D and Equation 6. This approach allows us to construct the matrices $M^{(1)}, \ldots, M^{(\dim(X))}$ with coefficients in $\mathbb{Z}[\pi]$, such that $M^{(k)}$ represents the boundary operator $\partial : C_k(\tilde{X}) \to C_{k-1}(\tilde{X})$ with respect to the bases defined in 5. This describes the algorithm computing the cellular chain complex of the universal cover.

We choose an in situ approach to obtain the twist complex. Recall that the distinguished basis of the twist complex $C(X,\rho)$ consists of the tensor product basis obtained from the basis in 5 and the standard basis of $\mathbb{C}^n$ (see Definition 2.1.6 and Convention 3.0.3 for this terminology). To construct the matrices $C^{(1)}, \ldots, C^{(\dim(X))}$ with coefficients in $\mathbb{C}$, such that $C^{(k)}$ represents the boundary operator $C_k(X,\rho) \to C_{k-1}(X,\rho)$ of the twist-complex with respect to the distinguished bases, we may use the following lemma.

Lemma 3.0.6. By $\mathbb{Z}$-linearly extending the assignment $\pi \to \mathrm{GL}(\mathbb{C}^n)$, $g \mapsto \rho(g)^{-1}$ we obtain a $\mathbb{Z}$-linear map $\tilde{\rho} : \mathbb{Z}[\pi] \to \mathrm{Mat}_{n\times n}(\mathbb{C})$. For $k = 1, \ldots, \dim(X)$ the matrix $C^{(k)}$ is a block-matrix, where the (i,j)-th submatrix of $C^{(k)}$ is the $n \times n$-matrix obtained by applying $\tilde{\rho}$ to the (i,j)-th entry of $M^{(k)}$.

Proof. This is a reformulation of the fact that the Kronecker product of matrices gives the tensor product linear map with respect to a standard choice of basis. For the sake of completeness we give an algebraic derivation. If $\{\sigma_j^{(k)}\}_j$, $\{\sigma_i^{(k-1)}\}_i$ is the list of nondegenerate k- and $(k-1)$-simplices of X, respectively, and $e_1, \ldots, e_n$ is the standard basis of $\mathbb{C}^n$, then $\{e_s \otimes \tilde{\sigma}_j^{(k)}\}_{j,s}$ and $\{e_t \otimes \tilde{\sigma}_i^{(k)}\}_{i,t}$ is the distinguished basis of the k^{th} and $(k-1)^{\text{th}}$ chain group of the twist-complex $C(X,\rho)$, respectively. We can read off the coefficients of the matrix $C^{(k)}$ representing the boundary operator $\partial : C_k(X,\rho) \to C_{k-1}(X,\rho)$ with respect to the distinguished bases from the following Equation:

$$\begin{aligned}\partial(e_s \otimes \tilde{\sigma}_j^{(k)}) &= e_s \otimes \partial(\tilde{\sigma}_j^{(k)}) = e_s \otimes \left(\sum_i M_{i,j}^{(k)} \tilde{\sigma}_i^{(k-1)}\right) = \sum_i e_s \otimes \left(M_{i,j}^{(k)} \tilde{\sigma}_i^{(k-1)}\right) \\ &= \sum_i \left(\tilde{\rho}\left(M_{i,j}^{(k)}\right) e_s\right) \otimes \tilde{\sigma}_i^{(k-1)} = \sum_i \left(\sum_{t=1}^n \tilde{\rho}\left(M_{i,j}^{(k)}\right)_{t,s} e_t\right) \otimes \tilde{\sigma}_i^{(k-1)} \\ &= \sum_i \sum_{t=1}^n \tilde{\rho}\left(M_{i,j}^{(k)}\right)_{t,s} e_t \otimes \tilde{\sigma}_i^{(k-1)}.\end{aligned}$$

□

Instead of first constructing the matrix $M^{(k)}$ and then applying $\tilde{\rho}$ coefficient-wise, we do this procedure in-place. Already when we construct $M^{(k)}$ and when we fill the dictionary D, we will replace any deck-transformation ϕ_g by the matrix $\rho(g)^{-1}$. This yields the desired algorithm computing the matrices of the twist complex $C(X, \rho)$.

The Reidemeister Torsion is then obtained by applying a function that computes the combinatorial torsion of the twist-complex from the matrices $C^{(1)}, \ldots, C^{(\dim(X))}$. Algorithms capable of this task will be introduced in Section 4.

Remark 3.0.7. We discuss two ways to implement the representation $\rho : \pi \to \mathrm{Gl}(\mathbb{C}^n)$. One possibility is that the user computes a spanning tree Γ of the 1-skeleton of X and provides the representation ρ by specifying the matrix $\rho([\gamma_\mu])$ for every edge $\epsilon_\mu \in X_1$ which is not contained in Γ. Recall that we defined the path γ_μ in the paragraph before Proposition 3.0.1.

Our second way of implementing the representation ρ tries to automate the whole process so that only the simplicial set X needs to be provided. If the fundamental group of X happens to be a finite group, then the Reidemeister torsion of X with respect to any representation ρ of π does only depend on the Reidemeister torsion's of X with respect to the irreducible subrepresentations of ρ. In that case, we use Proposition 3.0.1 to compute a presentation of π. We then ask a computer algebra system, such as GAP, to compute all irreducible representations of this presentation of π. The computer algebra system will provide the matrices $\rho(\epsilon_\mu)$ for every edge $\epsilon_\mu \in X_1 \setminus \Gamma$ and irreducible representation ρ. We then let our algorithm for the Reidemeister torsion of X run for every such irreducible representation ρ, just substituting $\rho(\epsilon_\mu)$ for the expression $\rho([\gamma_\mu])$ in line 8 of Algorithm 1.

4. Algorithms for Torsion of Chain Complexes

In the context of Reidemeister torsion we might be interested in a representation $\rho : \pi \to \mathrm{GL}(\mathbb{C}^n)$, which assign matrices with entries in some algebraic number field. In that case, we might want to implement an algorithm that computes symbolically and gives an exact result in terms of a rational function in the generators of the number field. On the other hand, we could be interested in a numerical result using floating point numbers, as this approach promises faster computations. We will present separate algorithms for each of these tasks. The symbolic calculations will be based on Gaussian elimination. This algorithm will turn out to be numerically unstable and we will do singular value decomposition instead, when constructing a numerical algorithm.

Convention 4.0.1. In this section, C is a based chain complex over a field $\mathbb{F}$ defined as in Definition 2.1.6. For $k = 1, \ldots, d$, we denote the matrix that represents the boundary operator $\partial_k : C_k \to C_{k-1}$ by $C^{(k)}$ or $C[k]$. Further, we denote the distinguished basis of C_k for $0 \le k \le d$ by $\{c_i^{(k)}\}_{1 \le i \le m_k}$.

Algorithm 1 Reidemeister torsion

Require: $\{\sigma_j^{(k)}\}$ with $k = 1, \dots, d$ and $j = 1, \dots, m_k$ the data structure of a connected simplicial set X, as defined in Section 2.2
Require: $\rho : \pi \to \mathrm{GL}(\mathbb{C}^n)$ a representation of the fundamental group π of X
Ensure: $\tau(X, \rho)$ the Reidemeister torsion of X with respect to the representation ρ

```
Compute a spanning tree Γ of the graph determined by the 1-skeleton of X
Store the nondegenerate edges ε_1, ..., ε_ν of X which are not contained in Γ
Initialize a empty dictionary D
Initialize a dictionary C with keys k = 1, ..., d and k^th-value the (n · m_{k-1}) × (n · m_k) zero
matrix                                  ▷ C encodes the boundary maps of the twist-complex.
for j = 1, ..., m_1 do
    C[1].submatrix(ni, nj, n × n) −= I_n, where d_1(σ_j^(1)) = σ_i^(0)
    if σ_j^(1) = ϵ_μ then
        Set D[σ_j^(1)] := ρ([γ_μ])^(−1), where γ_μ was defined before Proposition 3.0.1
    else
        Set D[σ_j^(1)] := I_n
    end if
    C[1].submatrix(nl, nj, n × n) += D[σ_j^(1)], where d_0(σ_j^(1)) = σ_l^(0)
end for
for k = 2, ..., d and j = 1, ..., m_k do
    for r = 1, ..., k if d_r(σ_j^(k)) is nondegenerate do
        C[k].submatrix(ni, nj, n × n) += (−1)^r ∗ I_n, where d_r(σ_j^(k)) = σ_i^(k−1)
    end for
    d_k(σ_j^(k)) = s_{i_1} ∘ s_{i_2} ∘ ··· ∘ s_{i_ς}(σ_f^(λ)) for σ_f^(λ) nondegenerate, i_1 > ··· > i_ς ≥ 0
    if i_ς = 0, set D[σ_j^(k)] := I_n, else set D[σ_j^(k)] := D[σ_f^(λ)]
    if d_0(σ_j^(k)) is nondegenerate then
        C[k].submatrix(ni, nj, n × n) += D[σ_j^(k)], where d_0(σ_j^(k)) = σ_i^(k−1)
    end if
end for
return the combinatorial torsion of a based chain complex with k^th-boundary map repre-
sented by the matrix C[k] with respect to the distinguished bases.
```

4.1. **Computing Symbolically.** In this section we present an algorithm, which computes $\pm\tau(C)$, i.e. the combinatorial torsion of a based chain complex C, up to a sign. Essentially, we implement matrix τ-chains as defined by Turaev in [Tur86]. The algorithm will be suitable for symbolical computations.

Definition 4.1.1. Let $A \in \mathrm{Mat}_{n\times m}(\mathbb{F})$ be a matrix and say A has column vectors $v_1, \dots, v_m \in \mathbb{F}^n$. A *set of pivots* of A is a subset P of $\{1, \dots, m\}$ such that $\{v_i : i \in P\}$ is a basis of the *column space* $\mathrm{span}_{\mathbb{F}}\{v_1, \dots, v_m\} \subseteq \mathbb{F}^n$ of A.

For a matrix $A \in \mathrm{Mat}_{n\times m}(\mathbb{F})$ a set of pivots P of A is given by the position of the first nonzero entry in each row of a row echelon form of A. When we ask for a set of pivots of a matrix A in side of an algorithm, we implicitly mean that a row echelon form of A is computed and a set of pivots of A is obtained from it. Here, the transformation matrix, which used to bring A in row echelon form, is not computed.

The first step of our algorithm will be computing a set of pivots $P[k]$ of the matrix $C[k]$. Then we check, whether C is exact via the following lemma.

Lemma 4.1.2. The chain complex C is exact if and only if for $k = 0, \dots, d$ the inequality

$$\operatorname{rank}(C^{(k)}) + \operatorname{rank}(C^{(k+1)}) \geq \dim(C_k)$$

holds, where it is understood that $C^{(d+1)} = C^{(0)} = 0$.

Proof. C is exact at k if and only if $\operatorname{rank}(C^{(k+1)}) \geq \dim(\ker(C^{(k)}))$ and the latter equals $\dim(C_k) - \operatorname{rank}(C^{(k)})$ by the rank-nullity theorem. □

From here on we can assume that C is exact, returning 0 otherwise. Now we let the integer k run backwards $k = d, d-1, \dots, 2$. We will set $r_k := \operatorname{rank}(C^{(k)})$, which is equal to the cardinality of $P[k]$. For i in $\{1, \dots, r_k\}$ and j the i^{th} biggest integer in $P[k]$ we define $a_i^{(k-1)} := \partial(c_j^{(k)})$. Then $\{a_i^{(k-1)}\}_{1 \leq i \leq r_k}$ is a basis of $\operatorname{im}(\partial_k) \subseteq C_{k-1}$. Further, we define $\{a_{i+r_{k+1}}^{(k)}\}_{1 \leq i \leq r_k} := \{c_i^{(k)} : i \in P[k]\}$ with the convention $r_{d+1} = 0$.

Lemma 4.1.3. For $k = 1, \dots, d$, the set $\{a_i^{(k)}\}_{1 \leq i \leq m_k}$ is a basis of C_k.

Proof. By construction $\{a_i^{(k)}\}_{1 \leq i \leq r_{k+1}}$ is a basis of $\operatorname{im}(\partial_{k+1}) \subseteq C_k$ and $\{\partial a_{i+r_{k+1}}^{(k)}\}_{1 \leq i \leq r_k}$ is a basis of $\operatorname{im}(\partial_k) \subseteq C_{k-1}$. The result follows because the Sequence 1 is split exact. □

Let $k \in \{0, \dots, d\}$. We denote by $B^{(k)} \in \operatorname{Mat}_{m_k, m_k}(\mathbb{F})$ the base-change matrix from $\{c_i^{(k)}\}$ to $\{a_j^{(k)}\}$, i.e. $a_j^{(k)} = \sum_{i=1}^{m_k} B_{i,j}^{(k)} c_i^{(k)}$ for all $1 \leq j \leq m_k$. The first r_{k+1} columns of $B^{(k)}$ consist of those columns of $C^{(k+1)}$, whose index lies $P[k+1]$. The set of the last r_k column vectors of $B^{(k)}$ equals $\{e_i : i \in P[k]\} \subseteq \mathbb{F}^{m_k}$, where $e_1, \dots, e_{m_k}$ are the standard-basis vectors of $\mathbb{F}^{m_k}$. We compute $\det(B^{(k)})$ by iterated Laplace expansion to get rid of the last r_k columns of $B^{(k)}$. Consequently, up to a sign, the determinant of $B^{(k)}$ equals the determinant of the matrix, which is obtained from $C^{(k+1)}$ by deleting the columns, whose index doesn't lie in $P[k+1]$, and deleting the rows, whose index lies in $P[k]$. This yields Algorithm 2.

Algorithm 2 Torsion of Chain Complexes with Gaussian Elimination

Require: A based chain complex C as defined in definition 2.1.6.
In detail, C is a dictionary with keys $1, \dots, d$ such that for all keys $1 \leq k \leq d$ the corresponding value $C[k]$ is the matrix representing the boundary map $\partial : C_k \to C_{k-1}$ with respect to the distinguished bases.
Ensure: Up to a sign, the combinatorial torsion of the based chain complex C.

Initialize a dictionary P by setting $P[d+1]$ and $P[0]$ to be the empty set
for $k = d, d-1, \dots, 2, 1$ **do**
  Extend the dictionary P by setting $P[k]$ to be a set of pivots of the matrix $C[k]$
  if $\#(P[k]) + \#(P[k+1]) < m_k$ **then**
    return 0
  end if
end for
if $\#(P[1]) < m_0$ **then**
  return 0
end if
Set $\tau := 1$ ▷ This is the determinant of $B^{(d)}$.
for $k = d-1, d-2, \dots, 1, 0$ **do**
  Delete the columns of $C[k+1]$ with index not in $P[k+1]$
  Delete the rows of $C[k+1]$ with index in $P[k]$.
  Set $\tau = \tau * \det(C[k+1])^{(-1)^k}$ ▷ Compute this via LU-decomposition.
end for
return τ

Algorithm 2 does Gaussian elimination twice per matrix $C^{(k)}$. It is natural to ask if we can find an alternative algorithm, in which we only do it once.

We claim that Algorithm 2 and Algorithm 3 compute the same numbers, up to possibly a different choice of pivots:

Assume that $P[k]$ is a set of pivots of $C^{(k)}$. The projection

$$\mathrm{pr} : C_k \to \mathrm{span}_{\mathbb{F}}\{c_i^{(k)} : i \notin P[k]\}, \quad \sum_{i=1}^{m_k} \alpha_i c_i^{(k)} \mapsto \sum_{i \notin P[k]}^{m_k} \alpha_i c_i^{(k)}$$

is injective restricted to $\mathrm{im}(\partial_{k+1})$ as $\ker(\partial_k) \cap \ker(\mathrm{pr}) = \{0\}$ and $\mathrm{im}(\partial_{k+1}) \subseteq \ker(\partial_k)$. So, a set of pivots of the matrix representing $\mathrm{pr} \circ \partial_{k+1}$ with respect to the distinguished bases is a set of pivots of $C^{(k+1)}$. Hence, if $\mathrm{Piv}[k]$ is a set of pivots of $C^{(k)}$, then in the for-loop of Algorithm 3 a set of pivots of $C^{(k+1)}$ is written into $\mathrm{Piv}[k+1]$. Further, by Lemma 4.1.2, the chain complex C is exact at C_k if and only if $\mathrm{pr} \circ \partial_{k+1}$ is surjective, which is checked in line 8 of Algorithm 3. Lastly, up to sign, the value of z in line 16 equals the determinant of a matrix, which is obtained from $C^{(k+1)}$ by deleting the rows with index in $\mathrm{Piv}[k]$ and deleting the columns with index not contained in $\mathrm{Piv}[k+1]$.

Remark 4.1.4. When computing the Reidemeister torsion for a simplicial set X in cases with $\#X_k >> k$ we recommend to implement the above algorithms using sparse matrices.

4.2. **Computing Numerically.** In this subsection, as in Convention 4.0.1, we assume that C is a based chain complex in the sense of Definition 2.1.6 and we further assume that the field $\mathbb{F}$ over which C is defined equals (is contained in) the field of complex numbers $\mathbb{C}$.

We will now explain the following preliminary algorithm.

Algorithm 3 Torsion of Chain Complexes: One LU-decomposition per chain group

Require: A based chain complex C as defined in definition 2.1.6.
In detail, C is a dictionary with keys $1,\dots,d$ such that for all keys $1 \le k \le d$ the corresponding value $C[k]$ is the matrix representing the boundary map $\partial : C_k \to C_{k-1}$ with respect to the distinguished bases.
Ensure: Up to a sign, the combinatorial torsion of the based chain complex C.

1: Set $\tau := 1 \in \mathbb{F}$
2: Initialize a dictionary Piv with keys $0,\dots,d$ and all values empty lists
3: **for** $k = 0,1,\dots,d-2,d-1$ **do**
4:     Delete the rows of $C^{(k+1)}$, whose row-index is contained in $\mathrm{Piv}[k]$
5:     Compute a LU-decomposition of $C^{(k+1)}$: $C^{(k+1)} = PLU$
6:     Set $z := 1$
7:     **for** $i = 1,\dots,r$, with r the number of rows of $C^{(k+1)}$ **do**
8:         **if** the i^{th} row-vector of U is the zero vector **then**
9:             **return** 0
10:         **else**
11:             Let j be the column-index of the first nonzero entry of the i^{th} row of U
12:             Set $z = z * U_{i,j}$
13:             Append the integer j to the list $\mathrm{Piv}[k+1]$
14:         **end if**
15:     **end for**
16:     Set $\tau = \tau * z^{(-1)^k}$
17: **end for**
18: **if** $\#(\mathrm{Piv}[d]) < m_d$ **then**
19:     **return** 0
20: **else**
21:     **return** τ
22: **end if**

The dictionary Ra has values $\mathrm{Ra}[k] = \mathrm{rank}(C[k])$ for all keys $k \in \{d, d-1, \dots, 1\}$, as long C is exact at k. By Lemma 4.1.2, the pseudocode in line 5 constitutes a test whether C is exact at C_{k+1} and the code in line 12 constitutes a test whether C is exact at C_0. Now the combinatorial torsion of C is computed by an inductive process $k \to k-1$. In every step, we extend a basis of $\mathrm{im}(\partial_{k+1})$ to a basis of C_k, then we take the image under ∂_k of that extended part as a basis of $\mathrm{im}(\partial_k)$ and proceed. The coefficients of the extended part with respect to the distinguished basis of C_k will be stored in $\mathrm{Ex}[k]$. The coefficients of the image of the extended part with respect to the distinguished basis of C_{k-1} will be stored in $\mathrm{Im}[k-1]$.

We extend the basis $\mathrm{im}(\partial_{d+1}) = \{0\}$ to a basis of C_d by the distinguished basis itself, so that $\mathrm{Ex}[d]$ is the identity matrix. For the code in line 10 we use the fact that kernel of the Hermitian adjoint equals the orthogonal complement of the image. So, in line 10 we actually extend a basis of $\mathrm{im}(\partial_{k+1})$ via an basis of $\mathrm{im}(\partial_{k+1})^{\perp}$ to a basis of C_k. Using these chosen bases, the number in line 13 is, by definition, exactly the Reidemeister torsion of the based chain complex C.

Now we derive Algorithm 5 from the preliminary Algorithm 4. The advantage of Algorithm 5 is that it is numerically stable and it suffices to do the following computations per chain group C_k: A singular value decomposition $C[k] = U \cdot \Sigma \cdot V^{\dagger}$, the determinant calculations $\det(U)$, $\det(W)$ and a matrix product $V^{\dagger} \cdot (\mathrm{Ex})$.

Algorithm 4 Preliminary Algorithm: Torsion of Chain Complexes over $\mathbb{C}$

Require: A based chain complex C over $\mathbb{C}$ as defined in Definition 2.1.6.
In detail, C is a dictionary with keys $1, \dots, d$ such that for all keys $1 \leq k \leq d$ the corresponding value $C[k] \in \mathrm{Mat}_{m_{k-1},m_k}(\mathbb{C})$ is the matrix representing the boundary map $\partial : C_k \to C_{k-1}$ with respect to the distinguished bases.
Ensure: The combinatorial torsion of the based chain complex C.

```
1:  Initialize a dictionary Ra := {d : m_d}
2:  Initialize a dictionary Ex = {d : I_{m_d} ∈ Mat_{m_d,m_d}(C)}      ▷ I_{m_d} the identity matrix
3:  Initialize a dictionary Im = {}                                   ▷ The empty dictionary
4:  for k = d − 1, d − 2, ..., 1, 0 do
5:      if rank(C[k + 1]) < Ra[k + 1] then
6:          return 0
7:      end if
8:      Extend the dictionary Ra by Ra[k] := m_k − Ra[k + 1]
9:      Extend the dictionary Im by Im[k] := C[k + 1] · (Ex[k + 1])
10:     Extend the dictionary Ex by a full-rank matrix Ex[k] ∈ Mat_{m_k,Ra[k]}(C) with (Im[k])^† ·
    (Ex[k]) = 0
11: end for
12: if Ra[0] == 0 then
13:     return ∏_{k=0}^{d−1} det(Im[k] | Ex[k])^{(−1)^k}
14: else
15:     return 0
16: end if
```

The if-clause in line 6 of Algorithm 5 serves exactly the same purpose as the if-clause in line 5 of algorithm 4. As, also, the if-condition in the end of both algorithms agree, we might assume in deriving Algorithm 5 that the chain complex C is exact.

Let us put ourselves in the situation of the for-loop of Algorithm 5. We identify the matrix Ex with $\mathrm{Ex}[k+1]$ in Algorithm 5. This will be justified inductively $k \to k-1$ and is clear for $k = d$. If $C[k+1] = U\Sigma V^\dagger$ is a singular value decomposition as in Algorithm 5, then $(C[k+1] \cdot (\mathrm{Ex}))^\dagger = (\mathrm{Ex})^\dagger \cdot V \cdot \Sigma \cdot U^\dagger$. For now let us denote the matrix defined in line 12 by M. Because, $U^\dagger U$ is the identity matrix, the matrix M is a full rank-matrix with $(C[k+1] \cdot (\mathrm{Ex}))^\dagger \cdot W = 0$, where we used that the last $\mathrm{Ra}[k]$-column-vectors of Σ are zero vectors. This justifies that the preliminary Algorithm 4 would return the correct result, when for every $k \in \{1, \dots, d\}$ we set $\mathrm{Ex}[k+1] = \mathrm{Ex}$, $\mathrm{Ex}[k] = M$ and $\mathrm{Im}[k] = C[k+1] \cdot (\mathrm{Ex})$, where the matrices Ex and M come from the for-loop of Algorithm 5. Using this translation from Algorithm 4 to Algorithm 5 we obtain that,

$$\begin{aligned}
\det(\mathrm{Im}[k] \,|\, \mathrm{Ex}[k]) &= \det\left(C[k+1] \cdot (\mathrm{Ex}) | M\right) = \det\left(U\Sigma V^\dagger(\mathrm{Ex}) \middle| U \cdot \begin{pmatrix} 0 \\ I_{\mathrm{Ra}[k]} \end{pmatrix}\right) \\
&= \det(U) \det\begin{pmatrix} \mathrm{diag}(\zeta_1, \dots, \zeta_{\mathrm{Ra}[k+1]})W & 0 \\ 0 & I_{\mathrm{Ra}[k]} \end{pmatrix} \\
&= \det(U) \cdot \left(\prod_{i=1}^{\mathrm{Ra}[k+1]} \zeta_i\right) \cdot \det(W),
\end{aligned}$$

which proves the correctness of Algorithm 5.

Algorithm 5 Torsion of Chain Complexes with Singular Value Decomposition

Require: A based chain complex C over $\mathbb{C}$ as defined in Definition 2.1.6.
In detail, C is a dictionary with keys $1, \dots, d$ such that for all keys $1 \leq k \leq d$ the corresponding value $C[k] \in \mathrm{Mat}_{m_{k-1}, m_k}(\mathbb{C})$ is the matrix representing the boundary map $\partial : C_k \to C_{k-1}$ with respect to the distinguished bases.
Ensure: The combinatorial torsion of the based chain complex C.

```
Initialize a dictionary Ra := {d : m_d}
Set Ex := I_{m_d} ∈ Mat_{m_d,m_d}(ℂ)                    ▷ I_{m_d} the identity matrix.
Set τ := 1
for k = d − 1, d − 2, ..., 1, 0 do
    Compute a singular value decomposition C[k + 1] = UΣV^† with singular values
    ζ_1, ..., ζ_{min{m_k, m_{k+1}}}, ordered by absolute value in decreasing order
    if Ra[k + 1] > m_k or (Ra[k + 1] > 0 and ζ_{Ra[k+1]} = 0) then
        return 0
    end if
    Extend the dictionary Ra by Ra[k] := m_k − Ra[k + 1]
    Let W ∈ Mat_{Ra[k+1],Ra[k+1]}(ℂ) be the submatrix of V^† · (Ex) consisting of first Ra[k+1]-
    rows of V^† · (Ex)
    Set τ = τ · det(U)^{(−1)^k} · (∏_{i=1}^{r_k} ζ_i)^{(−1)^k} · det(W)^{(−1)^k}
    Set Ex ∈ Mat_{m_k,Ra[k]}(ℂ) to consist of the last Ra[k]-columns of U
end for
if Ra[0] == 0 then
    return τ
else
    return 0
end if
```

5. Implementation and Results

For $\zeta_5 := e^{2\pi i/5} \in \mathbb{C}$ the Reidemeister torsion of the Poincaré homology 3-sphere with respect to the 9 irreducible representations of its fundamental group are $0, 1, \frac{2}{3}, \frac{3}{5}, \frac{1}{2}$ and

$$\begin{aligned} 4 + 2\zeta_5^2 + 2\zeta_5^3 &\approx 0.763932022500211, \\ -2 + 2\zeta_5^2 + 2\zeta_5^3 &\approx 5.23606797749979, \\ -1 + \frac{1}{2}\zeta_5^2 + \frac{1}{2}\zeta_5^3 &\approx 1.80901699437495, \\ \frac{3}{2} + \frac{1}{2}\zeta_5^2 + \frac{1}{2}\zeta_5^3 &\approx 0.690983005625053, \end{aligned}$$

which are real-numbers but not all of them are algebraic integers. These result were computed via the algorithm developed above. While the symbolic calculation of that 9 results in one run from the data of the simplicial triangulation takes approximately $135 - 288$ seconds and the numerical algorithm runs in $61 - 81$ second. This computation have been performed from a Jupyter Notebook on a Windows Subsystem for Linux on a laptop with 16 GB RAM and 2.00 GHz processor.

Recall the following fact on Reidemeister torsion Let X, Y be CW-complexes and $\phi : \pi_1(X) \to \mathrm{GL}(\mathbb{C}^n)$, $\psi : \pi_1(Y) \to \mathrm{GL}(\mathbb{C}^m)$ representations of their fundamental groups. Then, we have for the Reidemeister torsions

$$\tau(X\times Y,\phi\otimes\psi)=\tau(X,\phi)^{\chi(Y)}\tau(Y,\psi)^{\chi(X)},$$

with the usual convention $0^0=1$. For reference see [Fre92].We successfully tested our algorithm by comparing the above statement to our direct computation in the following examples:

$$S^2\times\mathbb{R}P^3,\ L(3;1,1)\times\mathbb{R}P^2,\ L(3;1,1)\times S^2, L(3;1,2)\times\mathbb{R}P^2,\ L(3;1,2)\times S^2,$$
$$L(3;1,1)\times L(3;1,2),\ L(3;1,2)\times\mathbb{C}P^2,\ \mathbb{R}P^3\times\mathbb{C}P^2,$$

where $L(q;r_1,\dots,r_n)$ is the notation for lens spaces used in Turaev's book [Tur01]. We obtained a algorithmic construction of general lens spaces by implementing joins of simplicial sets: The lens space $L(q;r_1,\dots,r_n)$ is the pushout of the identity morphism of $S^{2n-1}\cong S^1\star\cdots\star S^1$ and the n-fold join $\rho_{r_1}\star\cdots\star\rho_{r_n}$. Here $\rho_{r_i}:S^1\to S^1$ denotes the automorphism of the directed cycle graph with q vertices, which maps the j^{th}-vertex to the $(r_i+j \mod q)^{\text{th}}$-vertex.

5.1. **Outlook.** Given a homotopy equivalence, with out loss of generality an inclusion of a pair $X\hookrightarrow Y$, we can inspect its Whitehead torsion. A modificaton of our algorithm allows to check for triviality of the Reidemeister torsion of the pair (Y,X), which in turn will imply finite order of (Y,X) in the geometric Whitehead group, see [Mil66, Theorem 8.1]. The geometric Whitehead group is described in [Coh12, Chapter II.§6]. It would be interesting to obtain a description of the simple homotopy equivalence witnessing the finite order of (Y,X) in the Whitehead group of X. For example, one might like to find a discrete vector field encoding this.

Another natural question emerging from the results in this paper, is to find a theoretical condition for the Reidemeister torsion being an algebraic integer. While our computation of the Reidemeister torsion of the Poincaré homology 3-sphere gives an example of a 3-manifold where this is not the case, in the recent paper [KN22] the authors provide a large class of 3-manifolds whose Reidemeister torsions are algebraic integers.

References

[BL00] Anders Björner and Frank H Lutz. Simplicial manifolds, bistellar flips and a 16-vertex triangulation of the Poincaré homology 3-sphere. *Experimental Mathematics*, 9(2):275–289, 2000. 2

[Che77] Jeff Cheeger. Analytic torsion and Reidemeister torsion. *Proceedings of the National Academy of Sciences*, 74(7):2651–2654, 1977. 1

[Coh12] Marshall M Cohen. *A course in simple-homotopy theory*, volume 10. Springer Science & Business Media, 2012. 1, 17

[EK21] Graham Ellis and Kelvin Killeen. Cohomology with local coefficients and knotted manifolds. *Journal of Symbolic Computation*, 107:299–321, 2021. 2

[Fre92] Daniel S Freed. Reidemeister torsion, spectral sequences, and brieskorn spheres. 1992. 17

[GZ12] Peter Gabriel and Michel Zisman. *Calculus of fractions and homotopy theory*, volume 35. Springer Science & Business Media, 2012. 3

[Hat02] Allen Hatcher. Algebraic topology. *Available on the author's webpage*, 2002. URL: `https://pi.math.cornell.edu/~hatcher/AT/ATpage.html`. 3, 6

[KN22] Teruaki Kitano and Yuta Nozaki. An algebraic property of Reidemeister torsion. *Transactions of the London Mathematical Society*, 9(1):136–157, 2022. 17

[Lee10] John Lee. *Introduction to topological manifolds*, volume 202. Springer Science & Business Media, 2010. 6

[Lur23] Jacob Lurie. Kerodon. `https://kerodon.net`, 2023. 3, 5, 6

[MBRD23] Miguel A Marco-Buzunáriz, Ana Romero, and Jose Divasón. Effective homology of universal covers. *ACM Communications in Computer Algebra*, 57(2):47–50, 2023. 2

[Mil62] John Milnor. A duality theorem for Reidemeister torsion. *Annals of Mathematics*, pages 137–147, 1962. 1

[Mil66] John Milnor. Whitehead torsion. *Bulletin of the American Mathematical Society*, 72(3):358–426, 1966. 1, 17

[Rei35] Kurt Reidemeister. Homotopieringe und Linsenräume. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 11, pages 102–109, 1935. 1

[The23] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 10.1)*, 2023. URL: https://www.sagemath.org. 2, 5

[Tur86] Vladimir G Turaev. Reidemeister torsion in knot theory. *Russian Mathematical Surveys*, 41(1):119, 1986. 11

[Tur01] Vladimir Turaev. *Introduction to combinatorial torsions*. Springer, 2001. 1, 5, 17