*Journal of Homotopy and Related Structures, vol. ??(??), ????, pp.1–21*

# COMPUTING BREDON HOMOLOGY OF GROUPS

BUI ANH TUAN AND GRAHAM ELLIS

*Abstract*

We describe the basic ingredients of a general computational framework for performing machine calculations in the cohomology of groups. This has been implemented in the GAP system for computational algebra and the paper is intended to aid those wishing to extend that implementation to their own needs.

*Dedicated to Ronnie Brown on his 80th birthday*

## 1. Introduction

Explicit calculations play a role in the theoretical development of the cohomology of groups and it is becoming more common for such calculations to be derived with the aid of a computer. In this paper we describe a basic framework for performing machine computations in the cohomology of groups. This has been implemented in the HAP package [**16**] for the GAP system for computational algebra [**21**] and the paper is intended to aid those wishing to extend that implementation to their own needs. The framework is extremely simple, involving just two ingredients: (i) a data type for a computer representation of Bredon modules that permits the easy implementation and easy combination of a range of homological algorithms; (ii) an emphasis on explicit homotopies as a means of translating non-constructive homological proofs into constructive homological algorithms. The paper is thus no more than a description of a single data type reflecting a particular choice of approach – the explicit homotopy approach – to implementing homological algorithms.

Given an implementation of our Bredon module data type it is a triviality to extract the corresponding Bredon homology. We view the standard Eilenberg-MacLane homology of a group as a special instance of Bredon homology. The data type has been used in [**12, 11, 3, 19, 25**] to calculate the low dimensional integral group homology $H_n(G, \mathbb{Z})$ of groups such as $PSL_4(\mathbb{Z})$, $M_{24}$, $SL_2(\mathbb{Z}[1/6])$, various Artin groups and various Bieberbach groups. In [**17**] the data type is used to provide examples of infinite families of finite $p$-groups $G$ for which a single finite computation determines the mod-$p$ homology $H_n(G, \mathbb{Z}_p)$ for all $n \geqslant 0$ and all $G$ in a family. These papers describe the mathematics underlying the calculations and omit details of the data type. Other calculations in the literature can also be recovered using the

---

.
2000 Mathematics Subject Classification: 20J06
Key words and phrases: Bredon homology and cohomology, Bredon module, classifying space for proper actions, cohomology of groups

data type. We are grateful to Alexander Rahm and Rubén Sánchez-García for contributing to the HAP package implementations of the data type that allow package users to recover calculations on the Bredon homology of a group $G$ with coefficients in the complex representation ring, for $G$ equal to $SL_3(\mathbb{Z})$ [**32**] and $G$ equal to various Bianchi groups [**26**] and Coxeter groups [**31, 28**]. We are grateful to Sebastian Schönnenbeck for contributing further implementations for Bianchi groups [**33**]. We are grateful to Mathieu Dutour-Sikirić for contributing implementations for groups such as $PSL(3,\mathbb{Z}[\mathbf{i}])$, $PSL(3,\mathcal{O}_{Eis})$ with $\mathcal{O}_{Eis}$ the Eisenstein integers, $PSL_4(\mathbb{Z})$, $Sp(4,\mathbb{Z})$. The first author has contributed an implementation of the data type for calculating Bredon homology of crystallographic groups.

The paper is structured as follows. In Section 2 we indicate more clearly the kind of homological computations covered by our computational framework. These include standard Eilenberg-MacLane homology of a group $G$ with coefficients in a finitely generated $\mathbb{Z}G$-module, as well as Bredon homology of the classifying space for proper actions $\underline{E}G$ with coefficients in a Bredon module such as the complex representation ring or the Burnside ring. In Section 3 we recall some basic definitions from the cohomology of groups. In Section 4 we describe our data type for representing Bredon modules; this incorporates a data type for representing rigid $G$-CW spaces. In Section 5 we review some implementations of this Bredon module data type that are currently available in HAP. In Section 6 we summarize the role of explicit homotopies in constructive homological algebra. Section 7 details methods for *rigidifying* $G$-CW spaces so that they can be used in implementations of the Bredon module data type.

## 2. Some typical homological computations

We use the term *$G$-CW space* to mean a CW space $X$ on which a discrete group $G$ acts in such a way that the action induces a permutation of the cells of $X$. We say the space is *rigid* if each element in the stabilizer of any cell fixes the cell pointwise. (Our terminology differs slightly from standard usage where rigidity is required of all $G$-CW spaces.) A *family* of subgroups of $G$ is any collection $\mathcal{F}$ of subgroups of $G$ that is closed under conjugation and taking subgroups. Thus $H \in \mathcal{F}$ implies that $gHg^{-1} \in \mathcal{F}$ for all $g \in G$ and that all subgroups of $H$ are in $\mathcal{F}$. We say that a $G$-CW space $X$ is an *$\mathcal{F}$-space* if it is rigid and for each cell $e \subset X$ the *stabilizer group* $Stab_G(e) = \{g \in G \mid {}^g e = e\}$ belongs to $\mathcal{F}$. By a *morphism* $\phi\colon X \to Y$ between two $G$-CW spaces we mean a cellular map satisfying $\phi({}^g x) = {}^g\phi(x)$ for $g \in G, x \in X$. For any subgroup $H \subseteq G$ the set of left cosets $G/H$ can be viewed as a discrete $G$-CW space. For a family $\mathcal{F}$ of subgroups of $G$ let $\mathcal{O}_{\mathcal{F}}$ denote the category with one object $G/H$ for each subgroup $H \in \mathcal{F}$ and with maps $\phi\colon G/H \to G/H'$ the morphisms of $G$-CW spaces. A map $\phi$ is determined by its image $\phi(H) = gH'$ and we have $g^{-1}Hg \subseteq H'$. The map $\phi$ is completely determined by $g$, and any $g \in G$ determines such a map. A functor $\mathcal{M}\colon \mathcal{O}_{\mathcal{F}} \to \mathcal{AB}$ to the category of abelian groups is called a *left Bredon module*. Given any $\mathcal{F}$-space $X$ and left Bredon module $\mathcal{M}\colon \mathcal{O}_{\mathcal{F}} \to \mathcal{AB}$ one can define the Bredon homology groups $H_n^{\mathcal{F}}(X,\mathcal{M})$ for $n \geqslant 0$. The definition is recalled in Section 3 along with that of Bredon cohomology.

Our aim is to present a data type for representing Bredon modules in a way that facilitates the computation of Bredon homology and cohomology of a range of $G$-CW spaces. The data type is implemented in the HAP package for GAP and the following GAP commands illustrate how it can be used in the computation of $H_1^{\mathcal{F}}(K, \mathcal{B}) = 0$ for $K$ the Quillen simplicial complex at the prime $p = 3$ for the symmetric group of degree 9 and $\mathcal{B}$ the Burnside module; the definitions of this $K$ and $\mathcal{B}$ are recalled in Section 3. The cellular chain complex $R_* = C_*(K)$ is used in the computation.

```
gap> G:=SymmetricGroup(9);;
gap> K:=QuillenComplex(G,3);
Simplicial complex of dimension 2.
gap> R:=GChainComplex(K,G);
G-chain complex in characteristic 0 for Sym( [ 1 .. 9 ] ) .
gap> C:=TensorWithBurnsideRing(R);
Chain complex of length 2 in characteristic 0 .
gap> Homology(C,1);
[  ]
```

This computation is essentially just a routine transcription of classical mathematical definitions into computational data types, but may be of interest since the computation of Bredon cohomology of finite simplicial complexes $K$ constructed from families of subgroups of $G$ has recently been shown in [**23**] to be relevant to the gluing problem in certain fusion systems.

In Section 5 we review a selection of practical algorithms for constructing models of two particular $\mathcal{F}$-spaces $EG$ and $\underline{E}G$. For the definition of $EG$ and $\underline{E}G$ we say that an $\mathcal{F}$-space $X$ is a *model for* $E_{\mathcal{F}}$ if the $G$-CW space $X^H = \{x \in X \mid {}^h x = x \text{ for all } h \in H\}$ is contractible for each $H \in \mathcal{F}$ and empty for each subgroup $H \notin \mathcal{F}$. In the special case when the family $\mathcal{F} = \{1\}$ consists of just the identity subgroup we say that such a model is a *total space* for $G$ and denote it by $EG$. Thus a total space $EG$ is any contractible CW space on which $G$ acts freely by permuting cells; the quotient space $BG = EG/G$ obtained from an $EG$ by identifying $x$ with ${}^g x$ for $x \in EG, g \in G$ is said to be a *classifying space* for $G$. In the special case when the family $\mathcal{F} = \mathfrak{F}in$ consists of all finite subgroups of $G$ we say that a model for $E_{\mathcal{F}}$ is a *proper total space* for $G$ and denote it by $\underline{E}G$. Some authors refer to $\underline{E}G$ as a *classifying space for proper actions* [**24**]. If $G$ is torsion free then a proper total space $\underline{E}G$ is the same thing as a total space $EG$.

If $\mathcal{F} = \{1\}$ then a left Bredon module $\mathcal{M}$ is just a $\mathbb{Z}G$-module and the classical homology (also known as the Eilenberg-MacLane homology) of the group $G$ with coefficients in $\mathcal{M}$ can be defined as $H_n(G, \mathcal{M}) = H_n^{\mathcal{F}}(EG, \mathcal{M})$. For $\mathcal{F} = \mathfrak{F}in$ there is a well-defined analogue $\underline{H}_n(G, \mathcal{M}) = H_n^{\mathcal{F}}(\underline{E}G, \mathcal{M})$ which we refer to as the *Bredon homology* of $G$. In order to implement these two homology definitions on a computer we need a practical method for constructing finitely many dimensions of the spaces $EG$ and $\underline{E}G$ on a computer. One approach is to take some theoretical descriptions of the spaces and simply transcribe them into a suitable data type. The following GAP commands illustrate this approach to computing the Eilenberg-MacLane homology with integer coefficients $H_2(SL_3(\mathbb{Z}), \mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ and the Bredon homology

$\underline{H}_0(SL_3(\mathbb{Z}), \mathcal{R}) = \mathbb{Z}^8$ with coefficients in the complex representation ring $\mathcal{R}$. The definition of the Bredon module $\mathcal{R}$ is recalled in Section 3.

```
gap> R:=ContractibleGcomplex("SL(3,Z)s");
Non-free resolution in characteristic 0 for <matrix group> .
gap> F:=FreeGResolution(R,3);
Resolution of length 3 in characteristic 0 for <matrix group> .
gap> C:=TensorWithIntegers(F);
Chain complex of length 3 in characteristic 0 .
gap> Homology(C,2);
[ 2, 2 ]

gap> D:=TensorWithComplexRepresentationRing(R);
Chain complex of length 3 in characteristic 0 .
gap> Homology(D,0);
[ 0, 0, 0, 0, 0, 0, 0, 0 ]
```

The computation involves the cellular chain complex $R_* = C_*(X)$ of the model $X$ for $\underline{E}SL_3(\mathbb{Z})$ constructed by C. Soulé in [**34**] and used in [**32**]. The algebraic structure of $C_*(X)$ has been transcribed into a suitable data type and stored in GAP. For the Eilenberg-MacLane homology the computation uses an algorithmic procedure to construct a free $\mathbb{Z}SL_3(\mathbb{Z})$-resolution $F_*$ from $C_*(X)$. A spectral sequence argument [**32**] shows that in examples such as this, with Bredon homology $\underline{H}_n(G, \mathcal{R})$ trivial for $n \geqslant 2$, there is an isomorphism $K_n^G(\underline{E}G) \cong \underline{H}_n(G, \mathcal{R})$, $n = 0, 1$, where $K_n^G(\underline{E}G)$ denotes equivariant $K$-theory. Interest in Bredon homology in this context stems from the Baum-Connes conjecture [**5**] asserting that $K_n^G(\underline{E}G)$ should be isomorphic to the $K$-theory of the reduced $C^*$-algebra of $G$.

A second approach to computing with $EG$ and $\underline{E}G$ is to design and implement algorithms that input generators (and possibly other group-theoretic information) for $G$ and return the low-dimensional algebraic structure of $C_*(EG)$ and $C_*(\underline{E}G)$. The following GAP commands illustrate this approach to computing $H_2(G, \mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$ and $\underline{H}_2(G, \mathcal{B}) = \mathbb{Z}^4 \oplus \mathbb{Z}_2$ for $G$ the Euclidean crystallographic group arising as the 32nd group of dimension 3 in the Cryst [**14**] library of crystallographic groups available as part of the GAP system.

```
gap> G:=SpaceGroupIT(3,32);;
gap> gens:=GeneratorsOfGroup(G);;
gap> B:=CrystGFullBasis(G);;
gap> R:=CrystGcomplex(gens,B,0);;
gap> F:=FreeGResolution(R,3);;
gap> C:=TensorWithIntegers(F);;
gap> Homology(C,2);
[ 2, 2 ]
gap> D:=TensorWithBurnsideRing(R);;
gap> Homology(D,1);
[ 2, 0, 0, 0 ]
```

The algorithm for constructing $\underline{E}G$ in this example is new and is detailed in Section 7.2.

## 3. Review of Bredon homology

Fix a group $G$, a family $\mathcal{F}$ of subgroups of $G$, an $\mathcal{F}$-space $X$, and a left Bredon module $\mathcal{M}\colon \mathcal{O}_{\mathcal{F}} \to \mathcal{AB}$. Let $\{e^n_\alpha\}$ denote a minimal set of representatives of the $G$-orbits of $n$-cells in $X$. Write $S^n_\alpha = Stab_G(e^n_\alpha) \in \mathcal{F}$. The terms in the cellular chain complex $C_*(X)$ have the form

$$C_n(X) \;\cong\; \bigoplus_{\{e^n_\alpha\}} \mathbb{Z}G \otimes_{S^n_\alpha} \mathbb{Z} \;\cong\; \bigoplus_{\{e^n_\alpha\}} \mathbb{Z}[G/S^n_\alpha]\ .$$

We set

$$M_n = \bigoplus_{\{e^n_\alpha\}} \mathcal{M}(G/S^n_\alpha)$$

and note that the boundary homomorphism $\partial_n\colon C_n(X) \to C_{n-1}(X)$ induces a homomorphism of abelian groups $d_n\colon M_n \to M_{n-1}$ for which $M_* = (M_n, d_n)_{n\geqslant 0}$ is a chain complex. If $\partial_n(e^n_\alpha) = \sum_{\{e^{n-1}_\beta\}} m_\beta g_\beta e^{n-1}_\beta$, with $m_\beta \in \mathbb{Z}, g_\beta \in G$, then the inclusions $g_\beta^{-1} S^n_\alpha g_\beta \hookrightarrow S^{n-1}_\beta$ induce homomorphisms $\iota^n_{\alpha,\beta}\colon \mathcal{M}(G/S^n_\alpha) \to \mathcal{M}(G/S^{n-1}_\beta)$. The homomorphism $d_n\colon M_n \to M_{n-1}$ is $d_n = \sum_{\{e^{n-1}_\beta\}} m_\beta \iota^n_{\alpha,\beta}$.

**Definition 3.1.** The *Bredon homology* of $X$ with coefficients in the Bredon module $\mathcal{M}$ is defined as

$$H^{\mathcal{F}}_n(X, \mathcal{M}) = H_n(M_*).$$

The family $\mathcal{F}$ doesn't play much of a role in this definition but it is useful to retain $\mathcal{F}$ in the notation.

The following example of an $\mathcal{F}$-space is among those implemented in [**16**].

**Example 3.1.** The *order complex* of a partially ordered set $\mathcal{A}$ is the simplicial complex $\Delta(\mathcal{A})$ with vertex set $\mathcal{A}$ and with $k$-simplices the chains $A_0 < A_1 < \cdots < A_k$ of $k+1$ distinct elements $A_i \in \mathcal{A}$. When $\mathcal{A}$ is a collection of subgroups of some group $G$ which is closed under conjugation, $gA_ig^{-1} \in \mathcal{A}$ for all $A_i \in \mathcal{A}$ and $g \in G$, then conjugation induces an action of $G$ on $\Delta(\mathcal{A})$. The simplicial complex $\Delta(\mathcal{A})$ is a rigid $G$-CW space with respect to this action. When $\mathcal{A} = \mathcal{A}_p$ is the poset of all non-trivial elementary abelian $p$-subgroups of $G$ the $G$-CW space $\Delta(\mathcal{A}_p)$ is called the *Quillen complex* of $G$ at the prime $p$. This is an example of an $\mathcal{F}$-space for $\mathcal{F}$ the family of all subgroups of $G$.

The following three examples of left Bredon modules are implemented in [**16**].

**Example 3.2.** For any family $\mathcal{F}$ of subgroups of $G$, and any abelian group $A$, we have the *constant Bredon module* $A\colon \mathcal{O}_{\mathcal{F}} \to \mathcal{AB}$ which sends each object $G/H$ to the abelian group $A$ and each morphism $\phi\colon G/H \to G/H'$ to the identity homomorphism $1\colon A \to A$. Of particular interest is the abelian group $A = \mathbb{Z}$ and the corresponding homology theories $H_n(G, \mathbb{Z})$ and $\underline{H}_n(G, \mathbb{Z})$.

**Example 3.3.** For a group $G$ and $\mathcal{F} = \mathfrak{F}in$ we define the *complex representation ring* to be the left Bredon module $\mathcal{R}\colon \mathcal{O}_\mathcal{F} \to \mathcal{AB}$ that sends an object $G/H$ to the vector space $R_\mathbb{C}(H)$ of complex representations of the finite group $H$. For a $G$-map $\phi\colon G/H \to G/H'$ we have $g^{-1}Hg \subset H'$ for some $g \in G$ and define $\mathcal{R}(\phi)\colon R_\mathbb{C}(H) = R_\mathbb{C}(g^{-1}Hg) \to R_\mathbb{C}(H')$ to be induction.

**Example 3.4.** For a group $G$ and $\mathcal{F} = \mathfrak{F}in$ we define the *Burnside ring* to be the left Bredon module $\mathcal{B}\colon \mathcal{O}_\mathcal{F} \to \mathcal{AB}$ that sends an object $G/H$ to the free abelian group $BS(H)$ with isomorphism types of transitive $H$-sets as basis. For a $G$-map $\phi\colon G/H \to G/H'$ with $g^{-1}Hg \subset H'$, $g \in G$, we again define $\mathcal{B}(\phi)\colon BS(H) = BS(g^{-1}Hg) \to BS(H')$ to be induction.

Another readily implemented example for $\mathcal{F} = \mathfrak{F}in$ is the left Bredon module that sends $G/H$ to the integral homology group $H_n(H, \mathbb{Z})$.

**Definition 3.2.** For a family $\mathcal{F}$ of subgroups of $G$ and for a left Bredon module $\mathcal{M}$ we define the *$\mathcal{F}$-Bredon homology* of $G$ to be

$$H_n^\mathcal{F}(G, \mathcal{M}) = H_n^\mathcal{F}(X, \mathcal{M})$$

where $X$ is any model for $E_\mathcal{F}G$. It is readily shown [**5**] that this definition does not depend on the choice of model $X$ and so we can write $H_n^\mathcal{F}(E_\mathcal{F}G, \mathcal{M})$ instead of $H_n^\mathcal{F}(X, \mathcal{M})$.

For $\mathcal{F} = \{1\}$ we write $H_n(G, \mathcal{M}) = H_n(E_\mathcal{F}G, \mathcal{M})$ and refer to this as the *classical homology*, or *Eilenberg-MacLane homology*, of $G$. For $\mathcal{F} = \mathfrak{F}in$ we write $\underline{H}_n(G, \mathcal{M}) = H_n(E_\mathcal{F}G, \mathcal{M})$ and refer to this as the *Bredon homology* of $G$.

A *right Bredon module* is a contravariant functor $\mathcal{M}\colon \mathcal{O}_\mathcal{F} \to \mathcal{A}B$. Such modules give rise to the definition of Bredon cohomology in the obvious way.

## 4. A data type for Bredon modules

We now specify data types for: (i) chain complexes and maps, (ii) cellular chain complexes and maps of $\mathcal{F}$-spaces, and (iii) Bredon modules. These are used in [**16**] to implement the examples of the preceding section.

The GAP language supports a form of object oriented programming which we use in our specification. We say that an object X is a *component object* if it is a collection of components X.component_1, X.component_2, ... .

**Data Type 4.1.** Let $C_* : \cdots \to C_j \to C_{j-1} \to \cdots \to C_0 \to 0$ be a chain complex of free modules over a ring $\mathbb{K}$. Suppose that all modules in negative degrees are zero, and suppose that ordered bases for modules in non-negative degrees have been specified. We represent this data as a component object C with the following components:

- C!.dimension(k) is a function which enters an integer $k \geqslant 0$ and returns the rank of the free module $C_k$. (This rank could be infinite.)
- C!.boundary(k,j) is a function which enters integers $k \geqslant 0$, $1 \leqslant j \leqslant \text{rank}_\mathbb{K}(C_k)$ and returns the image in $C_{k-1}$ of the $j$th free generator of $C_k$. Elements in

$C_{k-1}$ are represented as vectors (*i.e.* lists) over $\mathbb{K}$ of length equal to the rank of $C_{k-1}$.

- C!.properties is a list of properties of the complex, each property stored as a pair such as ["*characteristic*", 0].

The associated function IsHAPChainComplex(C) returns the boolean true when C is of this data type. A morphism $f_*\colon C_* \to C'_*$ of such chain complexes over $\mathbb{K}$ is represented as a component object f with the following components:

- f.source is a representation of the chain complex $C_*$.
- f.target is a representation of the chain complex $C'_*$.
- f.fun(k,j) is a function which inputs integers $k$, $1 \leqslant j \leqslant \mathrm{rank}_{\mathbb{K}}(C_k)$ and returns the image in $C'_k$ of the $j$th free generator of $C_k$ under the homomorphism $f_k$.

The boundary homomorphisms in a chain complex are represented by functions, rather than by matrix arrays, to allow for *lazy evaluation*: an implementation can opt to compute boundaries of generators only at the point when they are required. To illustrate why one might wish to compute in this way, consider some classifying space $BG = EG/G$ of a group $G$ which has been constructed with only finitely many cells in each degree. Then standard Smith Normal Form algorithms can be used to compute the homology group $H_n(G,\mathbb{Z}) = H_n(C_*(BG))$. Given a homomorphism $\phi\colon G \to G'$ of such groups we may wish to compute the induced homomorphism $H_n(\phi)\colon H_n(C_*(BG)) \to H_n(C_*(BG'))$. For this we would first need to compute an induced chain map $\phi_*\colon C_*(BG) \to C_*(BG')$. One approach is to consider the bar chain complex $C_*(B^{\mathrm{bar}}G)$ where $B^{\mathrm{bar}}G = \mathrm{Nerve}(G)$ is the simplicial nerve of the category $G$. There is a readily implemented functorial chain map $\phi^{\mathrm{bar}}_*\colon C_*(B^{\mathrm{bar}}G) \to C_*(B^{\mathrm{bar}}G')$. One could try to compute $\phi_*$ as a composite

$$C_*(BG) \xrightarrow{\alpha} C_*(B^{\mathrm{bar}}G) \xrightarrow{\phi^{\mathrm{bar}}_*} C_*(B^{\mathrm{bar}}G') \xrightarrow{\beta} C_*(BG')$$

in which $\alpha$ and $\beta$ are chain equivalences. The chain complexes $C_*(B^{\mathrm{bar}}G)$ and $C_*(B^{\mathrm{bar}}G')$ will have an extremely large number of free generators in each degree, and for infinite groups will have infinitely many free generators in each degree. For this reason these chain complexes would need to be implemented using lazy evaluation. This approach to induced homology homomorphisms is one of the methods implemented in [**16**] and relies on the availability of explicitly defined contracting homotopies on the universal covers $\widetilde{B(G)} = EG$ and $\widetilde{\mathrm{Nerve}(G)}$ as explained in Section 6.

**Data Type 4.2.** Let $R_* = C_*(X)$ be the cellular chain complex of a $G$-CW space $X$. The space $X$ need not be rigid. Each chain group $R_k = C_k(X)$ is free abelian with $\mathbb{Z}$-basis corresponding to the $k$-cells of $X$. We suppose that some ordering of the $k$-cells has been chosen and fixed, and that for each $k$-cell some orientation has also been chosen and fixed. Each $R_k$ is also a $\mathbb{Z}G$-module. The action of $G$ permutes the free abelian generators, and the boundary homomorphisms are $G$-equivariant. We suppose that some minimal set $\{e^k_j\}$ of representatives of $G$-orbits of the $k$-cells has been chosen. We represent $R_*$ as a component object R with the following components.

- R!.dimension(k) is a function which inputs an integer $k \geqslant 0$ and returns the cardinality $r_k$ of the set $\{e_j^k\}$.
- R!.elts is a list of some elements of $G$. Not all elements of $G$ need to be in this list, and a given element of $G$ may appear several times in the list. Furthermore, elements of $G$ can be appended to the list during the course of a computation. If $G$ is a small finite group then typically the list would be a full listing of all the elements of $G$.
  (This approach to referring to elements allows for: (i) certain computations with groups, such as finitely presented groups, where no algorithm for solving the word problem is available; (ii) easy implementation of topological constructions that combine $G$-equivariant chain complexes involving groups $G$ represented in different ways, such as matrix groups, finitely presented groups, permutation groups, ... .)
- R!.boundary(k,j) is a function which inputs integers $k \geqslant 0$, $1 \leqslant j \leqslant r_k$ and returns the image in $R_{k-1}$ of the $j$th $\mathbb{Z}G$-generator $e_j^k$ of the $\mathbb{Z}G$-module $R_k$. An element $w \in R_{k-1}$ can be expressed, in a non-unique way, as a sum
$$w = \epsilon_1 g_1 e_{i_1}^{k-1} + \epsilon_2 g_2 e_{i_2}^{k-1} + \cdots + \epsilon_n g_n e_{i_n}^{k-1} \in R_{k-1}, \qquad (g_i \in G, \epsilon_i = \pm 1) \ .$$
  We represent this sum as a list of integer pairs
$$w = [[\epsilon_1 i_1, i'_1], [\epsilon_2 i_2, i'_2], \ldots, [\epsilon_n i_n, i'_n]]$$
  with R!.elts$[i'_m] = g_m$ for $m = 1, 2, \ldots, n$.
- R!.stabilizer(k,j) is a function which inputs integers $k \geqslant 0$, $1 \leqslant j \leqslant r_k$ and returns the subgroup of G consisting of those elements that fix, up to sign, the $j$th $\mathbb{Z}G$-generator $e_j^k \in R_k$.
- R!.action(k,j,i) is a function which inputs integers $k \geqslant 0$, $1 \leqslant j \leqslant r_k$, $1 \leqslant i \leqslant \ell$ where $\ell$ is the current length of the list R!.elts and returns $\pm 1$ according to how the group element R!.elts[i] acts on the orientation of the $j$th $\mathbb{Z}G$-generator $e_j^k \in R_k$.
- R!.group is the GAP respresentation of the group $G$. This representation could be, for instance, as a finitely presented group, or as a polycyclically presented group, or as a permutation group, or as a matrix group and so forth.
- R!.homotopy is, in many cases, the boolean variable fail. In cases where the space $X$ is contractible, and where an explicit contracting chain homotopy $h_k \colon C_k(X) \to C_{k+1}(X)$ $(k \geqslant 0)$ is available, R!.homotopy(k,[j,i]) is a function which inputs integers $k \geqslant 0$, $1 \leqslant j \leqslant r_k$, $1 \leqslant i \leqslant \ell$. This function returns the image $h(ge_j^k) \in R_{k+1}$ for $g =$R!.elts[i]$\in G$. The elements of the form $ge_j^k$ freely generate $R_k$ as an abelian group and so the value of $h_k(w)$ can be evaluated for arbitrary elements $w \in R_k$.
  (The role of contracting homotopies in computational homological algebra is explained in Section 6.)
- R!.properties is a list of properties of the complex, each property stored as a pair such as ["*characteristic*", 0].

The associated function IsHAPGChainComplex(R) returns the boolean true when R is of this data type.

Let $f\colon X \to X'$ be a cellular map from a $G$-CW space $X$ to a $G'$-CW space $Y$, and let $f_{\#}\colon G \to G'$ be a group homomorphism such that $f(gx) = (f_{\#}g)f(x)$. The induced $f_{\#}$-equivariant chain map $f_*\colon R_* = C_*X \to R'_* = C_*X'$ is represent as a component object f with the following components:

- f.source is a representation of the $G$-chain complex $R_*$.
- f.target is a representation of the $G'$-chain complex $R'_*$.
- f.fun(k,w) is a function which inputs an integer $k \geqslant 0$ and list w representing a word $w \in R_k$, and returns a list representing the image $f_k(w) \in R'_k$.

Data Type 4.2 has evolved over several years of development of the HAP package. A realization of the importance of storing homotopies dates back to conversations with Ronnie Brown and Larry Lambe some twenty years ago regarding the Homological Perturbation Lemma [**7**]. The choice of method for storing non-free group actions is based on discussions with Mathieu Dutour-Sikirić and was first used in [**12**].

It is useful to record certain mathematical properties that may be known to hold for a given instance R of Data Type 4.2. In particular, the function IsHAPNonFreeResolution(R) returns the boolean true when R is known to correspond to an acyclic $\mathbb{Z}G$-chain complex $R_*$ with $H_0(R_*) \cong \mathbb{Z}$; we refer to such an $R_*$ as a *non-free resolution* or *non-free $\mathbb{Z}G$-resolution*. The function IsHAPResolution(R) returns the boolean true when R is known to correspond to an acyclic and $\mathbb{Z}G$-free chain complex $R_*$ with $H_0(R_*) \cong \mathbb{Z}$; we refer to such an $R_*$ as a *free resolution* or *free $\mathbb{Z}G$-resolution*. In this latter case the components R!.stabilizer(k,j) and R!.action(k,j,g) may be omitted. Note that our terminology allows for a non-free resolution to be a free resolution.

It is possible to algorithmically test whether certain properties hold for a given instance R of data type 4.2. In particular, let us say that the $\mathbb{Z}G$-chain complex $R_* = C_*(X)$ described in 4.2 is *rigid* if the following property holds for each $\mathbb{Z}G$-generator $e^k_\alpha \in R_k$, $k \geqslant 1$.

**Chain rigidity property:** Suppose $\partial_k(e^k_\alpha) = \sum_{\beta\in I} m_\beta g_\beta e^{k-1}_\beta$, where the indices $\beta$ are distinct and the integers $m_\beta$ are non-zero for $\beta \in I$. Then $Stab_G(e^k_\alpha) \leqslant Stab_G(g_\beta e^{k-1}_\beta)$ for $\beta \in I$ and, at the chain level, $ge^k_\alpha \neq -e^k_\alpha$ for $g \in Stab_G(e^k_\alpha)$.

The function IsRigid(R) returns the boolean true if the chain rigidity property holds for all $e^k_\alpha$, $k \geqslant 1$, and false otherwise.

The following is the main data type of the paper.

**Data Type 4.3.** A left Bredon module $\mathcal{M}\colon \mathcal{O}_{\mathcal{F}} \to \mathcal{AB}$ is represented as a function M that inputs a rigid $G$-chain complex R with stabilizer groups in the family $\mathcal{F}$ (represented using Data Type 4.2) and returns a chain complex M(R) over $\mathbb{Z}$ (represented using Data Type 4.1).

By an *implementation* of Data Type 4.3 we actually mean an implementation M of 4.3 together with an implementation R of Data Type 4.2, since the former is redundant without the latter. Given implementations of M and R it is straightforward to compute the homology of the chain complex M(R) using the Smith Normal Form algorithm.

The HAP package currently contains implementations of the the following Bredon functors M: tensor with the integers $-\otimes_{\mathbb{Z}G}\mathbb{Z}$; tensor with the integers modulo a prime $-\otimes_{\mathbb{Z}G}\mathbb{Z}/pZ$; tensor with a finitely generated $\mathbb{Z}G$-module $-\otimes_{\mathbb{Z}G}A$; the complex representation ring $\mathcal{R}$; and the Burnside ring $\mathcal{B}$.

The main implementations of rigid $G$-chain complexes R in HAP are of complexes that are (not necessarily free) $\mathbb{Z}G$-resolutions of the ring $\mathbb{Z}$.

## 5. Review of algorithms for resolutions

The HAP package includes implementations of a range of algorithms which input an integer $n \geqslant 0$ and information about a group $G$ in some given class and, at least in principle, use the information to output in finite time the first $n+1$ terms of a *free* $\mathbb{Z}G$-resolution $R_*$ of $\mathbb{Z}$ with $R_k$ a finitely generated free $\mathbb{Z}G$-module in each degree $0 \leqslant k \leqslant n$. For the purposes of this section, let us say that a class of groups is *HAP-resolvable* if HAP contains an implementation of such an algorithm that applies to all groups in the class.

The following classes of groups are HAP-resolvable (where references give details of the implemented resolutions/algorithms): finite groups [**15**]; finitely generated nilpotent groups and, more generally, any group $G$ admitting a subnormal chain of subgroups $N_1 \leqslant N_2 \leqslant \cdots \leqslant N_k = G$ where each quotient $N_{i+1}/N_i$ is HAP-resolvable [**18**]); Bieberbach groups [**29**]; crystallographic groups [**29, 2, 1**]; Coxeter groups [**10, 18**]; Artin groups for which the $K(\pi,1)$ conjecture is known to hold [**30, 35, 19**]; a few finite simple groups such as the Mathieu groups [**11**]; $PSL_4(\mathbb{Z})$ and a few related arithmetic groups [**12**]; $SL_2(\mathcal{O})$ and $GL_2(\mathcal{O})$ for a few rings $\mathcal{O}$ of integers in an imaginary quadratic field [**27, 33**]; $SL_2(\mathbb{Z}[1/m])$ for any integer $m \geqslant 1$ [**3**]; finite $p$-groups (yielding a minimal resolution over the field of $p$ elements) [**8**]; graphs of HAP-resolvable groups; finite index subgroups of HAP-resolvable groups; direct products of HAP-resolvable groups.

Let us say that a class of groups is *HAP-properly resolvable* if HAP contains an implementation of an algorithm that inputs any integer $n \geqslant 0$ and group $G$ in the class and, at least in principle, returns in finite time the first $n+1$ terms of a $\mathbb{Z}G$-resolution of the form $R_* = C_*(\underline{E}G)$ with $R_k$ finitely generated as a $\mathbb{Z}G$-module for each $0 \leqslant k \leqslant n$.

For a finite group $G$ one can take $\underline{E}G$ to be a single point, and for torsion free infinite groups $G$ one can take $\underline{E}G = EG$ and $R_*$ to be any free $\mathbb{Z}G$-resolution of $\mathbb{Z}$. Thus explicit constructions of $\underline{E}G$ are of particular interest for infinite groups $G$ with torsion. For any discrete group $G$ one can take $\underline{E}G$ to be the simplicial complex with $k$-simplices all $k+1$-element subsets of $G$ [**24**], though $R_k = C_k(\underline{E}G)$ will then not be finitely generated if $G$ is infinite.

As a first approximation to $\underline{E}G$ we denote by $\underline{E}'G$ any $G$-CW space $X$ such that

$X^H = \{x \in X \mid {}^h x = x \text{ for all } h \in H\}$ is contractible for each finite subgroup $H$ in $G$. The important distiction between $\underline{E}'G$ and $\underline{E}G$ is that the latter is required to be rigid.

At present HAP contains implementations of algorithms for returning a $\mathbb{Z}G$-resolution $C_*(\underline{E}'G)$ for just two classes of groups: Coxeter groups [**31**]; crystallographic groups [**29, 2, 1**]. It also includes a resolution $C_*(\underline{E}'G)$ for a number of specific groups including: $G = SL_3(\mathbb{Z})$ (obtained by transcribing data from [**32**]); $PGL_3(\mathbb{Z}[\mathbf{i}])$, $PSL_4(\mathbb{Z})$, $Sp_4(\mathbb{Z})$ (obtained by transcribing data from [**13**]); $GL_2(\mathcal{O}_{-m})$ and $SL_2(\mathcal{O}_{-m})$ for $m = 1, 2, 3, 5, 6, 7, 10, 11, 13, 14,\ 15, 17, 19, 21, 22, 23, 26, 43$ with $\mathcal{O}_{-m}$ the ring of integers in the field $\mathbb{Q}(\sqrt{-m})$. [**33**].

By subdividing cells it is often practical to convert a resolution $C_*(\underline{E}'G)$ into a rigid resolution $C_*(\underline{E}G)$. We discuss this in Section 7.

## 6. Homotopies and discrete vector fields

Various constructions in the cohomology of groups require the following element of choice.

Given a $\mathbb{Z}G$-resolution $R_*$ and an element $x \in \ker(\partial_k \colon R_k \to R_{k-1})$ choose an element $\tilde{x} \in R_{k+1}$ such that $\partial_{k+1}(\tilde{x}) = x$. (1)

In order to translate such constructions into procedures that can be implemented on a computer one needs a method of performing (1) algorithmically. In particular, these comments apply to the construction of (co)homology homomorphisms induced by group homomorphisms, the construction of cup products and other cohomology operations, the construction of explicit cocycles, and various constructions that express a resolution for a group in terms of simpler resolutions for its subgroups and quotient groups.

It is sometimes possible to choose $\tilde{x}$ by solving a system of equations using elementary linear algebra or by using more sophisticated techniques from Gröbner basis theory. If the resolution $R_*$ is endowed with a contracting homotopy, *i.e.* a family of $\mathbb{Z}$-linear homomorphisms $h_k \colon R_k \to R_{k+1}$ satisfying

$$h_{k-1}\partial_k + \partial_{k+1}h_k = 1$$

for $k \geqslant 0$ with $h_{-1} = 0$, then one can very quickly solve (1) as follows.

Given a $\mathbb{Z}G$-resolution $R_*$ and an element $x \in \ker(\partial_k \colon R_k \to R_{k-1})$ then the equation $\partial_{k+1}(\tilde{x}) = x$ is satisfied by setting $\tilde{x} := h_k(x) \in R_{k+1}$. (2)

The following notion can be useful in designing explicit contracting homotopies on cellular chain complexes of contractible CW spaces. Recall that a CW space $X$ is said to be *regular* if the attaching map of each cell is a homeomorphism on its boundary.

**Definition 6.1.** A *discrete vector field* on a regular CW-space $X$ is a collection of formal arrows $s \to t$ where

1. $s, t$ are cells of $X$ with $\dim(t) = \dim(s) + 1$ and with $s$ lying in the boundary of $t$. We say that $s$ and $t$ are *involved* in the arrow, that $s$ is the *source* of the arrow, and that $t$ is the *target* of the arrow.

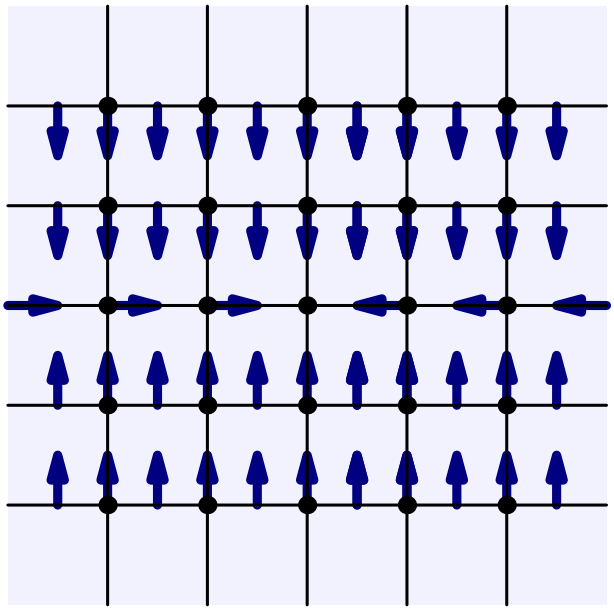

Figure 1: A contracting vector field on a tessellation of $X = \mathbb{R}^2$.

2. any cell is involved in at most one arrow.

The term *discrete vector field* is due to Forman [**20**]. In an earlier work [**22**] Jones calls this concept a *marking*.

By a *chain* in a discrete vector field we mean a sequence of arrows

$$\ldots, s_1 \to t_1, s_2 \to t_2, s_3 \to t_3, \ldots$$

where $s_{i+1}$ lies in the boundary of $t_i$ for each $i$. A chain is a *circuit* if it is of finite length with source $s_1$ of the initial arrow $s_1 \to t_1$ lying in the boundary of the target $t_n$ of the final arrow $s_n \to t_n$. A discrete vector field is said to be *admissible* if it contains no circuits and no chains that extend infinitely to the right. A cell in $X$ is said to be *critical* if it is not involved in any arrow.

We say that a discrete vector field on a regular CW space $X$ is a *contracting vector field* if it is admissible and if it has only one critical cell, the critical cell being of dimension 0. Figure 1 illustrates a contracting vector field on a CW structure on the plane $X = \mathbb{R}^2$ arising from a tessellation of the plane by unit squares.

We identify the cells of $X$ with free $\mathbb{Z}$-linear generators of the cellular chain complex $C_*(X)$, the $k$-cells corresponding to a basis of $C_k(X)$. We denote this basis by $B_k(X) \subset C_k(X)$. A discrete vector field on $X$ then induces a partial pairing between the sets $B_k(X)$ and $B_{k+1}(X)$ for $k \geqslant 0$. A pairing involving $s \in B_k(X)$ and $t \in B_{k+1}(X)$ is again denoted by an arrow $s \to t$. For an element $w \in C_k(X)$ and basis element $e \in B_k(X)$ we write $|w : e|$ for the coefficient of $e$ in the unique expression of $w$ as a linear combination of basis elements. The following proposition

is routine to verify.

**Proposition 6.1.** *A contracting vector field on a regular CW space $X$ induces a contracting homotopy on the cellular chain complex $C_*(X)$. The $\mathbb{Z}$-linear homomorphisms $h_k\colon C_k(X) \to C_{k+1}(X)$ of the contracting homotopy are given on generators $e \in B_k(X)$ by the following recursive formulae.*

$$h_k(e) = \begin{cases} 0, & \text{if } e \text{ is critical or if } e \text{ is the target of an arrow.} \\ f + h_k(\partial_{k+1} f - e) & \text{if } e \text{ is the source of an arrow } e \to f \text{ and } |\partial f : e| = 1. \\ -f - h_k(\partial_{k+1} f + e) & \text{if } e \text{ is the source of an arrow } e \to f \text{ and } |\partial f : e| = -1. \end{cases}$$

*The recursion stops due to the admissibility of the discrete vector field.*

Suppose now that the regular CW space $X$ is a $G$-CW space for some group $G$. If $X$ admits a contracting vector field then the chain complex $C_*(X)$ is a $\mathbb{Z}G$-resolution of $\mathbb{Z}$ and Proposition 6.1 provides formulae for a contracting homotopy on $C_*(X)$.

As an example, consider the finitely presented group $G = \langle a, b \mid aba = b \rangle$. An action of $G$ on the plane $X = \mathbb{R}^2$ is defined by $a\colon \mathbb{R}^2 \to \mathbb{R}^2, (x, y) \mapsto (x + 1, -y)$ and $b\colon \mathbb{R}^2 \to \mathbb{R}^2, (x, y) \mapsto (x, y + 1)$. It is readily checked that this is a free action admitting the unit square $[0, 1] \times [0, 1]$ as a fundamental domain. The quotient $X/G$ is the Klein bottle. The regular CW structure and contracting vector field illustrated in Figure 1 provide a free $\mathbb{Z}G$-resolution $C_*(X)$ with explicit contracting homotopy. This resolution is implemented in HAP and can be used to recover the calculations $H^1(G, \mathbb{Z}) = \mathbb{Z}$, $H^2(G, \mathbb{Z}) = \mathbb{Z}_2$, $H^n(G, \mathbb{Z}) = 0$ for $n \geqslant 3$. Moreover, the contracting homotopy on $C_*(X)$ and the standard contracting homotopy on the bar resolution $C_*(\widetilde{\text{Nerve}(G)})$ are implemented and can be used to construct chain equivalences

$$\phi_*\colon C_*(X) \otimes_{\mathbb{Z}G} \mathbb{Z} \rightleftarrows C_*(\text{Nerve}(G))\colon \psi_*$$

between a finitely generated chain complex and a chain complex that is infinitely generated in each degree. The larger chain complex is of interest because it is functorial and thus good for theoretical descriptions of certain constructions; the smaller chain complex is of interest because its homology is quickly calculated using the Smith Normal Form algorithm. Section 7.2 contains an explanation of how this example extends to certain other crystallographic groups.

## 7. Rigidification of $G$-CW spaces

The cellular structure of a regular CW space is captured, up to homotopy of attaching maps, by the cellular chain complex $C_*X$.

For any regular CW space $X$ one defines $sd(X)$ to be the simplicial complex with one vertex for each cell in $X$, and with one $k$-simplex for each sequence $e^0, e^1, \cdots, e^k$ of cells of $X$ with $e^i$ of dimension $i$ for $0 \leqslant i \leqslant k$ and with $e^{i-1}$ in the boundary of $e^i$ for $1 \leqslant i \leqslant k$. The geometric realization $|sd(X)|$ is called the *barycentric subdivision* of $X$.

Any regular CW space $X$ is homeomorphic to the geometric realization $|sd(X)|$ of its barycentric subdivision. If $X$ is a $G$-CW space then the action of $G$ on $X$ induces an action of $G$ on $|sd(X)|$, and the latter action is clearly rigid. We thus have the following computationally useful result.

**Lemma 7.1.** *Let $X$ be a $G$-CW space with regular cell structure. Then the homeomorphic space $|sd(X)|$ is a rigid $G$-CW space.*

The following GAP commands apply barycentric subdivision to a non-rigid contractible $G$-CW space $X$ for the group $G = SL_2(\mathcal{O}_{-3})$. The 2-dimensional space $X$ is obtained from [**27, 33**] and the subdivided space $sd(X)$ is used to compute the Bredon homology

$$\underline{H}_0(SL_2(\mathcal{O}_{-3}), \mathcal{R}) = \mathbb{Z}_2 \oplus \mathbb{Z}^9,$$

$$\underline{H}_1(SL_2(\mathcal{O}_{-3}), \mathcal{R}) = \mathbb{Z}$$

with coefficients in the complex representation ring $\mathcal{R}$. The analogous Bredon homology for $PSL_2(\mathcal{O}_{-3})$ has been calculated in [**26**] using the same space $X$.

```
gap> R:=ContractibleGcomplex("SL(2,O-3)");;
gap> IsRigid(R);
false
gap> S:=BaryCentricSubdivision(R);;
gap> C:=TensorWithComplexRepresentationRing(S);;
gap> Homology(C,0);
[ 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(C,1);
[ 0 ]
```

### 7.1. Coxeter groups

A *Coxeter matrix* is a symmetric $n \times n$ matrix whose entries $m(i,j)$ are either positive integers or the symbol $\infty$, with $m(i,j) = 1$ if and only if $i = j$. Such a matrix is represented by an $n$-vertex labelled graph $D$ (called a *Coxeter graph*) with edge joining vertices $i$ and $j$ if and only if $m(i,j) \geqslant 3$; the edge is labelled by $m(i,j)$. The *Coxeter group* $W_D$ is defined to be the group generated by the set of symbols $S = \{x_1, \cdots, x_n\}$ subject to relations $x_i^2 = 1$ and $(x_i x_j)_{m(i,j)} = (x_j x_i)_{m(i,j)}$ for all $i \neq j$, where $(xy)_m$ denotes the word $xyxyx\ldots$ of length m. There is a faithful matrix representation $\rho\colon W_D \to GL_n(\mathbb{R})$ obtained by constructing a vector space $V$ over $\mathbb{R}$ with basis $\{\alpha_i\}_{x_i \in S}$ and inner product

$$\langle \alpha_i, \alpha_j \rangle = -\cos(\pi/m_{ij})$$

where we interpret $\cos(\pi/\infty) = 1$. The representation $\rho$ is defined on generators $x_i \in S$ by letting $\rho(x_i)$ be the matrix of the linear map

$$V \to V, v \mapsto v - 2\langle \alpha_i, v \rangle \alpha_i.$$

Assume for the moment that the Coxeter group $W_D$ is finite. The representation $\rho$ realizes $W_D$ as a group of orthogonal transformations of $\mathbb{R}^n$ with generators $\rho(x_i)$

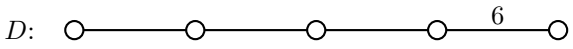


Figure 2:

equal to reflections [**9**]. For convenience we identify $W_D = \rho(W_D)$ and $x_i = \rho(x_i)$. Let $\mathcal{A}$ be the set of hyperplanes corresponding to all the reflections in $W_D$. For any point $v$ in $\mathbb{R}^n \setminus \mathcal{A}$ we denote by $P_D$ the convex hull of the orbit of $v$ under the action of $W_D$. The face lattice of the $n$-dimensional convex polytope $P_D$ depends only on the graph $D$. (To see this, first note that the vertices of $P_D$ are the points $g \cdot v$ for $g \in W_D$ and that there is an edge between $g \cdot v$ and $g' \cdot v$ if and only if $g^{-1}g' \in S$. Thus the combinatorial type of the 1-skeleton of $P_D$ does not depend on the choice of point $v$. Furthermore, each vertex of the $n$-dimensional polytope $P_D$ is incident with precisely $n$ edges; hence $P_D$ is simple and the face lattice of the polytope is determined by the combinatorial type of the 1-skeleton [**6**].) The polytope $P_D$ is a $G$-CW space for $G = W_D$. The contractible simplicial space $|sd(P_D)|$ is called the *Davis complex* for the finite group $G = W_D$.

To define the Davis complex for an arbitrary Coxeter group $W_D$ we first consider the *Cayley graph* $\Gamma_D$ consisting of one vertex for each $g \in W_D$ and one undirected edge connecting the pair of vertices $\{g, gx\}$ for each $g \in W_D$ and $x \in S$. We can consider any subset of generators $T \subset S$ that generates a finite subgroup $\langle T \rangle \leqslant W_D$. This subgroup can be shown to be a Coxeter group $W_{D^T} = \langle T \rangle$ where $D^T$ is the full subgraph of $D$ on those vertices corresponding to generators in $T$. The 1-skeleton of the convex polytope $P_{D^T}$ can be identified with the Cayley graph $\Gamma_{D^T}$. We view $\Gamma_{D^T}$ as a full subgraph of $\Gamma_D$. For each coset $gW_{D^T}$ in $W_D$ there is a corresponding subgraph $g\Gamma_{D^T} \subset \Gamma_D$. We take $X_D$ to be the union of the graph $\Gamma_D$ with the polytopes $gP_{D^T}$ where $T$ ranges over all maximal subsets $T \subset S$ that generate a finite subgroup, and $g$ ranges over a set of coset representative for $W_{D^T}$, and where any two polytope faces with identical 1-skeleta in $\Gamma_D$ are identified. The space $X_D$ is a $G$-CW space for $G = W_D$ and is known to be a model for $\underline{E}'G$ [**9**]. The contractible simplicial space $|sd(X_D)|$ is called the *Davis complex* for $G$ and is a model for $\underline{E}G$.

The following GAP commands use the data types of Section 3 to compute the non-rigid $G$-CW space $X_D$ for the infinite Coxeter group $G = W_D$ corresponding to the diagram $D$ of Figure 7.1. The figure follows convention in omitting to print the number 3 on those edges of $D$ labelled by $m(i, j) = 3$. The $\mathbb{Z}G$-chain complex $R_* = C_*(X_D)$ is converted to a free $\mathbb{Z}G$-resolution $F_*$ and used to compute $H_2(W_D, \mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$. Barycentric subdivision could be applied to $R_*$ to obtain the rigid chain complex $C_* = C_*(|sd(X_D)|)$ from which the Bredon homology group $\underline{H}_0(W_D, \mathcal{R}) = \mathbb{Z}^{21}$ with coefficients in the complex representation ring is computed. The GAP session instead uses the implementation [**28**] to compute $C_*$ directly from $D$.

```
gap> D:=[ [1,[2,3]], [2,[3,3]], [3,[4,3]], [4,[5,6]]];;
gap> R:=CoxeterComplex(D);;
gap> F:=FreeGResolution(R,3);;
gap> Homology(TensorWithIntegers(F),2);
```

```
[ 2, 2, 2 ]

gap> C:=DavisComplex(D);;
gap> D:=TensorWithComplexRepresentationRing(C);;
gap> Homology(D,0);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0 ]
```

### 7.2. Cubical crystallographic groups

In [**2, 1**] a procedure is given which inputs an $n$-dimensional crystallographic group $G$ and attempts to return a $G$-CW structure on $\mathbb{R}^n$ with just one orbit of $n$-dimensional cells for which a representative $n$-cell has the CW structure of an $n$-cube $I^n$. To be precise, here $I = [0, 1]$ is understood to have the CW structure involving two 0-cells and one 1-cell, and the $n$-cube $I^n$ has the product CW structure. We refer to such a $G$-CW structure on $\mathbb{R}^n$ as a *cubical $G$-CW space*. The procedure succeeds on 12 of the 17 two-dimensional crystallographic groups; it succeeds on 114 of the 219 three-dimensional crystallographic groups; it succeeds on 1996 of the 4783 four-dimensional crystallographic groups. When the procedure succeeds the $n$-dimensional analogue of the contracting vector field of Figure 1 can be used to implement a contracting homotopy on $C_*(\mathbb{R}^n)$. In many cases where the procedure fails one can prove that no cubical $G$-CW structure exists on $\mathbb{R}^n$.

The procedure begins by constructing a cubical $T$-CW structure on $\mathbb{R}^n$ for the translation subgroup $T \leqslant G$. This can always be done. The procedure then considers the order $|G/T|$ of the (finite) point group and tries to decompose a representative cubical $n$-cell for $T$ into $|G/T|$ or fewer smaller cubical $n$-cells such that the smaller $n$-cells form a $G$-CW structure on $\mathbb{R}^n$.

In some cases where the procedure succeeds the resulting cubical $G$-CW space is rigid and can be used as a model for $\underline{E}G$. In other cases where the procedure succeeds the resulting space is non-rigid and thus a model only for $\underline{E}'G$. A problem with using barycentric subdivision $\underline{E}G = |sd(\underline{E}'G)|$ is the prohibitively large number of cells involved for very modest dimensions $n$. A compromise is to use a cubical subdivision based on the CW space $J = [0, 1]$ involving three 0-cells $0, 1/2, 1$ and two 1-cells $(0, 1/2)$, $(1/2, 1)$. The product $J^n$ has just $5^n$ cells, compared to the $3^n$ cells in $I^n$. We denote by $bcsd(\underline{E}'G)$ the resulting *binary cubical subdivision* of $\underline{E}'G$. This subdivision will not always be rigid, but in many cases it is.

The following GAP code constructs $R_* = C_*(\underline{E}'G)$ for the second group of dimension 4 in the Cryst [**14**] library of crystallographic groups and determines that $R_*$ is not rigid. The commands then use binary cubical subdivision to construct a rigid resolution $S_* = C_*(bcsd(\underline{E}'G))$.

```
gap> G:=SpaceGroup(4,2);;
gap> gens:=GeneratorsOfGroup(G);;
gap> B:=CrystGFullBasis(G);;
gap> R:=CrystGcomplex(gens,B,1);;
gap> IsRigid(R);
false
```

```
gap> S:=CrystGcomplex(gens,B,0);;
gap> IsRigid(S);
true
```

There are many ways to subdivide an $n$-cube into equally sized sub cubes. In general we can let $J_i = [0,1]$ be the CW space involving $m_i+1$ equally spaced 0-cells and $m_i$ equally long 1-cells. Then the product $J_1 \times \cdots \times J_n$ is a CW structure on the $n$-cube involving $m_1 \times \cdots \times m_n$ equally sized sub cubes. To apply this general cubical subdivision to a cubical space $\underline{E}'G$ for an $n$-dimensional crytallographic group $G$ one can proceed as follows. First, represent $G$ so that its translation subgroup $T \leqslant G$ is generated by $n$ orthogonal unit translations $x \mapsto x + t_1, \ldots, x \mapsto x + t_n \in T$. This can be done using a change of basis for $\mathbb{R}^n$ if necessary. Then consider the crystallographic group $H$ generated by the generators of $G$ together with the translations

$$x \mapsto x + \frac{1}{m_1}t_1, \ \ldots, \ x \mapsto x + \frac{1}{m_n}t_n.$$

The group $G$ is a finite index subgroup of $H$ of index $m_1 \times \cdots \times m_n$. We can attempt to use the function `CrystGcomplex` to construct $R_* = C_*(\underline{E}'H)$. If the attempt is successful then $R_*$ can be viewed as a $\mathbb{Z}G$-resolution corresponding to a general cubical subdivision of a cubical $G$-CW structure on $\mathbb{R}^n$.

### 7.3. Simplification of barycentric subdivision

Two disadvantages to cubical subdivison are: it applies only to $G$-CW spaces whose cells are cubes; it is not guaranteed to yield a rigid space. A naive algorithm for converting an arbitrary $G$-CW space $X$ with regular CW structure into a homotopy equivalent rigid $G$-CW space is to first construct the barycentric subdivision $|sd(X)|$ and then to simplify the cell structure on $|sd(X)|$ while preserving rigidity. We describe one method for simplifying the cell structure below. The method has not yet been fully implemented.

Let $Y$ denote any rigid $G$-CW space $X$ with regular CW structure. For instance, $Y$ could be the barycentric subdivision $|sd(X)|$. Rigidity ensures that the quotient space $W = Y/G$ is a regular CW space. Let $\rho\colon Y \twoheadrightarrow W$ denote the quotient map. On any given cell $e \subset Y$ this quotient map restricts to a homeomorpism $e \to \rho(e)$. For any cell $f = \rho(e) \subset W$ the preimage $\rho^{-1}(f)$ is $e \times G/Stab_G(e)$. By writing $G_f = Stab_G(e)$ we associate to each cell $f = \rho(e) \subset W$ a finite group $G_f \leqslant G$.

We are interested in cases where $W$ has only finitely many cells. Such a regular CW space is stored in [**16**] using the following data type.

**Data Type 7.1.** A finite regular CW space is represented as a component object W with the following components:

- W!.boundaries[n+1][k] is a list of integers $[t, a_1, ..., a_t]$ recording that the $a_i$th cell of dimension $n-1$ lies in the boundary of the $k$th cell of dimension $n$.
- W!.coboundaries[n+1][k] is a list of integers $[t, a_1, ..., a_t]$ recording that the $k$th cell of dimension $n$ lies in the coboundary of the $a_i$th cell of dimension $n-1$.
- W!.nrCells(n) is a function returning the number of cells in dimension $n$.

- W!.properties is a list of properties of the complex, each property stored as a pair such as ["*dimension*", 4].

The associated boolean valued function IsHAPRegularCWComplex(W) returns true when W is of this data type.

Our proposed simplification procedure for a regular CW space $W$ is based on the observation that if $W$ contains a $k$-cell $e^k$ lying in the boundary of precisely two $(k+1)$-cells $e_1^{k+1}, e_2^{k+1}$ with identical coboundaries then these three cells can be removed and replaced by a single cell of dimension $k+1$. The topological space $W$ is unchanged; only its CW-structure changes. The resulting CW-structure will not in general be regular. However, it will be regular if the sets of cells $V_0, V_1, V_2$ lying in the boundaries of $e^k, e_1^{k+1}, e_2^{k+1}$ respectively are such that $V_1 \cap V_2 = V_0 \cup \{e^k\}$. Thus by repeatedly replacing three cells by a single cell we will arrive at a potentially smaller CW space $V$; as topological spaces $W$ and $V$ will be identical.

As an illustration of this simplification procedure consider the 2-dimensional CW space $S$ consisting of a $9 \times 9$ array of unit squares with interior of the central square removed. Thus $S$ is of the homotopy type of a circle. Now consider $W = S \times S$. The following GAP commands construct this regular CW space $W$ and then applies an implementation of the simplification procedure to obtain a small regular CW space $V$. The space $W$ involves 129600 cells and the space $V$ involves 224 cells.

```
gap> A:=0*IdentityMat(9)+1;;A[4][4]:=0;;
gap> S:=RegularCWComplex(PureCubicalComplex(A));;
gap> W:=DirectProduct(S,S);
Regular CW-complex of dimension 4
gap> Size(W);
129600
gap> V:=SimplifiedComplex(W);
Regular CW-complex of dimension 4
gap> Size(V);
224
```

Returning now to the rigid $G$-CW space $Y$ and quotient map $\rho\colon Y \twoheadrightarrow W$ we note that if $G_f$ is the trivial group for all cells $f \subset W$ then $\rho$ is a covering map. In this case it is possible to lift the simplified CW-structure from $W$ to $Y$ and hence to $C_*Y$. In order to lift the simplified structure in the general case the simplification procedure needs one easily implemented refinement: only replace the three cells $e^k$, $e_1^{k+1}, e_2^{k+1}$ by a single cell if $G_{e^k} = G_{e_1^{k+1}} = G_{e_2^{k+1}}$.

Further algorithmic approaches to rigidification of $G$-CW spaces are developed in [**4**].

## References

[**1**] Bui A.T. An algorithm for cohomology of certain crystallographic groups. *preprint, National University of Ireland, Galway*, 2015.

[**2**] Bui A.T. Discrete vector fields and the cohomology of certain arithmetic and crystallographic groups. *PhD Thesis, National University of Ireland, Galway*, 2015.

[**3**] Bui A.T. and G. Ellis. The homology of $SL_2(\mathbb{Z}[1/m])$ for small $m$. *J. Algebra*, 408:102–108, 2014.

[**4**] Bui A.T. and A.D. Rahm. An algorithm for controlled subdivision yielding stabilizers which fix their cells pointwise. *in preparation*, 2015.

[**5**] P. Baum, A. Connes, and N. Higson. Classifying space for proper actions and $K$-theory of group $C^*$-algebras. In *$C^*$-algebras: 1943–1993 (San Antonio, TX, 1993)*, volume 167 of *Contemp. Math.*, pages 240–291. Amer. Math. Soc., Providence, RI, 1994.

[**6**] R. Blind and P. Mani-Levitska. Puzzles and polytope isomorphisms. *Aequationes Math.*, 34(2-3):287–297, 1987.

[**7**] R. Brown. The twisted Eilenberg-Zilber theorem. In *Simposio di Topologia (Messina, 1964)*, pages 33–37. Edizioni Oderisi, Gubbio, 1965.

[**8**] J.F. Carlson. Calculating group cohomology: tests for completion. *J. Symbolic Comput.*, 31(1-2):229–242, 2001. Computational algebra and number theory (Milwaukee, WI, 1996).

[**9**] M.W. Davis. *The geometry and topology of Coxeter groups*, volume 32 of *London Mathematical Society Monographs Series*. Princeton University Press, Princeton, NJ, 2008.

[**10**] C. De Concini and M. Salvetti. Cohomology of Coxeter groups and Artin groups. *Math. Res. Lett.*, 7(2-3):213–232, 2000.

[**11**] M. Dutour Sikirić and G. Ellis. Wythoff polytopes and low-dimensional homology of Mathieu groups. *J. Algebra*, 322(11):4143–4150, 2009.

[**12**] M. Dutour Sikirić, G. Ellis, and A. Schürmann. On the integral homology of $\mathrm{PSL}_4(\mathbb{Z})$ and other arithmetic groups. *J. Number Theory*, 131(12):2368–2375, 2011.

[**13**] M. Dutour-Sikríc. Data for complexes on which certain arithmetic groups act with nice stabilizer subgroups. *private communication*, 2011.

[**14**] B. Eick, F. Gahler, and W. Nickel. *CrystGap – The Crystallographic Groups Package, Version 4.1.12*, 2013. (`http://www.gap-system.org/Packages/cryst.html`).

[**15**] G. Ellis. Computing group resolutions. *J. Symbolic Comput.*, 38(3):1077–1118, 2004.

[**16**] G. Ellis. *HAP – Homological Algebra Programming, Version 1.10.13*, 2013. (`http://www.gap-system.org/Packages/hap.html`).

[**17**] G. Ellis. Cohomological periodicities of crystallographic groups. *J. Algebra*, 2015.

[**18**] G. Ellis, J. Harris, and E. Sköldberg. Polytopal resolutions for finite groups. *J. Reine Angew. Math.*, 598:131–137, 2006.

[**19**] G. Ellis and E. Sköldberg. The $K(\pi, 1)$ conjecture for a class of Artin groups. *Comment. Math. Helv.*, 85(2):409–415, 2010.

[**20**] R. Forman. Morse theory for cell complexes. *Adv. Math.*, 134(1):90–145, 1998.

[**21**] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.5.6*, 2013. (`http://www.gap-system.org`).

[**22**] D.W. Jones. A general theory of polyhedral sets and the corresponding $T$-complexes. *Dissertationes Math. (Rozprawy Mat.)*, 266:110, 1988.

[**23**] A. Libman. The gluing problem in the fusion systems of the symmetric, alternating and linear groups. *J. Algebra*, 341:209–245, 2011.

[**24**] G. Mislin. On the classifying space for proper actions. In *Cohomological methods in homotopy theory (Bellaterra, 1998)*, volume 196 of *Progr. Math.*, pages 263–269. Birkhäuser, Basel, 2001.

[**25**] N. Petrosyan and B. Putrycz. On cohomology of crystallographic groups with cyclic holonomy of split type. *J. Algebra*, 367:237–246, 2012.

[**26**] A.D. Rahm. The homological torsion of $\mathrm{PSL}_2$ of the imaginary quadratic integers. *Trans. Amer. Math. Soc.*, 365(3):1603–1635.

[**27**] A.D. Rahm and M. Fuchs. The integral homology of $\mathrm{PSL}_2$ of imaginary quadratic integers with nontrivial class group. *J. Pure Appl. Algebra*, 215(6):1443–1472, 2011.

[**28**] A.D. Rahm and R.J. Sánchez-García. GAP implementation of the Davis complex. 2015. (`http://www.gap-system.org/Packages/hap.html`).

[**29**] M. Röder. Geometric algorithms for resolutions for Bieberbach groups. In *Computational group theory and the theory of groups, II*, volume 511 of *Contemp. Math.*, pages 167–178. Amer. Math. Soc., Providence, RI, 2010.

[**30**] M. Salvetti. The homotopy type of Artin groups. *Math. Res. Lett.*, 1(5):565–577, 1994.

[**31**] R.J. Sánchez-García. Equivariant $K$-homology for some Coxeter groups. *J. Lond. Math. Soc. (2)*, 75(3):773–790, 2007.

[**32**] R.J. Sánchez-García. Bredon homology and equivariant $K$-homology of $\mathrm{SL}(3, \mathbb{Z})$. *J. Pure Appl. Algebra*, 212(5):1046–1059, 2008.

[**33**] S. Schönnenbeck. Data for complexes on which certain $PSL_2(\mathcal{O})$ act with nice stabilizer subgroups. *private communication*, 2013.

[**34**] C. Soulé. The cohomology of $\mathrm{SL}_3(\mathbf{Z})$. *Topology*, 17(1):1–22, 1978.

[**35**] C.C. Squier. The homological algebra of Artin groups. *Math. Scand.*, 75(1):5–43, 1994.

Bui Anh Tuan

Faculty of Mathematics and Computer Science, University of Science, VNU - HCM City, Vietnam

Graham Ellis `graham.ellis@nuigalway.ie`

School of Mathematics, National University of Ireland Galway, Ireland