Journal of Homotopy and Related Structures, vol. ??(??), ????, pp.1–33

COMPUTATIONAL HOMOTOPY OF FINITE REGULAR CW-SPACES

GRAHAM ELLIS AND FINTAN HEGARTY

Abstract

We describe a computational approach to the basic homotopy theory of finite regular CW-spaces, paying particular attention to spaces arising as a union of closures of n-cells with isomorphic face posets. Deformation retraction methods for computing fundamental groups, integral homology and persistent homology are given in this context. The approach is not new but our account contains some new features: (i) certain computational advantages of a permutahedral face poset are identified and utilized; (ii) the notion of *lattice complex* is introduced as a data type for implementing a general class of regular CW-spaces (including certain pure cubical and pure permutahedral subspaces of flat manifolds such as $\mathbb{R}^n$); (iii) *zig-zag homotopy retractions* are introduced as an initial procedure for reducing the number of cells of low-dimensional lattice spaces; more standard discrete vector field techniques are applied for further cellular reduction; (iv) a persistent homology approach to feature recognition in low-dimensional digital images is illustrated; (v) fundamental groups are computed; (vi) algorithms are implemented in the GAP system for computational algebra, allowing for their output to benefit from the system's vast library of efficient algebraic procedures.

Dedicated to Hvedri Inassaridze on his 80th birthday

1. Introduction

There is growing interest in efficient computer methods for calculating homotopical and homological properties of spaces. This is motivated by topics in applied topology such as topological data analysis [**1**], rigorous numerical analysis of dynamical systems [**18**], topological robotics [**4**], image analysis [**18**] and a range of

The first author wishes to thank Marian Mrozek and Pawel Dlotko at Uniwersytet Jagiellonski for helpful discussions that were supported by the National Science Centre (NCN) of Poland, grant N N201 419639. Both authors thank the referee and JHRS editors for their improvements to the article. The second author has been supported by a Science Foundation Ireland Mathematics Initiative PhD fellowship.

.

2000 Mathematics Subject Classification: 55N99

Key words and phrases: fundamental group, integral homology, persistent Betti numbers, computational topology

other applied topics [**6, 7, 5**]. It is also motivated by computational questions in the cohomology of groups [**9**] and theoretical algebraic topology [**23**].

In this paper we provide a self-contained account of an approach to computing homotopy theoretic properties of finite regular CW-spaces involving a large number of cells, paying particular attention to those spaces arising as a union of closures of n-cells with isomorphic face posets. The approach, which relies on a heavy use of deformation retracts and discrete vector fields, underlies the CAPD homology software [**20, 21**] for computer assisted proofs in dynamical systems and has been described in that context in [**18, 14, 15**] and elsewhere. It also underlies the first author's software package [**8**] for group cohomology and related homotopical algebra. The approach complements that of the software package PLEX [**22**] which is aimed primarily at persistent homology computations of filtered simplicial complexes and is based on efficient algorithms for row reduction of sparse matrices. It also complements the KENZO software package [**23**] which is aimed primarily at homology and homotopy groups of simply connected spaces arising in theoretical algebraic topology. However, in common with KENZO, we give a central computational role to deformation retracts and their induced chain maps.

An excellent treatment of computational homology has already been given in the textbook [**18**]. The approach described in this paper is essentially the same as that in [**18**] but our account emphasizes some new computational features that may be of general interest. In particular:

(i) We introduce the notion of *lattice complex* as a data type for implementing a range of algorithms on a general class of regular CW-spaces which we call *lattice spaces*. The class includes certain pure cubical and pure permutahedral subspaces of $\mathbb{R}^n$. The data type can also be used to implement certain pure cubical and permutahedral subspaces of other flat manifolds $\mathbb{R}^n/G$.

(ii) We identify two computational advantages of permutahedral cells over cubical cells. The first is that an n-cell in the standard permutahedral tessellation of $\mathbb{R}^n$ has fewer neighbours than an n-cell in the standard cubical tessellation. This allows us to extend a practical procedure for obtaining 'minimal' deformation retracts of cubical lattice spaces of dimension $\leqslant 3$ to the permutahedral lattice spaces of dimension $n \leqslant 4$. The second advantage is that our permutahedral lattice spaces are manifolds, and thus behave nicely with respect to taking complements.

(iii) We describe a persistent homology approach to feature recognition in low-dimensional digital images.

(iv) We apply *zig-zag homotopy retractions* as an initial procedure for reducing the number of cells of low-dimensional lattice spaces. More standard discrete vector field techniques are applied for further cellular reduction (cf. [**15**]).

(v) We compute presentations of fundamental groups of finite regular CW-spaces. Furthermore, we demonstrate how these presentations can be used to compute strong invariants of the fundamental group such as the integral homology of its nilpotent quotients, and the abelian invariants of its low-index subgroups.

(vi) We implement our algorithms in the GAP system for computational algebra. The output of our algorithms (which takes the form of finitely presented groups, abelian groups, graded algebras etc.) can thus benefit from GAP's vast library of efficient procedures for symbolic algebra. We also benefit from GAP's sophisticated method selection mechanism for selecting appropriated algorithms when computing topological properties for a range of high level data types.

We begin by recalling some basic theory of CW-spaces, and throughout we emphasize the suitability of the classical data types of algebraic topology (such as regular CW-space, cubical space, deformation retract, simple homotopy collapse, discrete vector field, relative homology group ...) and the importance of classical results (such as Propositions 2.1, 2.2 and Theorem 9.1) for implementing efficient and modular software. As one compelling motivation for the development of such software we include toy examples that illustrate a standard general approach to topological data mining based on persistent homology.

2. Finite regular CW-spaces

A good introduction to the theory of CW-spaces can be found in [**19**]. Recall that a CW-space X is *regular* if every cell is attached by a map which restricts to a homeomorphism on the boundary of the cell. Recall that X is *finite* if it has only finitely many cells.

We shall restrict attention to finite regular CW-spaces X because their cell structure, and hence their homotopy type, can be completely encoded in terms of a finite collection of binary valued incidence numbers and hence can easily be stored on a computer. We let X^k denote the k-skeleton of X, and e^k_j the jth cell of dimension k. Thus e^k_j is a subspace of X homeomorphic to an open Euclidean ball.

The space X can be represented as a sequence of lists $R^0, R^1, ..., R^n$, where the jth term of the list $R^k = \{r^k_1, r^k_2, ...\}$ records those $(k-1)$-dimensional cells of X that lie in the boundary of the jth k-dimensional cell of X. For algorithmic efficiency, it is best to encode some additional redundant information in r^k_j, namely the list of those $(k+1)$-dimensional cells of X whose boundaries contain the jth k-dimensional cell of X.

We define the *closure* of a cell e^k_j in X to be the smallest CW-subspace $\overline{e}^k_j$ in X containing e^k_j. More generally, we define the *closure* of a subset $Y \subset X$ to be the smallest CW-subspace $\overline{Y}$ in X containing Y.

The sequence $R^0, ..., R^n$ is just an encoding of the partial order on the set of cells e^k_j in X given by setting $e^k_j < e^{k'}_{j'}$ if $k < k'$ and e^k_j lies in the closure of $e^{k'}_{j'}$. This *face poset* determines the regular CW-space X up to homeomorphism. Figure 1 shows the Hasse diagram of the face poset of a CW-decomposition of the 2-sphere.

We say that a cell e^k_j is *maximal* if it does not lie in the closure of any $(k+1)$-dimensional cell. We define the *complement* of a maximal cell e^k_j in X to be the CW-space $\overline{X \setminus \overline{e}^k_j}$ arising as the closure of $X \setminus \overline{e}^k_j$.

Definition 2.1. We define the *contact complex* of a maximal cell e^k_j in X to be

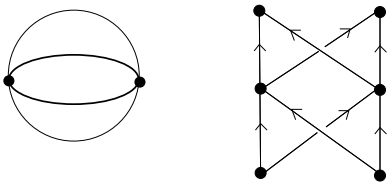


Figure 1: Regular CW-decomposition of the 2-sphere and corresponding face poset

the CW-space $Cont(e^k_j) = \overline{e}^k_j \cap \overline{X \setminus \overline{e}^k_j}$ arising as the intersection of the closure and complement of e^k_j. (See Figure 2.)

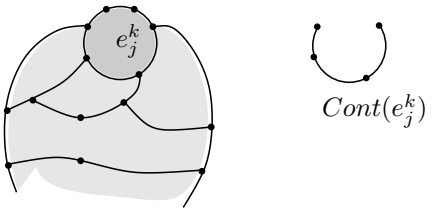


Figure 2: A cell e^k and its contact complex $Cont(e^k)$

Definition 2.2. We shall say that a maximal cell e^k_j in X is *redundant* if its contact complex $Cont(e^k_j)$ is a contractible CW-space.

Definition 2.2 is motivated by J.H.C. Whitehead's theory of *simple homotopy*, an account of which can be found in [**3**]. In this theory a CW-space Y is said to be obtained from a CW-space X by an *elementary collapse*, $X \searrow Y$, if

1. Y is a CW-subspace of X,
2. $X = Y \cup e^n \cup e^{n-1}$ with e^n, e^{n-1} cells not in Y,
3. there exists a standard ball pair (D^n, D^{n-1}), homeomorphic to $(\mathbb{I}^{n-1} \times \mathbb{I}, \mathbb{I}^{n-1} \times 1)$ with $\mathbb{I} = [0,1]$, and a characteristic map $\phi\colon D^n \to X$ for e^n which maps D^{n-1} to Y^{n-1},
4. the restriction of ϕ to the closure $\overline{\partial D^n \setminus D^{n-1}}$ of the boundary of D^n minus D^{n-1} is a characteristic map for e^{n-1} .

Simple homotopy is a combinatorial approach to homotopy theory based on the observation that Y is a deformation retract of X when $X \searrow Y$. (See [**3**] for details.)

If e^n is a redundant cell of a regular CW-space X then the contact complex $Cont(e^n)$ is homeomorphic to a standard ball D^{n-1}. Let e^{n-1} denote the union of those cells in the boundary of e^n not contained in $Cont(e^n)$. Then e^n is homeomorphic to an open ball and $X \searrow Y$ for Y the subspace of X obtained by removing e^n and e^{n-1}. The following proposition summarizes this statement.

Proposition 2.1. *If e_j^k is a redundant maximal cell in the finite regular CW-space X then the inclusion map $\overline{X \setminus \overline{e}_j^k} \hookrightarrow X$ is a homotopy equivalence, and $\overline{X \setminus \overline{e}_j^k}$ is a deformation retract of X.*

It can be computationally difficult to decide whether a given cell is redundant or not. However, for certain regular CW-spaces X one can readily design an efficient algorithm for recognising particular redundant cells in any CW-subspace of X. We shall call such an algorithm a *redundancy test* for X. If a redundancy test is able to recognise every redundant cell in any CW-subspace of X then we deem it to be *optimal*. Proposition 2.1 yields the following two trivial algorithms.

Algorithm 2.1.
Input: a finite regular CW-space X, a CW-subspace $A \subset X$, and an (optimal) redundancy test for X.
Output: a (minimal) CW-subspace $X' \subset X$ which is a deformation retract of X and which contains A.
Procedure:

initialise $X' := X$;
while X' has a recognisably redundant maximal cell e_j^k not in A
set $X' := \overline{X' \setminus \overline{e}_j^k}$;

Algorithm 2.2.
Input: a finite CW-space X, a non-empty CW-subspace $A \subset X$, and an (optimal) redundancy test for X.
Output: a (maximal) CW-subspace $X' \subset X$ which contains A as a deformation retract.
Procedure:

initialise $X' := A$;
while X has a cell e_j^k which is recognisably a redundant maximal cell of $X' \cup \overline{e}_j^k$
set $X' := X' \cup \overline{e}_j^k$;

Recall that the cellular chain complex

$$C_*(X): \quad \cdots \to C_k(X) \xrightarrow{\partial_k} C_{k-1}(X) \to \cdots \to C_0(X)$$

of a finite regular CW-space X is constructed by taking $C_k(X)$ to be the free abelian group with free generators corresponding to the k-cells of X, and with boundary homomorphisms given by

$$\partial_k(e_j^k) = \sum_i \epsilon_{ij}^k b_{ij}^k e_i^{k-1}$$

where

$$b_{ij}^k = \begin{cases} 1 & \text{if } e_i^{k-1} \text{ lies in the closure of } e_j^k \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\epsilon_{ij}^k = \begin{cases} 0 & \text{when } b_{ij}^k = 0, \\ 1 & \text{when } k = 1, b_{ij}^1 = 1, b_{i'j}^1 = 1, i < i', \\ -1 & \text{when } k = 1, b_{ij}^1 = 1, b_{i'j}^1 = 1, i > i', \\ \pm 1 & \text{when } k > 1, b_{ij}^k = 1, \text{with sign chosen to satisfy} \\ & \text{the equation } \partial_{k-1}\partial_k = 0. \end{cases}$$

Given a CW-subspace $A \subset X$, the inclusion map induces a chain monomorphism $C_*(A) \hookrightarrow C_*(X)$ which allows us to regard $C_*(A)$ as a sub chain complex of $C_*(X)$. One defines $C_*(X, A) = C_*(X)/C_*(A)$. The chain groups $C_k(X, A) = C_k(X)/C_k(A)$ are again free abelian with generators corresponding to those k-dimensional cells of X not contained in A.

When X and the subspace A both have a large number of cells, it may be impractical to represent the chain complexes $C_*(X)$, $C_*(A)$ on a computer and yet be quite practical to represent the chain complex $C_*(X, A)$.

For any abelian group $\mathbb{A}$ one defines the homology of the space X and pair (X, A) as

$$H_k(X; \mathbb{A}) = H_k(C_*(X) \otimes_{\mathbb{Z}} \mathbb{A}) \ ,$$

$$H_k(X, A; \mathbb{A}) = H_k(C_*(X, A) \otimes_{\mathbb{Z}} \mathbb{A}) \ .$$

The natural long exact sequence

$$\cdots \to H_k(A; \mathbb{A}) \to H_k(X; \mathbb{A}) \to H_k(X, A; \mathbb{A}) \to H_{k-1}(A; \mathbb{A}) \to \cdots$$

together with the homotopy invariance of homology readily yields the following standard computational tool.

Proposition 2.2. *(i) Let X be a finite regular CW-space. Let $X' \subseteq X$ be a deformation CW-retract, and let A be a contractible CW-subspace of X'. Then*

$$H_k(X; \mathbb{A}) \cong H_k(X', A; \mathbb{A}), \qquad k \geqslant 1.$$

(ii) Let $f\colon X \hookrightarrow Y$ be an inclusion of finite regular CW-spaces. Let X', A be as in (i), let Y' be a deformation CW-retract of Y, and let B be a contractible CW-subspace of Y' with $A \subset B$. Then the following commutative natural diagram

$$\begin{array}{ccc} H_k(X; \mathbb{A}) & \longrightarrow & H_k(Y; \mathbb{A}) \\ \downarrow \cong & & \downarrow \cong \\ H_k(X', A; \mathbb{A}) & \longrightarrow & H_k(Y', B; \mathbb{A}) \end{array}$$

exists for $k \geqslant 1$.

The statement of Proposition 2.2 needs a minor adjustment for the case $k = 0$. We leave this to the reader.

The remainder of the paper is structured as follows. In Sections 3-7 we describe a practical implementation of Algorithms 2.1 and 2.2 and Proposition 2.2 for a class of

regular CW-spaces which we call *lattice spaces*. These CW-spaces admit particularly efficient redundancy tests. Some details in Sections 3-7 are expressed through the slightly more general notion of a *tessellated space*. In Sections 9 and 10 we describe a more general homology implementation, based on the notion of discrete vector field, for arbitrary finite regular CW-spaces. We give examples in which best results are obtained using a mixture of the two implementations. Section 8 discusses the computation of homology homomorphisms induced by non-cellular maps of spaces. Section 11 describes a potential application to feature recognition in digital images. In Section 12 we illustrate the user interface to the implementations.

3. Tessellated spaces and lattice complexes

We use the term *tessellated space* to mean a regular CW-space X such that for some $n \geqslant 0$:

(i) X is the union of the closures of its n-cells and;

(ii) all closures of n-cells have face poset isomorphic to that of some fixed polytope.

For instance, all n-cells could be n-cubes, or they could all be n-simplices, or they might all be n-permutahedra. We say that such a space has *dimension* n and that the closure of any n-dimensional cell is a *facet* of the space. (We remark that our use of the term *facet* is consistent with standard terminology. Cells in a CW-space are homeomorphic to open balls. In contrast, for convex polytopes or polyhedra one usually works with closed cells called *faces*. A *facet* of an $(n+1)$-dimensional convex polytope or polyhedron is a face of dimension n. We think of an n-dimensional tessellated space X as lying in the boundary of some ambient $(n+1)$-dimensional space. We therefore view the closure of an n-cell in X as a facet of the ambient $(n+1)$-dimensional space.) Thus a tessellated space X is the union of its facets. We say that a tessellated space is *finite* if it has only finitely many facets.

The restrictions (i) and (ii) can lead to computational efficiencies. Such spaces often admit a more efficient computer representation than that given in Section 2 for arbitrary finite regular CW-spaces. Furthermore, for low values of n such spaces admit efficient redundancy tests for maximal cells.

Recall that a *finite simplicial space* X is a CW-subspace of a standard N-simplex Δ^N. Recall that a cellular space is *pure* if all inclusion-maximal cells have common dimension. A pure finite simplicial space is an example of a tessellated space.

Any pure CW-subspace of the standard N-cube $\mathbb{I}^N = [0,1]^N$ is an example of a tessellated space. A special class of such spaces - which we call a *cubical lattice space* - will be of particular interest to us.

Let L be a *lattice* in $\mathbb{R}^n$, *i.e.* an additive subgroup generated by n linearly independent vectors in $\mathbb{R}^n$. Any $v \in L$ determines a Dirichlet-Voronoi cell

$$D_L(v) = \{x \in \mathbb{R}^n : ||v - x|| \leqslant ||w - x|| \text{ for any } w \in L, w \neq v\}$$

from which Euclidean space $\mathbb{R}^n$ inherits the structure of a tessellated space with facets the cells $D_L(v)$. We denote this tessellated space by $\mathbb{R}^n_L$.

For the lattice C generated by n orthogonal vectors the facets of the tessellated space $\mathbb{R}^n_C$ are combinatorially equivalent to an n-cube.

To describe a second interesting lattice we identify $\mathbb{R}^n$ with the hyperplane in $\mathbb{R}^{n+1}$ consisting of all vectors whose coordinates sum to zero. Let P be the abelian group in this hyperplane generated by the columns of the $(n+1) \times (n+1)$ matrix

$$\begin{pmatrix} -n & 1 & 1 & \cdots & 1 \\ 1 & -n & 1 & \cdots & 1 \\ 1 & 1 & -n & \cdots & 1 \\ \vdots & & & & \\ 1 & 1 & 1 & \cdots & -n \end{pmatrix}.$$

The facets of the tessellated space $\mathbb{R}^n_P$ are combinatorially equivalent to the n-dimensional convex polytope known as the *permutahedron*. See Figure 3.

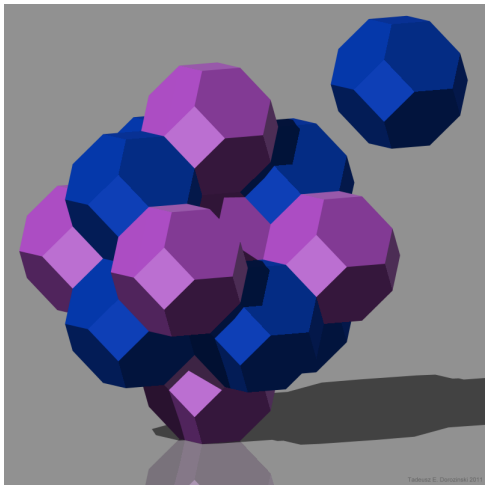


Figure 3: 3-dimensional permutahedral facets. (Image reprinted from [**24**].)

Recall that an n-dimensional *Bieberbach group* G is a group of isometries that act fixed-point freely and cocompactly on $\mathbb{R}^n$. The quotient $\mathbb{R}^n/G$ is a flat manifold with fundamental group G. Up to homeomorphism there are two such manifolds for dimension $n = 2$: the torus and Klein bottle. There are ten such manifolds for $n = 3$, and a list of Bieberbach groups in dimensions 4 or less is available in the crystallographic groups catalogue [**10**]. Let $L \subset \mathbb{R}^n$ be a lattice with n linearly independent generators and let G be some Bieberbach group whose action on $\mathbb{R}^n$ preserves the lattice L. The CW-structure of $\mathbb{R}^n_L$ will be inhereted by the quotient $\mathbb{R}^n/G$. Those cases where the CW-structure on $\mathbb{R}^n/G$ is regular provide examples of tessellated spaces with just finitely many facets. For example, the hexagonally tessellated Klein bottle of Figure 4 can be constructed from the permutahedral lattice $L = P \subset \mathbb{R}^2$ by taking G to be the group generated by a translation (sending facet B to facet B' in Figure 4) and a glide-reflection (sending facet A to facet A' in Figure 4).

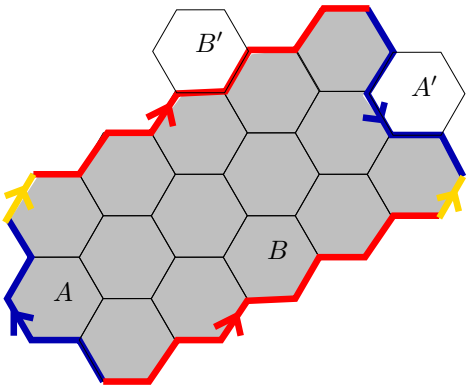


Figure 4: A hexagonally tiled Klein bottle

Definition 3.1. We define a *lattice space* to be a union of finitely many facets in $\mathbb{R}^n_L$. We define a *quotient lattice space* to be a union of finitely many facets in the quotient space $\mathbb{R}^n_L/G$ for any Bieberbach group G preserving L with regular CW-quotient. A lattice space or quotient lattice space is said to be *cubical* if $L = C$ and *permutahedral* if $L = P$.

Note that any cubical lattice space can be embedded in some unit cube $\mathbb{I}^N$ by using a Gray code to identify vertices of X with vertices of $\mathbb{I}^N$. Such a space X is thus an example of a pure CW-subspace of $\mathbb{I}^N$.

Any finite tessellated space X can be represented on a computer using the representation of a finite regular CW-space described in Section 2. The maximal cells have some common dimension n. The contact complex $Cont(e^n)$ of a maximal cell e^n is particularly easy to compute from the representation of X; it consists of those cells in the boundary of e^n that lie also in the boundary of at least one other n-cell.

The fact that all n-cells have isomorphic face poset can help in the implementation of efficient redundancy tests. Fix some maximal cell e^n. Let F denote the face poset of the boundary ∂e^n of e^n. Thus F is the poset of a regular CW-structure on the sphere S^{n-1}. Let $\mathcal{P}(F)$ denote the set of those subposets of F that arise from CW-subspaces of ∂e^n. If $\mathcal{P}(F)$ is not too large then it should be possible to decide and record, for each poset in $\mathcal{P}(F)$, whether or not the corresponding CW-subspace of ∂e^n is contractible. The contact complex $Cont(e'^n)$ of any maximal cell e'^n has face poset isomorphic to a poset in $\mathcal{P}(F)$. Assuming that this isomorphism can be quickly realised, we can use our pre-computed record of contractible posets in $\mathcal{P}(F)$ to quickly test the contractibility of $Cont(e'^n)$. Such a test is required in an implementation of Algorithms 2.1 and 2.2.

For instance, if each facet $\overline{e}^n$ is an n-simplex then there are precisely $k = 2^{n+1} - 2$ cells in ∂e^n, and fewer than 2^k posets in $\mathcal{P}(F)$. So for $n \leqslant 4$ it is practical to store a list of all the contractible simplicial posets in $\mathcal{P}(F)$. The poset F is isomorphic to the poset of subsets of $\{0, 1, \ldots, n\}$ and by using a total order on the set of

vertices of X one can efficiently compute this isomorphism. It should thus be quite practical to implement Algorithms 2.1 and 2.2, with optimal redundancy tests, for pure simplicial spaces of dimension $\leqslant 4$.

Lattice spaces admit a computer representation that is more efficient than the face poset representation for general regular CW-spaces. Given a lattice L in $\mathbb{R}^n$, we can choose a basis $b_1, \ldots, b_n \in L$ and represent any $v \in L$ uniquely as a combination $v = \lambda_1 b_1 + \cdots + \lambda_n b_n$ with integer coordinates λ_i. Let X be the tessellated space arising as a union of finitely many facets in $\mathbb{R}^n_L$. By applying a translation if necessary, we can assume that X is a union of facets $D_L(v)$ where v has only positive integer coordinates λ_i with respect to the chosen basis. We can represent X by an n-dimensional binary array $A = (a_{\lambda_1,\lambda_2\cdots,\lambda_n})$ with

$$a_{\lambda_1,\lambda_2\cdots,\lambda_n} = \begin{cases} 1 & \text{if } D(\lambda_1 b_1 + \cdots + \lambda_n b_n) \text{ lies in } X, \\ 0 & \text{otherwise.} \end{cases}$$

Our computer representation of X also needs to store some information about the lattice L. For our algorithms the most appropriate extra information can be stored as a finite set of integer vectors

$$B_L := \{\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{Z}^n \ : \ D_L(\underline{0}) \cap D_L(\lambda_1 b_1 + \cdots + \lambda_n b_n) \neq \emptyset\}$$

where $\underline{0}$ is the zero vector in L. We call B_L the *L-ball*.

Definition 3.2. A *lattice complex* consists of a pair $K = (A, B_L)$ with $A = (a_{\lambda_1,\lambda_2\cdots,\lambda_n})$ a binary array and B_L an L-ball for some n-dimensional lattice $L \subset \mathbb{R}^n$. The indices λ_i range over $1 \leqslant \lambda_i \leqslant m_i$ for some integer vector $m = (m_1, \ldots, m_n)$. We say that K has *dimension* n and *shape* m.

Cubical lattice spaces, and permutahedral lattice spaces, are both conveniently represented on a computer as lattice complexes.

Let $X \subset \mathbb{R}^n_L/G$ be a quotient lattice space. Choose a basis $b_1, \ldots, b_n$ for the finite abelian group $L/G \cong \mathbb{Z}_{m_1} \oplus \cdots \oplus \mathbb{Z}_{m_n}$. Any $v \in L/G$ is a unique combination $v = \overline{\lambda}_1 b_1 + \cdots + \overline{\lambda}_n b_n$ with coordinates $\overline{\lambda}_i \in \mathbb{Z}_{m_i}$. We can represent X by the *modular* lattice complex consisting of $A = (a_{\overline{\lambda}_1,\overline{\lambda}_2\cdots,\overline{\lambda}_n})$ and the L-ball B_L. More precisely:

Definition 3.3. A *modular lattice complex* consists of a pair $K = (A, B_L)$ with $A = (a_{\overline{\lambda}_1,\overline{\lambda}_2\cdots,\overline{\lambda}_n})$ a binary array and B_L an L-ball for some n-dimensional lattice $L \subset \mathbb{R}^n$. The indices $\overline{\lambda}_i$ range over $\overline{\lambda}_i \in \mathbb{Z}/m_i\mathbb{Z}$ for some integer vector $m = (m_1, \ldots, m_n)$.

Note that by working in the compact manifold $\mathbb{R}^n_L/G$ rather than Euclidean space $\mathbb{R}^n_L$ we are able to reduce the ambient dimension of the binary array when representing certain basic spaces such as Klein bottles, tori and their products. Lattice complexes and modular lattice complexes are particularly well suited to efficient implementations of Algorithms 2.1 and 2.2. An obvious advantage to this representation of a space X is that only a record of the facets of X is stored, rather than the full cell poset.

To present details of a practical redundancy test for low-dimensional lattice complexes, we introduce some notation and terminology. Let X be a tessellated space and let e, f be two of its facets. (Recall that we use the term *facet* to mean the closure of a top-dimensional cell.) We say that e is a *neighbour* of f if $e \cap f \neq \emptyset$. In particular, f is a neighbour of itself. We define the *neighbourhood* of f to be the union $N_X(f)$ of all neighbours of f. Let Y be a tessellated subspace of X (*i.e* some union of facets of X). We define the *neighbourhood* of Y to be the union $N_X(Y)$ of all the neighbourhoods $N_X(f)$ of facets f of Y. We note that $N_X(Y)$ is again a tessellated subspace of X. For a facet f in X we define the *complementary neighbourhood* $\hat{N}_X(f)$ to be the union of all neighbours of f except for the facet f itself. We denote by $X - f$ the union of all facets of X except for the facet f itself. Thus $\hat{N}_X(f) = N_X(f) - f$.

Definition 3.4. We shall say that a tessellated space X is *locally contractible* if, for each of its facets f, the neighbourhood $N_X(f)$ is contractible.

The following lemma is immediate.

Lemma 3.1. *Let e^n_j be a maximal cell of a locally contractible tessellated space X and let e be the facet of X arising as the closure of e^n_j. The contact complex $Cont(e^n_j)$ is a deformation CW-retract of the complementary neighbourhood $\hat{N}_X(e)$.*

Lattice spaces $X \subset \mathbb{R}^n_L$ (and many quotient lattice spaces $X \subset \mathbb{R}^n_L/G$) are locally connected. So Lemma 3.1 implies that a maximal cell in X is redundant if and only if the complementary neighbourhood of its closure is contractible. Up to translation in $\mathbb{R}^n_L$ (or $\mathbb{R}^n_L/G$) there are only a finite number of possible complementary neighbourhoods in X. These complementary neigbourhoods can be represented as certain binary arrays $(a_{\lambda_1 \cdots \lambda_n})$ with $\lambda_i \in \{-1, 0, 1\}$. For low values of n, it is possible to record precisely which of these binary arrays correspond to contractible complementary neighbourhoods, and this record can be used to test redundancy. We give details for some particular cases.

We first consider the cubical setting. A facet f in a cubical lattice space $X \subset \mathbb{R}^n_C$ has $3^n - 1$ possible neighbours different from f. The number of possible complementary neighbourhoods $\hat{N}_X(f)$, up to translation in $\mathbb{R}^n$, is thus 2^{3^n-1}. For $n = 2$ there are 256 possible complementary neighbourhoods $\hat{N}_X(f)$, of which precisely 116 are contractible. For $n = 3$ there are 67108864 possible complementary neighbourhoods $N_X(f)$, of which precisely 41123720 are contractible. An optimal redundancy test for $n = 2, 3$ can be implemented by recording all possible contractible complementary neigbourhoods and, given any facet f in X, checking if $\hat{N}_X(f)$ is recorded as contractible. For $n \geqslant 4$ it is practical to record only some of the contractible complementary neighbourhoods (on say a standard PC with only a few gigabytes of RAM). So only a non-optimal redundancy test can be implemented in this way for $n \geqslant 4$.

We now consider the permutahedral setting. A facet f in a permutahedral lattice space $X \subset \mathbb{R}^n_P$ has $2^{n+1} - 2$ possible neighbours different to f. The number of possible complementary neighbourhoods, up to translation, is thus $2^{2^{n+1}-2}$. For $n = 2$ there are 64 possible complementary neighbourhoods $\hat{N}_X(f)$, of which 30 are

contractible. For $n = 3$ there are 16384 possible complementary neighbourhoods $\hat{N}_X(f)$, of which 7500 are contractible. For $n = 4$ there are 1073741824 possible complementary neighbourhoods $\hat{N}_X(f)$, of which 280694770 are contractible. When searching for deformation retracts of subspaces of $\mathbb{R}^4$ there is thus a computational advantage to working with permutahedral lattice spaces rather than cubical lattice spaces since in the permutahedral case the list of contractible complementary neighbourhoods can be stored.

From the lattice complex representation of a lattice space $X \subset \mathbb{R}^n_L$ or quotient lattice space $X \subset \mathbb{R}^n_L/G$ one can cheaply compute the neighbourhood of any facet and hence directly compute the number β_0 of path-components in X and homology $H_0(X; \mathbb{A}) \cong \mathbb{A}^{\beta_0}$. The following obvious algorithm can be used for computing β_0.

Algorithm 3.1.
Input: finite tessellated space X represented so that neighbourhoods of facets can be cheaply computed.
Output: list $X_1, \ldots, X_c$ of the path components of X.
Procedure:

initialise $c := 0$;
deem all facets of X to be uncoloured and no facets to be coloured;
while there exists an uncoloured facet of X
    set $c := c + 1$;
    choose some uncoloured facet and assign it the colour c;
    while there exists an uncoloured facet f in the neighbourhood of some facet of colour c
        assign the colour c to f;
for $0 \leqslant i \leqslant c$ let X_i be the union of the facets of colour i;

4. Two toy examples

Lattice spaces arise naturally in applied topology. As a first illustration we consider a toy example of the following general problem.

> Given a set S of points randomly sampled from an unknown manifold M, what can we infer about the topology of M?

For the toy example we suppose that $M \subseteq \mathbb{R}^2$ and consider the random sample S of 57906 distinct points $(x, y) \in \mathbb{N}^2$ represented in Figure 5. Let C denote the lattice spanned by $(1, 0)$ and $(0, 1)$. We associate to S the lattice space X_1 consisting of those facets $D_C(v)$ of $\mathbb{R}^2_C$ with $v \in S$. A standard approach to understanding M is to produce a sequence $X_1 \subset X_2 \subset X_3 \subset \cdots$ of successive neighbourhoods $X_{i+1} = N_{\mathbb{R}^2_C}(X_i)$ and search for "persistent" topological properties in the sequence. Several of the spaces X_i are illustrated in Figure 6.

The hope is that persistent properties would reflect properties of the unknown manifold M. In this paper we are particularly interested in homology groups and Betti numbers. The Betti numbers $\beta_i(X)$ are the rank of the torsion free subgroup of the homology $H_i(X; \mathbb{Z})$ and are recorded in Table 1 for our sequence of spaces.

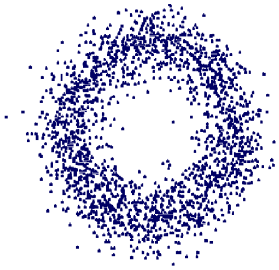


Figure 5: Sample S of points from an unkown manifold $M \subset \mathbb{R}^2$

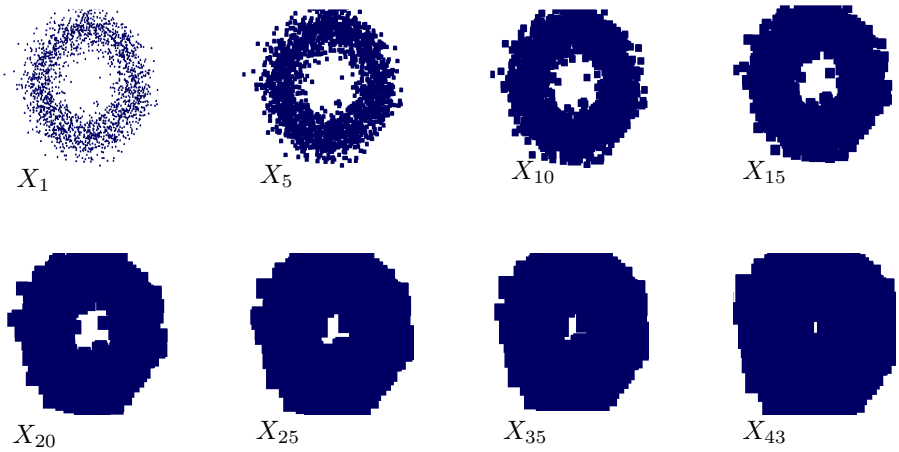


Figure 6: Various cubical neighbourhoods of the set S

The numbers in Table 1 give only a partial indication of homological persistence. They indicate for instance that $H_1(X_5;\mathbb{Z})$ is generated by 115 independent cycles and $H_1(X_{10};\mathbb{Z})$ is generated by 18 cycles. But they do not indicate how many of the 115 cycles lie in the kernel of the induced homology homomorphism $H_1(X_5;\mathbb{Z}) \to H_1(X_{10};\mathbb{Z})$.

In order to better capture the persistence of homology cycles one defines the *persistence Betti numbers* β_n^{ij}. For $i \leqslant j$ one sets β_n^{ij} equal to the rank of the torsion free subgroup of the image of the induced homomorphism $H_n(X_i;\mathbb{Z}) \to H_n(X_j;\mathbb{Z})$. For $i > j$ one sets $\beta_n^{ij} = 0$.

Following [**2**] we represent the matrix (β_n^{ij}) by a graph, called a *bar code*, with horizontal edges and vertices arranged in columns. The ith column has $\beta_n^{ii} = \beta_n(X_i)$ vertices. There are β_n^{ij} paths from the ith column to the jth column.

When a bar code has many rows it is convenient to represent a collection of n horizontal paths with common starting column and common finishing column as a

	X_1	X_5	X_{10}	X_{15}	X_{20}	X_{25}	X_{35}	X_{43}
β_0	455	32	9	2	1	1	1	1
β_1	0	115	18	4	1	1	1	1

Table 1: Betti numbers for various neighboorhoods of X_1

single line labelled by the integer n. We say that such as bar code is in *compact form.* The β_1 bar code, in compact form, for our toy example is illustrated in Figure 7. The

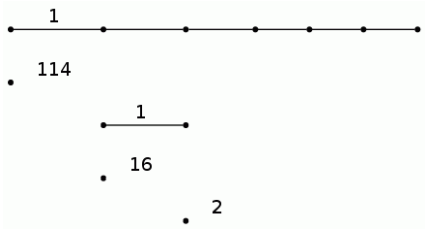


Figure 7: β_1 bar code for the spaces in Figure 6

single long horizontal path indicates that there is a single 1-dimensional homology generator that persists from X_5 to X_{43}. The suggestion is thus that the set S was randomly sampled from a manifold $M \subset \mathbb{R}^2$ with $H_1(M;\mathbb{Z}) = \mathbb{Z}$.

As a second illustration of the use of lattice spaces in applied topology we follow [**18**] in considering digital images. The digital image of Figure 8 can be represented

Figure 8: A digital image with 3264×2448 pixels.

as a 5-dimensional cubical lattice space whose facets are centred on the integer vectors (x, y, r, g, b) with x, y the Euclidean coordinates of a pixel, and with r, g, b integers in the range 0 to 255 representing the red/green/blue colour of the pixel.

A 2-dimensional cubical lattice space Y_t can be obtained by choosing a threshold number t and including one 2-dimensional facet centred at (x, y) for each pixel with $r + g + b \leqslant t$. (The colour white has $r = 255$, $g = 255$, $b = 255$.) Algorithm 3.1 can be used to compute the number $\beta_0(Y_t)$ of path components of Y_t for various thresholds t. Figure 9 shows a plot of $\beta_0(Y_t)$ against thresholds $r + g + b \leqslant t = 5n$ in the range $0 < t < 620$, $0 < n < 124$. For thresholds in the region of $t = 300$ the value of $\beta_0(Y_t)$ is comparatively stable. As the threshold t increases new path

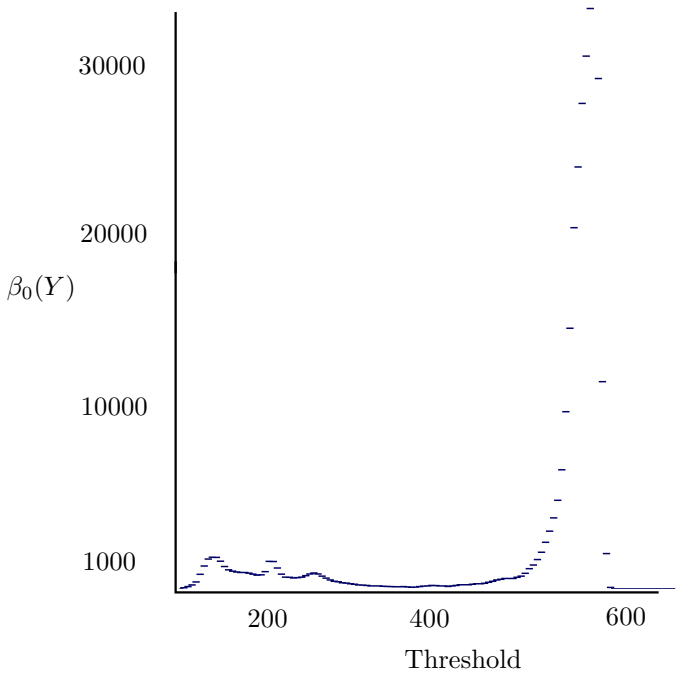


Figure 9: Plot of threshold t versus $\beta_0(Y_t)$ for Y_t from Figure 8.

components are born, and groups of existing path components coalesce to form single path components. The graph of $\beta_0(Y_t)$ thus increases and decreases. The number $\beta_0^{st}(Y)$ of path components that persist from Y_s to Y_t can be obtained by using Algorithm 3.1 to determine the number of path components of Y_t that intersect non-trivially with the space Y_s. For large $|t - s|$ this should be a robust estimate of the number of objects in the image that would be detected by the human eye. Indeed, a computation determines

$$\beta_0(Y_{200}) = 362,\ \beta_0(Y_{400}) = 629,\ \beta_0^{200,400}(Y) = 20.$$

The value of $\beta_0^{200,400}(Y)$ agrees with the 20 visible objects in the digital image.

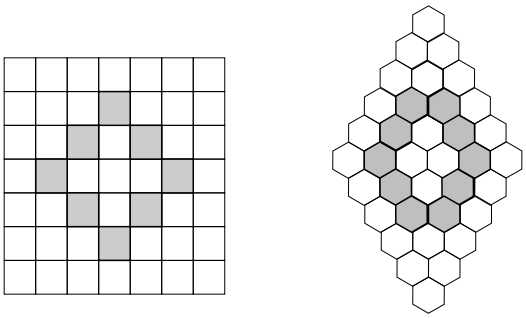
There are two homotopy types to the "visual path components" in the image of Figure 8, one being the homotopy type of a point (coins and bolts), the other being the homotopy type of a circle (nuts and washers). The persistence Betti numbers

$\beta_1^{st}(Y)$ can be used to distinguish these two types of path component. In Section 11 we explain how Betti numbers can also be used to identify geometric features that would distinguish washers from nuts.

We mention one drawback of a cubical approach to digital image analysis. We might wish to compute $\beta_1(Y_t)$ using Algorithm 3.1 and an isomorphism $H_1(Y_t;\mathbb{Z}) \oplus \mathbb{Z} \cong H_0(\mathbb{R}^2 \setminus Y_t;\mathbb{Z})$. Since the complement $\mathbb{R}^2 \setminus Y_t$ is not a CW-space it is tempting to obtain a CW-approximation using the following easily implemented definition.

Definition 4.1. Let $K = (A, B_L)$ be a lattice complex. We define the *combinatorial complement* of K to be the lattice complex $K^c = (A^c, B_L)$ where $A^c = (a'_{\lambda_1,\ldots,\lambda_n})$ is the binary array defined by $a'_{\lambda_1,\ldots,\lambda_n} = 1$ if $a_{\lambda_1,\ldots,\lambda_n} = 0$, $a'_{\lambda_1,\ldots,\lambda_n} = 0$ if $a_{\lambda_1,\ldots,\lambda_n} = 1$.

Lattice complexes correspond to lattice spaces and so this definition provides a notion of *combinatorial complement* for lattice spaces. However, Figure 10(a) shows a cubical lattice space which is homotopic to a circle but whose combinatorial complement is path connected and non-simply connected. This illustrates that the combinatorial complement of a cubical lattice space may not be homotopic to the actual complement.



(a) (b)

Figure 10: A cubical and permutahedral circle.

Permutahedral lattice spaces have the advantage of always being manifolds since, in the standard permutahedral tessellation of $\mathbb{R}^n$, two distinct permutahedral facets can intersect only in a face of dimension $n-1$. Consequently the notion of combinatorial complement works well for permutahedral lattice spaces and can be used to benefit from Alexander duality.

One aim of this paper is the computation of persistence Betti numbers and homology homomorphisms arising from a sequence of inclusions of regular CW-spaces. The classical algorithm due to Zomorodian and Carlsson [**25**] can always be used for this computation. We describe an alternative implementation [**8, 16**] based on the deformation retract approach outlined in Sections 2 and 3. This approach is particularly well suited to tessellated subspaces of $\mathbb{R}^n_L$, and to integral homology

computations. Furthermore, the approach is easily extended to the computation of other homotopy invariants such as (presentations of) fundamental groups.

Our first toy example above illustrates how the obvious naive approach to homology computation is likely to run into difficulties. Consider for example the space X_{43} of Figure 6 which involves 296998 2-dimensional facets, 595267 1-dimensional edges, and 298269 0-dimensional vertices. In its cellular chain complex

$$C_2(X_{43}) \xrightarrow{\partial_2} C_1(X_{43}) \xrightarrow{\partial_1} C_0(X_{43})$$

the boundary homomorphism ∂_2 is represented by a 595267×296998 integer matrix and the boundary homomorphism ∂_1 is represented by a 298269×595267 integer matrix. A direct computation of $H_1(X_{43};\mathbb{Z}) = \ker \partial_1 / \text{image}\, \partial_2$ using the Smith Normal Form algorithm would be time consuming.

There are of course methods for calculating $H_1(X_{43};\mathbb{Z})$ without computing kernels and Smith Normal Form. (For any tessellated subspace $X \subset \mathbb{R}^2$ we have $H_1(X;\mathbb{Z}) = \mathbb{Z}^{\beta_1(X)}$ where $\beta_1(X) = \beta_0(X) + \chi(X)$, $\beta_0(X)$ being the number of path components of X and $\chi(X)$ being its Euler characteristic. Both $\beta_0(X)$ and $\chi(X)$ are easily computed.) However, such methods do not easily extend to higher-dimensional spaces.

An implementation of Algorithm 2.1 was applied to the toy space X_{43} involving 296990 cubical facets, with $A = \emptyset$, and resulted in the deformation retract X'_{43} of Figure 11 involving 570 facets. The retract is 2-dimensional but its facets are so small as to give it the appearence of a 1-dimensional space. The diagonal shape of the retract is not significant; it is just a consequence of certain choices in our implementation of Algorithm 2.1. Full details of the retract can be seen by enlarging Figure 11 in the electronic version of this article. An implementation of Algorithm

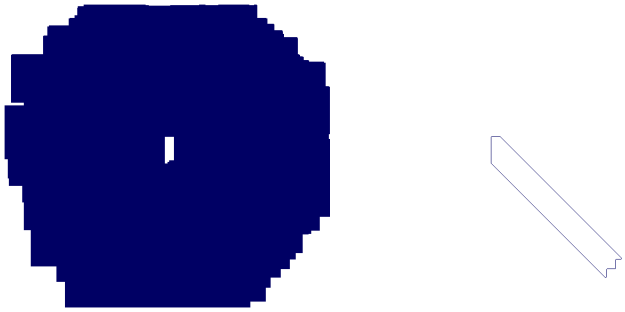


Figure 11: The cubical lattice space X_{43} and a deformation retract X'_{43}

2.2 was applied to the deformation retract X'_{43} of Figure 11, with A consisting of a single randomly chosen facet of X'_{43}, and resulted in a contractible subspace X''_{43} involving 569 facets. The chain complex $C_*(X'_{43}, X''_{43})$ has just one generator in degree 2, three generators in degree 1, and one generator in degree 0. We can use it to compute

$$H_1(X_{43};\mathbb{Z}) \cong H_1(X'_{43};\mathbb{Z}) \cong H_1(C_*(X'_{43}, X''_{43})) = \mathbb{Z}\ .$$

The computation took about 1 second on a 1GHz Linux laptop with 1GB of RAM.

5. Zig-zag retracts

For a poset F we define a *tessellated F-space* to be a tessellated space whose facets have face posets isomorphic to F. There are a number of computational advantages to working in the category of tessellated F-spaces and cellular maps. However, in the final stage of the computation of homotopy invariants of a space one usually has to use its CW-structure, and this structure can be extremely large for a tessellated F-space X. Algorithm 2.1 can be used to search for a small deformation retract $Y \subset X$ in the category of tessellated F-spaces. But often there will be other much smaller tessellated F-spaces X' that are homotopy equivalent to X yet not contained in X. (Consider a 2-dimensional annulus X with large inner radius, as in Figure 12. Any

Figure 12: Digital image of an annulus

retract will be large, and yet the annulus is homotopy equivalent to a circle of small radius.) We describe an elementary procedure for searching for such spaces X'.

We say that a tessellated F-space X' is a *zig-zag retract* of a tessellated F-space X if there exists a sequence of deformation retracts

$$X = B_0 \hookleftarrow A_1 \hookrightarrow B_1 \hookleftarrow A_2 \hookrightarrow B_2 \hookleftarrow \cdots \hookrightarrow B_{k-1} \hookleftarrow A_k = X'$$

in the category of tessellated F-spaces. Any implementation of Algorithms 2.1 and 2.2 could be used to find zig-zag retracts that are potentially smaller than actual retracts. To this end, let us say that a tessellated F-space W is a *bounding space* for X if W contains X as a cellular subspace. In practice we choose only bounding spaces that are contractible, but there is no theoretical necessity for this choice. (In the case of a 2-dimensional annulus X we would choose W to be a circular disk of radius equal to the outer radius of X.)

To produce a zig-zag retract of a tessellated space $X = B_0$ we can first use Algorithm 2.1 to produce a retract A_1. We can then construct a small bounding space $A_1 \subset W_1$. Then using Algorithm 2.2 we can compute a subspace $B_1 \subset W_1$ which is maximal with respect to containing A_1 as a deformation retract. We can then use Algorithm 2.1 to find a retract $A_2 \subset B_1$. While A_{n+1} has fewer facets than A_n this process can be repeated.

An illustration of a zig-zag retract is given in Figure 13. The above method was applied to a 2-dimensional cubical lattice space with 58570 facets. It resulted in a homotopy equivalent cubical lattice space with 172 facets. In the middle space the second path component has been retracted to a cubical lattice space with just four facets and is thus not visible in the figure at normal resolution. Details of these 2-

dimensional spaces can be obtained by enlarging Figure 13 in the electronic version of this article.

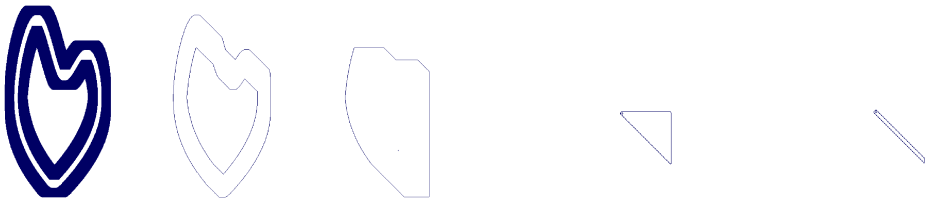


Figure 13: A zig-zag homotopy retract

6. Homology homomorphism of a strictly cellular map

We say that a continuous map $\phi\colon X \to Y$ of regular CW-spaces is *strictly cellular* if it maps each cell $f \in X$ onto some cell $e = \phi(f) \in Y$. The cell e may be of a lower dimension than the cell f, but not of higher dimension. Such a map induces a chain homomorphism $\phi_*\colon C_*(X) \to C_*(Y)$ which is readily implemented on a computer. In order to compute invariants of the induced homology homomorphism $H_k(\phi_*)\colon H_k(X;\mathbb{Z}) \to H_k(Y;\mathbb{Z})$ we can use Proposition 2.2(ii).

This approach was used to obtain the bar code of Figure 7. More precisely, let $X_1 \subset X_2 \subset X_3 \subset \cdots$ be a sequence of inclusions of finite tessellated spaces. In order to determine the persistence Betti numbers β_k^{ij} is suffices to: (i) compute the induced homology homomorphisms $\iota_{i,i+1}\colon H_k(X_i;\mathbb{Z}) \to H_k(X_{i+1};\mathbb{Z})$; (ii) use standard linear algebra algorithms to compute the composite homomorphisms $\iota_{i,j} = \iota_{j-1,j} \cdots \iota_{i+1,i+2}\iota_{i,i+1}$; (iii) and then determine β_k^{ij} as the rank of the homomorphism $\iota_{i,j}$.

The above approach requires tessellated spaces $X_1' \subset X_2' \subset X_3' \subset \cdots$ and $X_1'' \subset X_2'' \subset X_3'' \subset \cdots$ for which X_i'' is a contractible subset of X_i' and X_i' is a deformation retract of X_i. We then benefit from the isomorphisms $H_k(X_i;\mathbb{Z}) \cong H_k(C_*(X_i', X_i''))$, $k \geqslant 1$. We can construct the tessellated spaces X_i', X_i'' as follows. Use Algorithm 2.1 to find a minimal deformation retract $X_1' \subset X_1$. Then use Algorithm 2.1 to inductively construct a minimal deformation retract $X_i' \subset X_i$ subject to $X_{i-1}' \subset X_i'$. Then use Algorithm 2.2 to construct a maximal contractible subspace $X_1'' \subset X_1'$. (Alternatively one could construct X_1'' to be a maximal acyclic subspace, though examples suggest that this does not necessarily improve performance.) Finally use Algorithm 2.2 to construct a maximal subspace $X_i'' \subset X_i'$ subject to $X_{i-1}'' \subset X_i''$ being a deformation retract.

An implementation of this construction was applied to the sequence of cubical

lattice spaces in Figure 6. The number of facets in the resulting spaces is given in Table 2. The table shows that even in this toy example there can be a signifi-

$i =$	1	5	10	15	20	25	35	43
X_i	57906	155013	201310	226845	245255	259301	282183	296998
X_i'	455	51110	123168	144871	156316	157721	165108	171658
X_i''	12	49607	122748	144547	155717	157121	164509	170805

Table 2: Number of facets in spaces X_i, X_i', X_i''

cantly greater number of facets in X_i' than in X_i'', leading to large chain complexes $C_*(X_i', X_i'')$ that require significant linear algebra computations. To improve the efficiency of the computation one should search for regular CW-spaces Y_i', Y_i'' that are not tessellated but which satisfy $X_i'' \subset Y_i'' \subset Y_i' \subset X_i'$ with X_i'' a deformation retract of Y_i'' and Y_i' a deformation retract of X_i'. The discrete vector field approach described in Section 9 can be used to find suitable Y_i', Y_i''.

7. A third example

The above persistence techniques were applied to a set S of 400 integer points randomly chosen from a subspace of the 3-dimensional region $[0, 140] \times [0, 140] \times [0, 20] \subset \mathbb{R}^3$. The set S was represented as a 3-dimensional cubical lattice space Y_1 with one facet centred on each point in S. Successive neighbourhoods $Y_1 \subset Y_2 \subset \cdots \subset Y_{20}$ were constructed, with the space Y_{20} consisting of 313361 facets. The β_0 and β_1 bar codes for the sequence $Y_4 \subset Y_8 \subset Y_{12} \subset Y_{16} \subset Y_{20}$ are shown in Figure 14; the second homology of all but the first space is trivial and so there is no persistent homology in degree 2. The space Y_{20} contains a cubical lattice space

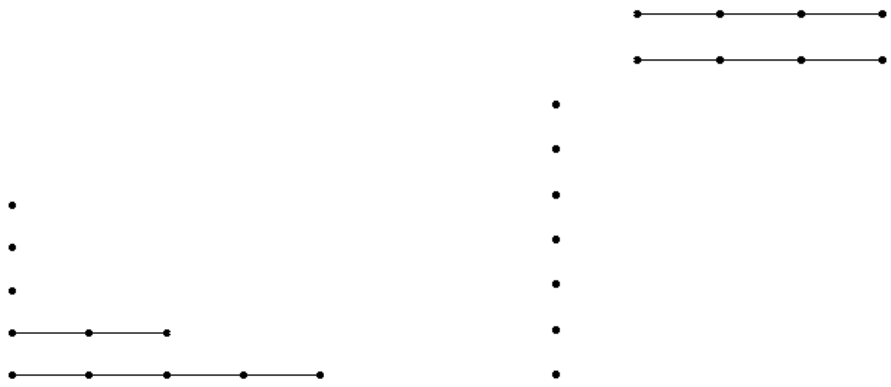


Figure 14: β_0 (left) and β_1 (right) bar codes for 3-dimensional cubical spaces Y_i

retract with just 220 facets.

The set S of 400 points was also used to construct an abstract graph G_{30} with vertex set S, and with an edge between vertices $(x_1, x_2, x_3), (x'_1, x'_2, x'_3) \in S$ if $|x_1 - x'_1| + |x_2 - x'_2| + |x_3 - x'_3| < 30$. The Vietoris-Rips complex $K(G_{30})$ was constructed for this graph. This is the simplicial complex whose simplices are the subsets $\sigma \subset S$ belonging to complete subgraphs of G_{30}. There are 400 vertices, 9647 edges, 128300 2-simplices and 1183632 3-simplices in $K(G_{30})$. The techniques described in Section 2 were used to compute the homology groups $H_0(K(G_{30}); \mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}$, $H_1(K(G_{30}); \mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}$ and $H_2(K(G_{30}); \mathbb{Z}) = 0$. These homology groups are in keeping with the bar codes of Figure 14 and suggest that S was sampled from a region of $\mathbb{R}^3$ having the homology of: (i) a disjoint union of two circles or alternatively, (ii) a disjoint union of a point and a wedge of two circles. A picture of the graph G_{30} is shown in Figure 15 and supports scenario (i). The picture was produced using the GRAPHVIZ software [**13**].

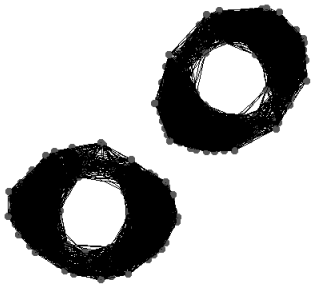


Figure 15: A GRAPHVIZ representation of the abstract graph G_{30}

8. Cellular approximation

Any continuous map $f\colon X \to Y$ of topological spaces is determined by its graph $\Gamma(f) = \{(x, f(x)) \in X \times Y\} \subset X \times Y$. The homeomorphism $X \to \Gamma(f), x \mapsto (x, f(x))$ means that topological properties of f are transfered to the map $\Gamma(f) \to Y, (x, f(x)) \mapsto f(x)$. For computing homotopical invariants of f it suffices to consider the map

$$\overline{f}\colon \overline{\Gamma(f)} \to Y, (x, y) \mapsto y$$

where $\overline{\Gamma(f)}$ is any subspace of $X \times Y$ containing $\Gamma(f)$ as a deformation retract. We say that such a map $\overline{f}$ is a *homotopy approximation* to f. For computing homology homomorphisms induced by f it suffices for $\overline{\Gamma(f)}$ to contain $\Gamma(f)$ and for the projection

$$\pi\colon \overline{\Gamma(f)} \to X, (x, y) \mapsto x$$

to induce homology isomorphisms

$$H_n(\pi)\colon H_n(\overline{\Gamma(f)}; \mathbb{Z}) \xrightarrow{\cong} H_n(X; \mathbb{Z}), \quad n \geqslant 0. \tag{1}$$

When (1) holds we say that $\overline{f}$ is a *homology approximation* to f.

If X is a tessellated F-space and Y a tessellated F'-space for posets F, F' then the direct product $X \times Y$ is a tessellated $F \times F'$-space. Moreover, the projections $X \times Y \to X, (x,y) \mapsto x$, $X \times Y \to Y, (x,y) \mapsto y$ are strictly cellular. For any tessellated subspace $A \subset X \times Y$ the projection restricts to a strictly cellular map $A \to Y$.

A homotopy or homology approximation $\overline{f}\colon \overline{\Gamma(f)} \to Y, (x,y) \mapsto y$ is said to be *strictly cellular* when $\overline{\Gamma(f)}$ is a cellular subspace of $X \times Y$.

One can attempt to compute a strictly cellular homology approximation as follows.

1. Choose a tessellated subspace $A \subset X \times Y$ which is known to contain the graph $\Gamma(f)$.
2. Compute the homology homomorphisms $H_n(\pi)\colon H_n(A;\mathbb{Z}) \longrightarrow H_n(X;\mathbb{Z})$, $n \geqslant 0$, induced by the strictly cellular projection $\pi\colon A \to X$.
3. If $H_n(\pi)$ is an isomorphism for $n \geqslant 0$ then set $\overline{\Gamma(f)} = A$. Otherwise restart with a refined choice of A.

9. Regular CW-spaces and discrete vector fields

When Algorithms 2.1 and 2.2 are used to compute a ziz-zag retract of a tessellated space X the result is a homotopy equivalent tessellated space X' with potentially fewer cells. There usually exist smaller CW-retracts of X' that are not tessellated subspaces. To take advantage of these smaller retracts we need to choose an appropriate represention for CW-subspaces of X' and design an algorithm for finding general CW-retracts of X'.

Cellular subspaces of tessellated spaces form a broad class of computationally accessible regular CW-spaces. The HAP software package [**8**] uses the following two computer representations of them.

1. The representation of a finite regular CW-space describe in Section 2 can be used.
2. For a lattice L with basis $b_1, \ldots, b_n$, and lattice space $X \subset \mathbb{R}^n_L$, the centre of any cell (of arbittrary dimension) in X is a point $\lambda_1 b_1 + \ldots + \lambda_n b_n \in \mathbb{R}^n$. By scaling and translating appropriately we can assume that the coefficients λ_i are positive integers. We can then represent a CW-subspace of X as an n-dimensional binary array $(a_{\lambda_1,\ldots,\lambda_n})$ with $a_{\lambda_1,\ldots,\lambda_n} = 1$ if and only if $\lambda_1 b_1 + \ldots + \lambda_n b_n$ is the centre of a cell in X. This representation is particularly convenient in the case when L is the cubical lattice C.

The language of discrete Morse theory is useful for describing an algorithm to find CW-retracts. In particular, a *discrete vector field* on a regular CW-space X is a collection of arrows $\alpha\colon s \to t$ where

(i) s, t are cells which are said to be *involved* in the arrow α, and any cell is involved in at most one arrow;

(ii) $\dim(t) = \dim(s) + 1$;

(iii) s lies in the boundary of t.

An example of a discrete vector field on a cubical lattice space is given in Figure 16. A sequence of arrows $\alpha_1\colon s_1 \to t_1$, $\alpha_2\colon s_2 \to t_2$, ... in a vector field is deemed

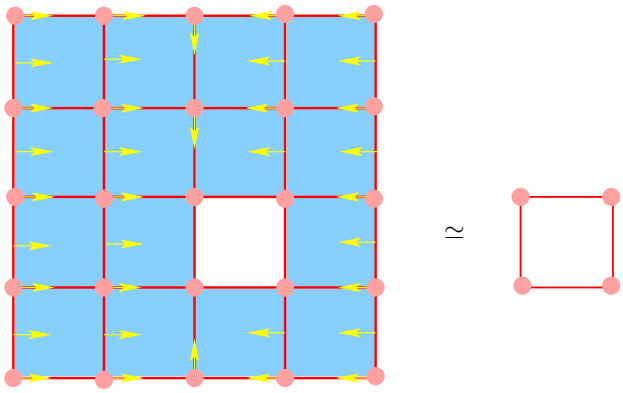


Figure 16: Discrete vector field on a cubical lattice space (left) and deformation retract (right)

to be a *path* if

(i) the s_i are cells of common dimension d say, and the t_i are cells of common dimension $d+1$;

(ii) each s_{i+1} lies in the boundary of t_i.

A discrete vector field is said to be *admissible* if it contains no infinite path of arrows and contains no finite cycles. The example in Figure 16 is admissible. A cell is deemed to be *critical* if it is involved in no arrows of the vector field.

We remark that the notion of a discrete vector field was called a *marking* in the PhD work of D. Jones [**17**]. A good account of the following "fundamental theorem" of discrete Morse theory can be found in [**11**], though the theorem itself essentially dates back to work of J.H.C. Whitehead on simple homotopy theory.

Theorem 9.1. *If X is a regular CW-space with an admissible discrete vector field then there is a homotopy equivalence*

$$X \simeq Y$$

where Y is a CW-space whose cells are in one-to-one correspondence with the critical cells of X.

The theorem is illustrated in Figure 16 and again in Figure 17. A difference between these two illustrations is that the first yields a deformation CW-retract, whereas the second yields a homotopically equivalent CW-space that is neither a subspace nor regular. We focus attention on retracts since we have efficient representations for these. (The cellular chain complex could be used to represent an arbitrary CW-space. However, this chain complex does not capture the homotopy

Figure 17: Illustration of Theorem 9.1

type of the space in general.) In particular, motivated by Theorem 9.1, we use discrete vector fields as a language for precisely describing deformation retracts. To this end we say that a pair (s,t) of cells in a regular CW-space X is *X-free* if

(i) $\dim(t) = \dim(s+1)$;

(ii) s lies in the boundary of t but in the boundary of no other cell of X. (In particular, s lies in the "boundary" of X.)

The following result is obvious and yields the subsequent algorithm.

Proposition 9.2. *Let X be a regular CW-space with an admissible discrete vector field whose critical cells form a CW-retract $X' \subset X$. Let (s,t) be a pair of cells in X' which is X'-free. Then adding the extra arrow $s \to t$ to the vector field on X results in a new admissible vector field whose critical cells form a CW-retract $X'' \subset X$.*

Algorithm 9.1.
Input: regular CW-space X.
Output: admissible discrete vector field on X whose critical cells form a retract $X' \subset X$.
Procedure:

initialize $X' := X$;
endow X with the trivial vector field involving no arrows;
while X' contains an X'-free pair of cells (s,t)
    let X'' be the retract of X' obtained by removing cells s and t;
    add the arrow $s \to t$ to the vector field on X;
    set $X' := X''$;

An implementation of Algorithm 9.1 was applied to the cubical lattice space X_{43} of Figure 11 and resulted in a 1-dimensional CW-retract $X' \subset X_{43}$ involving 720 1-dimensional cells and 720 0-dimensional cells. For the space X_{43} the implementation was an order of magnitude slower than our implementation of Algorithm 2.1. However, Algorithm 9.1 resulted in a lower-dimensional CW-retract, it yielded a corresponding discrete vector field on X_{43} and it did not require storage of a pre-computed list of contractible neighbourhoods of a cell. This comparison of Algorithms 2.1 and 9.1 also held when they were applied to various 3-dimensional cubical lattice spaces.

Algorithms 2.1 and 9.1 can be used together to good effect. Consider for instance the 3-dimensional permutahedral complex X obtained from a $100 \times 100 \times 100$ solid

array of permutahedra by removing a single central permutahedron. Then X has the homotopy type of a sphere S^2 and contains 999999 3-dimensional facets. Our implementation of Algorithm 2.1 took 5 seconds (on a 1GHz Linux laptop with 1GB of RAM) to produce a deformation retract $X' \subset X$ containing 14 facets, 160 2-dimensional cells, 336 edges and 192 vertices. Our implementation of Algorithm 9.1 then took 10 milliseconds to produce a CW-retract $X'' \subset X'$ containing 14 2-cells, 36 edges and 24 vertices. The integral homology of the chain complex $C_*(X'')$ is calculated in a few milliseconds.

The performance of Algorithms 2.1 and 9.1 is likely to depend heavily on the sophistication with which one iterates over cells $f \in X$ in loops of the form

while space X has a cell f with a given property, do ... .

Our implementations iterate in a fairly naive manner and could certainly be improved.

If one just wants to calculate homology, or a presentation for the fundamental group, of a regular CW-space X then it is desirable to compute an admissible discrete vector field on X which is as large as possible. This vector field can be used to construct a chain complex $C_*(Y)$ with one free generator for each critical cell in X and with homology isomorphic to the homology of X. There is no advantage in requiring that the critical cells form a retract. The following algorithm is used in the software package HAP for obtaining such a vector field.

Algorithm 9.2.
Input: regular CW-space X.
Output: admissible discrete vector field on X.
Procedure:

deem all cells in X to be unlabelled;
initialize $Y := X$;
while X contains unlabelled cells do

apply Algorithm 9.1 to obtain a discrete vector field on Y whose critical cells form a deformation retract $Y' \subset Y$;
transfer the vector field from Y to X and deem all cells of X involved in an arrow to be labelled;
if Y' is non-empty then remove some top dimensional cell from Y' and deem the corresponding cell in X to be labelled;
set $Y := Y'$;

10. Homology homomorphism of a zig-zag retraction

Consider the situation where we have formulae describing a strictly cellular map $\tau\colon X \to Y$ of large finite regular CW-spaces. Suppose that we have constructed zig-zag retracts X', Y' and wish to compute the homology homomorphism $H_n(\tau)\colon H_n(X') \to H_n(Y')$ induced by

$$X' \xrightarrow{\simeq} X \xrightarrow{f} Y \xrightarrow{\simeq} Y'.$$

The computation can be achieved using formulae that describe the chain homotopies $C_*(X') \to C_*(X)$ and $C_*(Y) \to C_*(Y')$. Formulae for these chain homotopies can be obtained as follows.

Suppose that X is endowed with an admissible discrete vector field whose critical cells form a deformation retract $X' \subset X$. The vector field determines a retraction $\rho\colon X \to X'$ which we shall describe at the level of the induced chain map $\rho_*\colon C_*(X) \to C_*(X')$. We regard $C_*(X')$ as a sub-chain complex of $C_*(X)$ and we denote the boundary homomorphisms of $C_*(X)$ by ∂_n. Given an arrow $s \to t$ in the vector field we say that the cell s is a *source* and that the cell t is a *target*; we let $\langle s, t\rangle$ denote the sign with which the chain group generator s appears in the boundary chain $\partial(t)$. The following result is a routine exercise.

Proposition 10.1. *Let X be a regular CW-space with admissible discrete vector field whose critical cells form a deformation retract $X' \subset X$. There exists a chain homomorphism $\rho\colon C_*(X) \to C_*(X')$ defined recursively on generators $f \in C_n(X)$ by*

$$\rho_n(f) = \begin{cases} f & \text{if } f \text{ is a critical cell} \\ 0 & \text{if } f \text{ is a target cell} \\ \rho_n(f - \langle f, t\rangle \partial_{n+1}(t)) & \text{if } f \text{ is a source cell with corresponding target } t. \end{cases}$$

The recursion terminates due to the admissibility of the vector field. The chain homomorphism induces isomorphisms on homology.

Any zig-zag retract of tessellated spaces

$$X = B_0 \hookleftarrow A_1 \hookrightarrow B_1 \hookleftarrow A_2 \hookrightarrow B_2 \hookleftarrow \cdots \hookrightarrow B_{k-1} \hookleftarrow A_k = X'$$

induces a sequence of chain homomorphisms

$$C_*(X) \xrightarrow{\rho_1} C_*(A_1) \xrightarrow{\iota_1} C_*(B_1) \xrightarrow{\rho_2} C_*(A_2) \to \cdots \to C_*(X')$$

where the ρ_i are induced by retractions and the ι_i are induced by inclusions. If each retract $A_i \subset B_{i-1}$ and $A_i \subset B_i$ is constructed from a discrete vector field then each of these chain maps can be computed and used to determine the homology homomorphisms $H_k(X;\mathbb{Z}) \to H_k(X';\mathbb{Z})$.

Proposition 10.1 is easily extended to provide the chain maps $C_*(X) \to C_*(Y)$, $C_*(Y) \to C_*(X)$ arising from an admissible discrete vector field on X with Y the homotopy equivalent space asserted by Theoprem 9.1. This provides an alternative approach to computing the persistence Betti numbers β_k^{ij} of a sequence of regular CW-spaces $X_1 \subset X_2 \subset X_3 \subset \cdots \subset X_n$. We could construct an admissible discrete vector field on each X_i. (For instance, one could construct a discrete vector field on X_n and induce from it vector fields on each X_i.) The vector fields induce chain homotopies $C_*(X_i) \to C_*(Y_i)$, $C_*(Y_i) \to C_*(X_i)$ from which we could compute the chain homomorphism $\overline{\iota}_{i,i+1}\colon C_*(Y_i) \to C_*(X_i) \to C_*(X_{i+1}) \to C_*(Y_{i+1})$. In this computation of $\overline{\iota}_{i,i+1}$ much of the information in the chain complexes $C_*(X_i)$ and $C_*(X_{i+1})$ is not required; memory and running time would thus be saved by

implementing these two chain complexes using a lazy evaluation programming style. The persistence Betti number β_k^{ij} is the rank of the image of the induced composite homology homomorphism $\iota_{i,j}\colon H_k(Y_i;\mathbb{Z}) \to H_k(Y_{i+1};\mathbb{Z}) \to \cdots \to H_k(Y_j;\mathbb{Z})$. We have not yet implemented this approach to persistence Betti numbers.

11. Geometry of data

Homological computations can be used to identify geometric features of data. Consider for instance the two 2-dimensional polygonal disks in Figure 18 (left). The first is a quadrilateral and the second has nine sides. Since both are homotopy

Figure 18: Two polygonal disks, their singularities, and thickened singularities.

equivalent to a point we can not distinguish them directly from their homological properties.

Let X be a tessellated space with finite tessellated subspace $Y \subset X$. (We are particularly interested in the case where X is the infinite tessellation $X = \mathbb{R}^n_L$.) By the *combinatorial complement* of Y we mean the tessellated subspace $X - Y$ consisting of the union of those facets of X that are not facets of Y. By the *combinatorial boundary* of Y we mean the tessellated subspace ∂Y consisting of the union of those facets f in Y whose neighbourhood $N_X(f)$ contains at least one facet e in $X - Y$. The *combinatorial interior* of Y is then the tessellated subspace $Y - \partial Y$. We write $\mathrm{size}(Y)$ to denote the number of facets in Y.

Given a facet f in X and integer $r \geqslant 1$ we define the *combinatorial ball* of radius r centred at f to be the tessellated subspace $B_X(r, f) = N_X(B_X(r-1, f))$ where $B_X(1, f) = N_X(f)$. We define the *sphere* of radius r centred at f to be the tessellated subspace $S_X(r, f) = \partial B_X(r, f)$.

The following notion of singular facet can be used to identify geometric features.

Definition 11.1. For an integer $r \geqslant 1$ and number $0 < \tau < 1$ we say that a facet f of Y is (r, τ)-smooth if either f lies in the combinatorial interior of Y or else the combinatorial complement $S_X(r, f) - \partial Y$ consists of precisely two contractible path-components C_1, C_2 satisfying

$$\frac{|\mathrm{size}(C_1) - \mathrm{size}(C_2)|}{\mathrm{size}(C_1) + \mathrm{size}(C_2)} < \tau.$$

We say that $f \in Y$ is (r, τ)-singular if it is not (r, τ)-smooth.

The two polygonal disks of Figure 18 were represented as 2-dimensional cubical lattice spaces with 45848 and 116825 facets respectively. For each polygonal disk

the cubical lattice subspace consisting of the $(10, 0.15)$-singular facets is shown in Figure 18 (middle). For the quadrilateral the space of singular facets has 4 path-components, and for the nine-sided polygon it has 11 path-components. Both spaces of singular facets were repeatedly thickened by taking neighbourhoods. One thickening is shown in Figure 18 (right). The β_0 persistence bar codes were constructed for the sequence of thickenings and show that there are respectively 4 and 9 persistent path components for the two polygons. This illustrates how homological methods can be used to distinguish between geometrically different tessellated spaces.

When using this method to identify geometric features of an object it seems likely that the permutahedral setting $\mathbb{R}^n_P$ should give better results than the cubical setting $\mathbb{R}^n_C$ since a permutahedral sphere of radius r is better than a cubical sphere as an approximation to an actual Euclidean sphere. However, we have not investigated this empirically.

12. Implementation

Implementations of the above algorithms are available in the first author's homological algebra package HAP [**8**] for the GAP compuational algebra system and in the second author's extension [**16**] to the package. The following three GAP sessions are intended to illustrate the modularity of the implementation, its user interface, and a few of its data types.

The first session reads the digital image of Figure 12 into HAP as a 2-dimensional cubical lattice complex S and then constructs the direct product $T = S \times S$ as a 4-dimensional cubical lattice complex. The complex T has the homotopy type of a torus and involves a total of 2452356 cells. Then T is converted into a regular CW-space Y with the same number of cells. A discrete vector field with just four critical cells is then constructed. A chain complex C of free abelian groups is then constructed. The chain complex involves one free generator for each critical cell and is chain homotopy equivalent to the cellular chain complex of Y. Finally the second homology $H_2(C) = \mathbb{Z}$ is computed. The homology computation uses GAP's implementation of a Smith Normal Form algorithm.

```
gap> S:=ReadImageAsPureCubicalComplex("annulus.eps",300);
Pure cubical complex of dimension 2.

gap> T:=DirectProductOfPureCubicalComplexes(S,S);
Pure cubical complex of dimension 4.

gap> Y:=CubicalComplexToRegularCWComplex(T);
Regular CW-space of dimension 4

gap> Size(Y);
2452356

gap> CriticalCellsOfRegularCWComplex(Y);
[ [ 2, 66910 ], [ 1, 52430 ], [ 1, 628618 ], [ 0, 173201 ] ]

gap> C:=ChainComplex(Y);
Chain complex of length 4 in characteristic 0 .

gap> Homology(C,2);
[ 0 ]
```

The second GAP session reads the hand-drawn image of Figure 19 into HAP as a 3-dimensional cubical lattice complex L. Each pixel is represented by a 3-cube. The image represents the linked Borromean rings and the first command of the session tries to identify the under crossings and build them into the 3-dimensional complex. The second command constructs the combinatorial complement K of L and the hope is that K has the homotopy type of the Euclidean complement of the Borromean rings. The combinatorial complement K is a cubical lattice complex

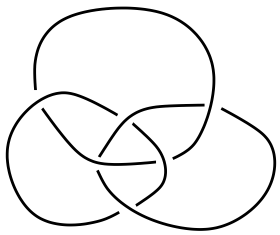


Figure 19: Hand-drawn image of Borromean rings

involving 1440266 3-dimensional cubes. A zig-zag homotopy retract of K is then constructed. This zig-zag rectract, M, is a 3-dimensional cubical lattice complex involving 116123 cells of dimension 3. The complex M is then converted to a regular CW-complex Y. Finally the fundamental group of Y is computed using the critical

cells of a discrete vector field. The fundamental group is stored as a finitely presented group.

```
gap> L:=ReadLinkImageAsPureCubicalComplex("borromean.jpg");
Pure cubical complex of dimension 3.

gap> K:=ComplementOfPureCubicalComplex(L);
Pure cubical complex of dimension 3.

gap> Size(K);
1440266

gap> M:=ZigZagContractedPureCubicalComplex(C);
Pure cubical complex of dimension 3.

gap> Size(M);
116123

gap> Y:=CubicalComplexToRegularCWSpace(M);
Regular CW-space of dimension 3

gap> CriticalCellsOfRegularCWComplex(Y);
[ [ 2, 334 ], [ 2, 115000 ], [ 2, 139630 ],
[ 1, 386713 ], [ 1, 404957 ], [ 1, 405056 ],
[ 1, 600331 ], [ 0, 164802 ], [ 0, 241782 ] ]
```

```
gap> G:=FundamentalGroup(Y);
#I  there are 3 generators and 2 relators of total length 20
<fp group of size infinity on the generators [ f1, f2, f3 ]>

gap> RelatorsOfFpGroup(G);
[ f3^-1*f2^-1*f3*f1^-1*f2*f3*f2^-1*f3^-1*f1*f2,
  f2^-1*f3^-1*f1*f3*f1^-1*f2*f1*f3^-1*f1^-1*f3 ]
```

The third GAP session below uses standard GAP and HAP commands to investigate the fundamental group G.

We first remark that permutahedral lattice complexes are a more efficient tool for accessing homotopical information of link complements. The reason has been mentioned above: permutahedral lattice spaces are manifolds and their combinatorial complement always has the homotopy type of their actual complement.

The third GAP session constructs the quotient $G_3 = G/\gamma_3(G)$ of the above fundamental group G by the 3rd term of its lower central series. The group G_3 is stored as

a power commutator presented group. A free $\mathbb{Z}G_3$-resolution is then constructed as the cellular chain complex $R_* = C_*(\tilde{X})$ of the universal cover of a classifying regular CW-space X. This universal cover contains infinitely many cells but can be stored on a computer using the G_3-action (see [**9**] for details). Finally the group homology $H_3(G/\gamma_3(G);\mathbb{Z}) \cong \mathbb{Z}^{40}$ is computed from the chain complex $T_* = C_*(X) \otimes_{\mathbb{Z}G_3} \mathbb{Z}$ using GAP's implementation of a Smith Normal Form algorithm.

```
gap> G3:=NilpotentQuotient(G,3);
Pcp-group with orders [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

gap> R:=ResolutionNilpotentGroup(G3,4);
Resolution of length 4 in characteristic 0 for Pcp-group with orders
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] .

gap> T:=TensorWithIntegers(R);
Chain complex of length 4 in characteristic 0 .

gap> Homology(T,3);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

We remark that $H_3(G/\gamma_3(G);\mathbb{Z})$ is a link invariant which can be used, for instance, to distinguish the Borromean rings L from a link L' obtained by changing one crossing in L.

We should also remark that this homological invariant of G would fail to distinguish the trefoil knot L from the trivial knot since $G/\gamma_n(G)$ is infinite cyclic for $n \geqslant 2$ when G is the fundamental group of the trefoil knot complement. However, an advantage of implementing software as a package for the GAP system is that one has easy access to a vast range of efficiently implemented algebraic algorithms. So had we taken L to be the trefoil knot then our knot group would have presentation $G = \langle x, y : x^2 = y^3 \rangle$. In order to distinguish this group G from the infinite cyclic group $G_0 = \langle x : \emptyset \rangle$ of the trivial knot we could use GAP's procedures for studying finitely presented groups. In particular, we could list the abelian invariants of all subgroups $H < G$ and of all subgroups $H < G_0$ of index at most 3. The following GAP session shows that $G \ncong G_0$ in this case.

```
gap> F:=FreeGroup(2);;x:=F.1;;y:=F.2;;
gap> G:=F/[x^2*y^-3];;
gap> G0:=F/[x*y^-1];;
gap> List(LowIndexSubgroupsFpGroup(G,3),AbelianInvariants);
[ [ 0 ], [ 0, 2, 2 ], [ 0, 0 ], [ 0, 3 ] ]
gap> List(LowIndexSubgroupsFpGroup(G0,3),AbelianInvariants);
[ [ 0 ], [ 0 ], [ 0 ] ]
```

References

[**1**] G. Carlsson, "Topology and data", *Bull. Amer. Math. Soc.* 46 (2009), 255–308.

[**2**] G. Carlsson, A. Zomorodian, A. Collins & L. Guibas, "Persistence barcodes for shapes", *International Journal of Shape Modeling*, 11 (2005), pp. 149–187.

[**3**] M. Cohen, *A course in simple homotopy theory*, Graduate Texts in Maths 10, Springer-Verlag, 1973.

[**4**] V. de Silva & R. Ghrist, "Homological sensor networks", *Notices Amer. Math. Soc.*, 54(1), 2007, 10–17.

[**5**] R. Ghrist, *Elementary Applied Topology*, in preparation, `http://hans.math.upenn.edu/ ghrist/notes.html`

[**6**] H. Edelsbrunner & J. Harer, "Persistent homology — a survey", In *Twenty Years After*, eds. J. E. Goodman, J. Pach and R. Pollack, AMS. (2007).

[**7**] H. Edelsbrunner & J. L. Harer, *Computational Topology. An Introduction*, Amer. Math. Soc., Providence, Rhode Island, 2010.

[**8**] G. Ellis, HAP – Homological Algebra programming, Version 1.9.4 (2011), a package for the GAP computational algebra system. (`http://www.gap-system.org/Packages/hap.html`)

[**9**] G. Ellis. Homological Algebra Programming, *Contemporary Mathematics*, 470 (2008), 63–74.

[**10**] V. Felsch, CRYSTCAT - Crystallographic Groups Catologue, a package for the GAP computer algebra system. (`http://www.math.uni-bielefeld.de/~gaehler/gap/CrystCat/`)

[**11**] R. Forman, "A user's guide to discrete Morse theory", In *Séminaire Lotharingien de Combinatoire*, volume 48, 2001.

[**12**] The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.4.9; 2006. (`http://www.gap-system.org`)

[**13**] GRAPHVIZ - Graph Visualization Software. `http://www.graphviz.org/`

[**14**] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, H. Wagner, M. Juda, P. Dlotko, "The Efficiency of a Homology Algorithm based on Discrete Morse Theory and Coreductions", published in: *Proceedings of the 3rd International Workshop on Computational Topology in Image Context, Chipiona, Spain, November 2010* (Rocio Gonzalez Diaz Pedro Real Jurado (Eds.)), Image A Vol. 1(2010), 41-47.

[**15**] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, "Discrete Morse Theoretic Algorithms for Computing Homology of Complexes and Maps", *Foundations of Computational Mathematics*, accepted.

[**16**] F. Hegarty, HAPPERMUTAHEDRAL – Version 1.0 (2011), a package for the GAP computational algebra system. (`http://hamilton.nuigalway.ie`)

[**17**] D. Jones, "A general theory of polyhedral sets", *Dissertationes Math.* (1988).

[**18**] T. Kaczyski, K.M. Mischaikow & Marian Mrozek k *Computational Homology*, Springer (2004), pp. 480.

[19] W.S. Massey, *A basic course in algebraic topology*, Graduate Texts in Mathematics 127, Springer-Verlag 1991.

[20] M. Mrozek *et al.*, Computer Assisted Proofs in Dynamics. `http://capd.ii.uj.edu.pl/`.

[21] M. Mrozek *et al.*, Reduction Homology Algorithms. `http://redhom.ii.uj.edu.pl/`.

[22] PLEX, software package for persistent homology of simplicial complexes, `http://comptop.stanford.edu/u/programs/jplex/`

[23] F. Sergereart et al., KENZO system for computational algebraic topology, `http://www-fourier-ujf-grenoble.fr/ sergerar/Kenzo`

[24] Tadeusz E. Dorozinski, Image of permutahedral tessellation `http://en.wikipedia.org/wiki/File:HC-A4.png`

[25] A. Zomorodian & G. Carlsson, "Computing persistent homology", *Discrete. Comput. Geom.* 33 (2005), 249–274.

Graham Ellis `graham.ellis@nuigalway.ie`

School of Mathematics, National University of Ireland Galway, Ireland

Fintan Hegarty `fintan.hegarty@gmail.com`

School of Mathematics, National University of Ireland Galway, Ireland