Homology, Homotopy and Applications, vol. 20(2), 2018, pp.41–59

DISTRIBUTED COMPUTATION OF LOW-DIMENSIONAL CUP PRODUCTS

NISREEN ALOKBI AND GRAHAM ELLIS

(*communicated by John Klein*)

Abstract

We describe a distributed algorithm for computing the cup product $\cup\colon H^1(X,\mathbb{Z})\times H^1(X,\mathbb{Z})\to H^2(X,\mathbb{Z})$ on the cohomology of a finite regular CW-space. A serial implementation of the algorithm is illustrated in two applied topological settings: (i) 3-dimensional digital images; (ii) topological data analysis of a finite sample of points from a metric space. For the second of these illustrations we introduce a cohomological enrichment of the Mapper clustering procedure which may be of independent interest.

1. Introduction

The additive structure of the low-dimensional homology and cohomology of a space is the mainstay of applied computational topology due to its ease of computation and interpretation. The goal of this paper is to advertize that the multiplicative structure on low-dimensional cohomology is also easy to compute and interpret. The paper builds on a practical algorithm for finding a finite presentation of the fundamental group $\pi_1(X,x_0)$ of an arbitrary finite regular CW-space X which was illustrated in [10] and described in detail in [1]. In Section 2 we explain how, from such a presentation, one can directly read off the cup product

$$\cup\colon H^1(X,\mathbb{Z})\times H^1(X,\mathbb{Z})\to H^2(X,\mathbb{Z}), \tag{1}$$

without need for any further significant computations since this product is essentially an invariant of $\pi_1(X,x_0)$. In Section 3 we illustrate the method on the integral cohomology ring of a 3-dimensional digital image. Previous papers [16, 18, 14, 15] have described different approaches to computing the cohomology ring, over $\mathbb{Z}/2\mathbb{Z}$, of cubical and simplicial spaces arising from 3-dimensional digital images; these papers are based on techniques in [28, 17, 24]. The fundamental group algorithm in [1] involves the construction of an admissible discrete vector field on X, and this construction can consume significant memory and time for large CW-spaces X. So in Section 4 we explain how the van Kampen theorem for fundamental groupoids [3, 7] yields a distributed version of the fundamental group algorithm of [1], and hence a distributed method for computing (1). In Section 5 we suggest how the distributed computation

Received May 8, 2017, revised December 16, 2017; published on April 18, 2018.
2010 Mathematics Subject Classification: 55N99.
Key words and phrases: cohomology, cup product, digital image, data analysis.
Article available at `http://dx.doi.org/10.4310/HHA.2018.v20.n2.a3`

of cup products could be used in conjunction with the Mapper clustering technique [29] from topological data analysis.

Acknowledgments

The first author was supported by a Government of Iraq postgraduate fellowship.

2. The low dimensional cup product

Let C_*X denote the cellular chain complex of a finite regular CW-space X, and let $H^k(X,\mathbb{Z}) = H^k(Hom_{\mathbb{Z}}(C_*X,\mathbb{Z}))$ denote the cellular cohomology group (see [21] for details). Recall that the cohomology ring structure is derived from the *diagonal map* $\Delta\colon X \to X\times X, x\mapsto (x,x)$ and induced *diagonal homomorphism*

$$H^k(\Delta,\mathbb{Z})\colon H^k(X\times X,\mathbb{Z}) \to H^k(X,\mathbb{Z}) \tag{2}$$

together with a bilinear *cross product* function

$$H^m(X,\mathbb{Z})\times H^n(X,\mathbb{Z}) \to H^{m+n}(X\times X,\mathbb{Z}), \tag{3}$$

for $k,m,n \geqslant 0$. The ring multiplication $H^*(X,\mathbb{Z})\times H^*(X,\mathbb{Z})\to H^*(X,\mathbb{Z})$ is obtained by composing (2) with (3) for $k=m+n$ and extending bilinearly over $H^*(X,\mathbb{Z})$. For an arbitrary regular CW-space the diagonal homomorphism (2) can be a challenge to implement efficiently on a computer. In contrast, an efficient implementation of (3) is straightforward. To implement (2) one can use the Alexander-Whitney diagonal approximation formula in the case of simplicial spaces, and Serre's analogue of this for cubical spaces. Details of the cubical analogue can be found in [26], and details on practical implementations of these two formulae can be found in [16, 18, 14, 15, 28, 17, 24]. In this section we assume that X is an arbitrary connected regular finite CW-space and observe that for $k=2$ the homomorphism (2), and hence the cup product (1), can be read directly from a group presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r}\rangle$ for the fundamental group $\pi_1 X$. One algorithm for computing such a presentation is described in [1]. It uses the notion of a *discrete vector field* on X, by which we mean 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;
2. any cell is involved in at most one arrow.

A cell in X is said to be *critical* if it is not involved in an arrow. 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 on a finite regular CW-space is said to be *admissible* if it contains no circuits. There are many accounts in the literature of algorithms for constructing admissible discrete vector fields (see for instance [10, 1, 20]). For connected X some of these algorithms ensure that precisely one of the 0-cells is critical. The following theorem can be viewed as a statement about simple homotopy equivalences phrased in the language of discrete vector fields. An informal explanation of this theorem is provided in [1, Example 2].

Theorem 2.1. *[30, 12, 11] If X is a regular CW-space with admissible discrete vector field then there is a homotopy equivalence*

$$X \simeq Y,$$

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

The presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r} \rangle$ for $\pi_1 X$ is produced by first constructing an admissible discrete vector on the 3-skeleton of X such that only one of the 0-cells is critical. By Theorem 2.1 there is then a homotopy equivalence $X \simeq Y$ where Y is a non-regular CW-space with a single 0-cell. The 2-skeleton of Y corresponds to the presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r} \rangle$. The generators in $\underline{x}$ correspond to the critical 1-cells of X and to the 1-cells of Y. The relators in $\underline{r}$ correspond to the critical 2-cells in X and to the 2-cells in Y. We shall denote the 2-skeleton Y^2 by $K(\mathcal{P})$ since this is precisely the 2-complex associated to the presentation $\mathcal{P}$ in geometric group theory.

On the computer we store Y by storing the face lattice of the regular CW-space X together with a list of those pairs of non-critical cells (s, t) that constitute the discrete vector field on X. This data can be used to realize a chain homotopy equivalence $h_*\colon C_*X \xrightarrow{\simeq} C_*Y$ on the computer; this in turn could be used to compute an isomorphism $H^k(X, \mathbb{Z}) \cong H^k(Y, \mathbb{Z})$ if desired. So we focus on computing the cup product $\cup\colon H^1(Y, \mathbb{Z}) \times H^1(Y, \mathbb{Z}) \to H^2(Y, \mathbb{Z})$.

The following notion, due to van Kampen and illustrated in Figure 1, is an aid to visualizing the diagonal homomorphism (2) for $k = 2$. Recall that a word in a free group is *cyclically reduced* if no conjugate of it is a shorter word, and that a presentation is *cyclically reduced* if all of its relators are. In this section we assume that presentations are cyclically reduced.

Definition 2.2. A *van Kampen diagram* over a cyclically reduced group presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r} \rangle$ is a finite, planar, connected and simply connected CW-space $\mathcal{D} \subset \mathbb{R}^2$ with each 1-cell labelled by an arrow and a generator in $\underline{x}$ in such a way that: for each 2-cell e^2 in $\mathcal{D}$, the sequence of oriented and labelled 1-cells in the boundary ∂e^2, for some choice of initial vertex and some choice of orientation, spells a relator word in $\underline{r} \subset F(\underline{x})$.

It is not difficult to see that for any initial vertex in the boundary of a van Kampen diagram $\mathcal{D}$ over $\mathcal{P} = \langle \underline{x} \mid \underline{r} \rangle$, and for either orientation on the boundary of the diagram, the labelled 1-cells of the boundary spell a word in $R = \langle \underline{r} \rangle^{F(\underline{x})}$ where R is the normal closure in $F(\underline{x})$ of the subgroup generated by $\underline{r}$. The converse is also true: any non-trivial word in R is the boundary of a van Kampen diagram over $\mathcal{P}$. We are interested in a very easy and very particular case of this converse statement.

Given a presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r} \rangle$ of a group G, let $\underline{x}' = \{x' : x \in \underline{x}\}$ be an isomorphic copy of the set $\underline{x}$. Let $F(\underline{x} \cup \underline{x}')$ be the free group on the disjoint union $\underline{x} \cup \underline{x}'$ and let $\iota_1, \iota_2\colon F(\underline{x}) \hookrightarrow F(\underline{x} \cup \underline{x}')$ be the inclusion homomorphisms defined by $\iota_1(x) = x$ and $\iota_2(x) = x'$ for $x \in \underline{x}$. We define the presentation

$$\mathcal{P} \times \mathcal{P} = \langle \underline{x} \cup \underline{x}' \mid \iota_1(r), \iota_2(r), \iota_1(x)\iota_2(y) = \iota_2(y)\iota_1(x) \quad (x, y \in \underline{x}, r \in \underline{r}) \rangle,$$

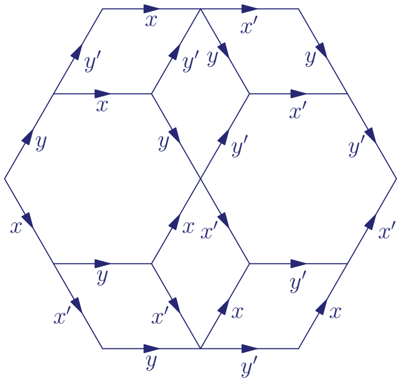


Figure 1: van Kampen diagram over the presentation $\langle x, y, x', y' \mid xyx = yxy, x'y'x' = y'x'y', xx' = x'x, yy' = y'y, xy' = y'x, yx' = x'y\rangle$.

for the direct product $G \times G$. In this context we define the *diagonal* group homomorphism $\Delta\colon F(\underline{x}) \to F(\underline{x} \cup \underline{x}')$ by $\Delta(x) = \iota_1(x)\iota_2(x)$ for $x \in \underline{x}$.

The following lemma is an easy exercise. In its statement we use the notation $i <^{\epsilon} j$ to mean $i < j$ if $\epsilon = +1$ and $i \geqslant j$ if $\epsilon = -1$.

Lemma 2.3. *For any presentation $\mathcal{P} = \langle \underline{x} \mid \underline{r}\rangle$ and any reduced word $w = x_1^{\epsilon_1}x_2^{\epsilon_2}\ldots x_n^{\epsilon_n} \in R = \langle \underline{r}\rangle^{F(\underline{x})}$, where $\epsilon_i = \pm 1$ and $x_i \in \underline{x}$, there is a van Kampen diagram $\mathcal{D}$ over $\mathcal{P} \times \mathcal{P}$ whose boundary spells the word $\Delta(w)$ for some choice of initial vertex and orientation. The diagram $\mathcal{D}$ consists of precisely one 2-cell e^2_w with boundary word $\iota_1(w)$, one 2-cell $e^2_{w'}$ with boundary word $\iota_2(w)$ and, for each $1 \leqslant i <^{\epsilon_i} j \leqslant n$, one 2-cell e^2_{ij} whose boundary spells the relator $\iota_1(x_i)\iota_2(x_j)\iota_1(x_i)^{-1}\iota_2(x_j)^{-1}$.*

The van Kampen diagram $\mathcal{D}$ of Lemma 2.3 is typically *non-reduced* in the sense that there may be two CW-subspaces $\mathcal{D}_1, \mathcal{D}_2 \subset \mathcal{D}$ containing no common 2-cell whose union $\mathcal{D}_1 \cup \mathcal{D}_2$ is connected, simply-connected and has boundary that spells a word representing the trivial element in the free group of the presentation $\mathcal{P}$. A van Kampen diagram is said to be *reduced* if it contains no such subspaces $\mathcal{D}_1, \mathcal{D}_2$. It is possible to transform any van Kampen diagram $\mathcal{D}$ into a reduced diagram $\mathcal{D}'$ such that the boundary words of $\mathcal{D}$ and $\mathcal{D}'$ represent the same element in the free group of $\mathcal{P}$. The details of this transformation are not required for our cup product algorithm.

Example 2.4. Consider the presentation $\mathcal{P} = \langle x, y \mid xyxy^{-1}x^{-1}y^{-1}\rangle$ and word $w = xyxy^{-1}x^{-1}y^{-1} \in F(\{x, y\})$. A reduced van Kampen diagram over $\mathcal{P} \times \mathcal{P}$ with boundary word $\Delta(w)$ is shown in Figure 1.

Lemma 2.3 describes a cellular diagonal approximation $\Delta\colon K(\mathcal{P}) \to K(\mathcal{P}) \times K(\mathcal{P})$ which, in turn, induces a diagonal chain approximation

$$\Delta_*\colon C_*(K(\mathcal{P})) \to C_*(K(\mathcal{P} \times \mathcal{P})) \cong C_*(K(\mathcal{P})) \otimes C_*(K(\mathcal{P})). \tag{4}$$

Using e^0, e^1_x, e^2_r ($x \in \underline{x}, r \in \underline{r}$) to denote free abelian chain group generators, the chain map (4) is given in degrees $0, 1$ by

$$\Delta_0(e^0) = e^0,$$
$$\Delta_1(e^1_x) = e^1_x + e^1_{x'}.$$

In degree 2 it is defined for each $r = x_1^{\epsilon_1} x_2^{\epsilon_2} \ldots x_n^{\epsilon_n} \in \underline{r}$ by

$$\Delta_2(e^2_r) = e^2_r + e^2_{r'} + \sum_{1 \leqslant i < j \leqslant n} e^2_{ij}. \tag{5}$$

The cup product on $H^*(K(\mathcal{P}), \mathbb{Z})$ is determined by the chain map (4) which, in turn, can be read off directly from the presentation $\mathcal{P}$ using Lemma 2.3. We emphasize that one can apply this Lemma algebraically – there is no need to exhibit an actual van Kampen diagram in order to apply formula (5).

The inclusion $K(\mathcal{P}) \hookrightarrow Y$ and homotopy equivalence $Y \overset{\simeq}{\to} X$ induce a ring homomorphism $H^*(X, \mathbb{Z}) \to H^*(K(\mathcal{P}), \mathbb{Z})$ which restricts to an isomorphism $H^k(X, \mathbb{Z}) \xrightarrow{\cong} H^k(K(\mathcal{P}), \mathbb{Z})$ for $k = 0, 1$ and inclusion $H^2(X, \mathbb{Z}) \hookrightarrow H^2(K(\mathcal{P}), \mathbb{Z})$. We can thus compute the cup product (1) by using Lemma 2.3 to compute the cup product in the bottom row of the following commutative diagram.

$$\begin{tikzcd}
H^1(\pi_1 X, \mathbb{Z}) \times H^1(\pi_1 X, \mathbb{Z}) \arrow[r, "\cup"] \arrow[d, "\cong"] & H^2(\pi_1 X, \mathbb{Z}) \arrow[d, hook] \\
H^1(X, \mathbb{Z}) \times H^1(X, \mathbb{Z}) \arrow[r, "\cup"] \arrow[d, "\cong"] & H^2(X, \mathbb{Z}) \arrow[d, hook] \\
H^1(K(\mathcal{P}), \mathbb{Z}) \times H^1(K(\mathcal{P}), \mathbb{Z}) \arrow[r, "\cup"] & H^2(K(\mathcal{P}), \mathbb{Z})
\end{tikzcd}$$

By attaching cells to X in dimensions $\geqslant 3$ we can construct a cellular inclusion $X \hookrightarrow B(\pi_1 X)$ into a space $B(\pi_1 X)$ with trivial homotopy groups $\pi_n(B(\pi_1 X)) = 0$ for $n \geqslant 2$ and with the inclusion inducing an isomorphism $\pi_1 X \cong \pi_1(B(\pi_1 X))$. The induced ring homomorphism $H^*(\pi_1 X, \mathbb{Z}) = H^*(B(\pi_1 X), \mathbb{Z}) \to H^*(X, \mathbb{Z})$ shows that the cup product (1) is induced from the cohomology of the group $\pi_1 X$ and hence can be calculated from *any* presentation $\mathcal{P}$ for $\pi_1 X$.

3. Illustration: digital images

Let T be the free abelian group of rank n with a specified free action on $\mathbb{R}^n$ as translations. Thus the elements $t \in T$ are distinct translations $\mathbb{R}^n \to \mathbb{R}^n, x \mapsto t(x)$. A fundamental domain for this action is

$$V = \{x \in \mathbb{R}^n \ : \ \|x\| \leqslant \|x - t(0)\| \text{ for all } t \in T\},$$

where $\| \ \|$ is the Euclidean metric and 0 denotes the zero vector in $\mathbb{R}^n$. This domain V is an intersection of finitely many half-spaces or, equivalently, the convex hull of a finite collection of vertices. We define a *digital image* to consist of a subset $S \subset T$. For each $t \in S$ we say that $tV = \{t(x) \ : \ x \in V\}$ is a *voxel* of the digital image S. We define the *geometric realization* of S to be the union of the voxels of S.

If the elements of a free generating set for T act on $\mathbb{R}^n$ as the unit translations along the standard axes then V is an n-dimensional cube. The geometric realization of a digital image in this case is called a *pure cubical complex*. But there are other useful choices of action. If we identify $\mathbb{R}^n$ with the hyperplane $\mathbb{R}^n = \{(x_1, \ldots, x_{n+1}) \in \mathbb{R}^{n+1} \ : \ x_1 + \cdots + x_{n+1} = 0\}$ and take the free generators of T to act as $x \mapsto x + v_i$ for $v_1 = (-n, 1, 1, \ldots, 1, 1), v_2 = (1, -n, 1, \ldots, 1, 1), \ldots, v_n = (1, 1, 1, \ldots, -n, 1)$ then V is the n-dimensional permutahedron. In this case the geometric realization of a digital image is called a *pure permutahedral complex*. An advantage to permutahedral complexes is that their boundary is a manifold; this is not necessarily so for pure cubical complexes. The term *lattice space* is used in [10] to refer to the geometric realization of an arbitrary digital image.

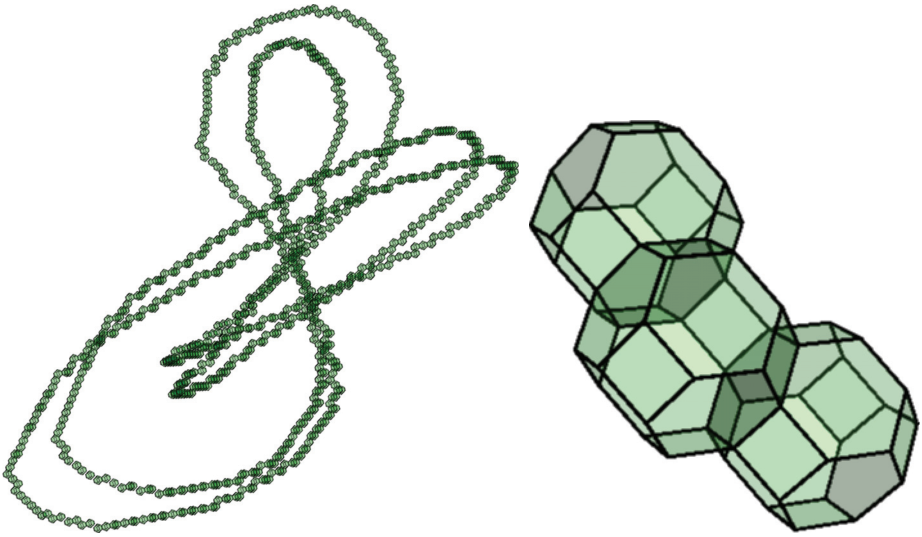


Figure 2: Pure permutahedral complex L representing a link with two components (left), an enlarged segment of which is also shown (right).

Figure 2 shows a pure permutahedral complex L representing a link with two components, one component winding around the other. To investigate the link we embed it into the interior of a contractible pure permutahedral complex R and form the complement $M = R \setminus \mathring{L}$ of the interior of L. The following GAP session loads such a complex M from the file `permutahedralcomplex.txt` available at [9]. The pure permutahedral complex M has 257851 voxels. As a CW-space it has 6982671 cells. The session first constructs a smaller homotopy equivalent pure permutahedral complex $XM \simeq M$ with just 3728 voxels and, as a CW-space, 156127 cells. The space XM is constructed using a *zig-zag deformation retract* technique based on simple homotopy collapses which is described explicitly in [10]. The session then uses XM to compute

$$H^n(M, \mathbb{Z}) = \begin{cases} \mathbb{Z}, & n = 0, \\ \mathbb{Z} \oplus \mathbb{Z}, & n = 1, 2, \\ 0, & n \geqslant 3, \end{cases}$$

and a presentation for $G = \pi_1 M$ using the algorithm of [1].

```
gap> Read("permutahedralcomplex.txt");
gap> Size(M); # number of 3-d voxels
257851
gap> XM:=RegularCWComplex(ZigZagContractedComplex(M));
Regular CW-complex of dimension 3
gap> Size(XM); # number of CW cells
156127
gap> List([0..3],n->Cohomology(XM,n));
[ [ 0 ], [ 0, 0 ], [ 0, 0 ], [  ] ]
gap> G:=FundamentalGroup(XM);
<fp group of size infinity on the generators [ f1, f2, f3 ]>
```

The following continuation of the GAP session uses the method described in Section 2 to compute the cup product $\alpha \cup \beta = 6(-\gamma + \delta)$ where α, β are free generators of $H^1(M, \mathbb{Z})$ and γ, δ are free generators of $H^2(M, \mathbb{Z})$.

```
gap> cup:=CupProduct(G);
function( a, b ) ... end
gap> cup([1,0],[0,1]);
[ -6, 6 ]
```

It is well known that the cup product $\alpha \cup \beta$ can be interpreted in terms of the linking number $Lk(K_1, K_2)$ where K_i are the two components in the link of Figure 2 (see for instance [**27**]).

We mention that to any lattice space M one can associate a homotopy equivalent simplicial complex SM simply by taking the nerve of its collection of voxels; we let XSM denote SM regarded as a CW-space. The following continuation of the GAP session shows that for our example above this construction leads to a smaller CW-space than the one above involving permutahedral cells; the CW-space XSM has 31013 cells compared to the 156127 cells of XM. So it would have been slightly advantageous to work with XSM rather than XM.

```
gap> Size(XM);
156127
gap> XSM:=RegularCWComplex(Nerve(ZigZagContractedComplex(M)));
Regular CW-complex of dimension 3
gap> Size(XSM);
31013
```

Indeed, one could have reduced the size of the problem further by applying a sequence of simple homotopy collapses to obtain deformation retracts $XM1 \subseteq XM$ and $XSM1 \subseteq XSM$. The following commands

```
gap> XM1:=ContractedComplex(XM);;
gap> Size(XM1);
78341
gap> XSM1:=ContractedComplex(XSM);;
gap> Size(XSM1);
25185
```

produce CW-spaces $XM1$, $XSM1$ with 78341 cells and 25185 cells respectively.

The simplicial complex SM provides a method for computing cup products in all dimensions when dealing with higher-dimensional M: one can use the Alexander-Whitney diagonal approximation on the cellular chain complex $C_*(SM)$. For practical computations one would construct an admissible discrete vector field on SM to produce a homotopy equivalence to a smaller non-regular CW-space Y and incorporate the induced chain homotopy equivalences $C_*(SM) \overset{\simeq}{\rightleftarrows} C_*Y$ into the computation. For the low-dimensional cup product (1) this approach tends to be slower than that described above. The following continuation of the GAP session computes, for our 3-dimensional example, a chain complex C_*Y with $C_0Y = \mathbb{Z}$, $C_1Y = \mathbb{Z}^3$, $C_2Y = \mathbb{Z}^3$, $C_nY = 0$ for $n \geqslant 3$ and so it is certainly quite practical to work with the chain equivalences $C_*(SM) \rightleftarrows C_*Y$ in this example.

```
gap> CY:=ChainComplex(XSM);
Chain complex of length 3 in characteristic 0 .
gap> List([0..3],CY!.dimension);
[ 1, 3, 3, 0 ]
```

4. Fundamental groupoids and a distributed algorithm

Let us analyse the computation of the previous section to see how it might be distributed over several computers. The computation begins with the representation of the link L as a pure permutahedral complex consisting of 864 voxels. The complement $M = R \setminus \mathring{L}$ is then constructed as a pure permutahedral complex with 257851 voxels. The voxels of M are stored as a list of points in $\mathbb{R}^3$. There are then three steps to computing the cohomology cup product on M, the first and second of which consume the most time and memory.

1. A zig-zag homotopy contraction (details of which are given in [10]) is performed to compute a pure permutahedral complex M' which is homotopy equivalent to M and which involves only 3728 voxels.
2. The complex M' is realized as a regular CW-space $X = XM$ involving 156127 cells in dimensions $\leqslant 3$. A maximal admissible discrete vector field is computed on X and the critical 1-cells and 2-cells are used to construct a presentation $\mathcal{P}$ for $\pi_1 X \cong \pi_1 M$.
3. The cup product $\alpha \cup \beta$ is read off directly from $\mathcal{P}$.

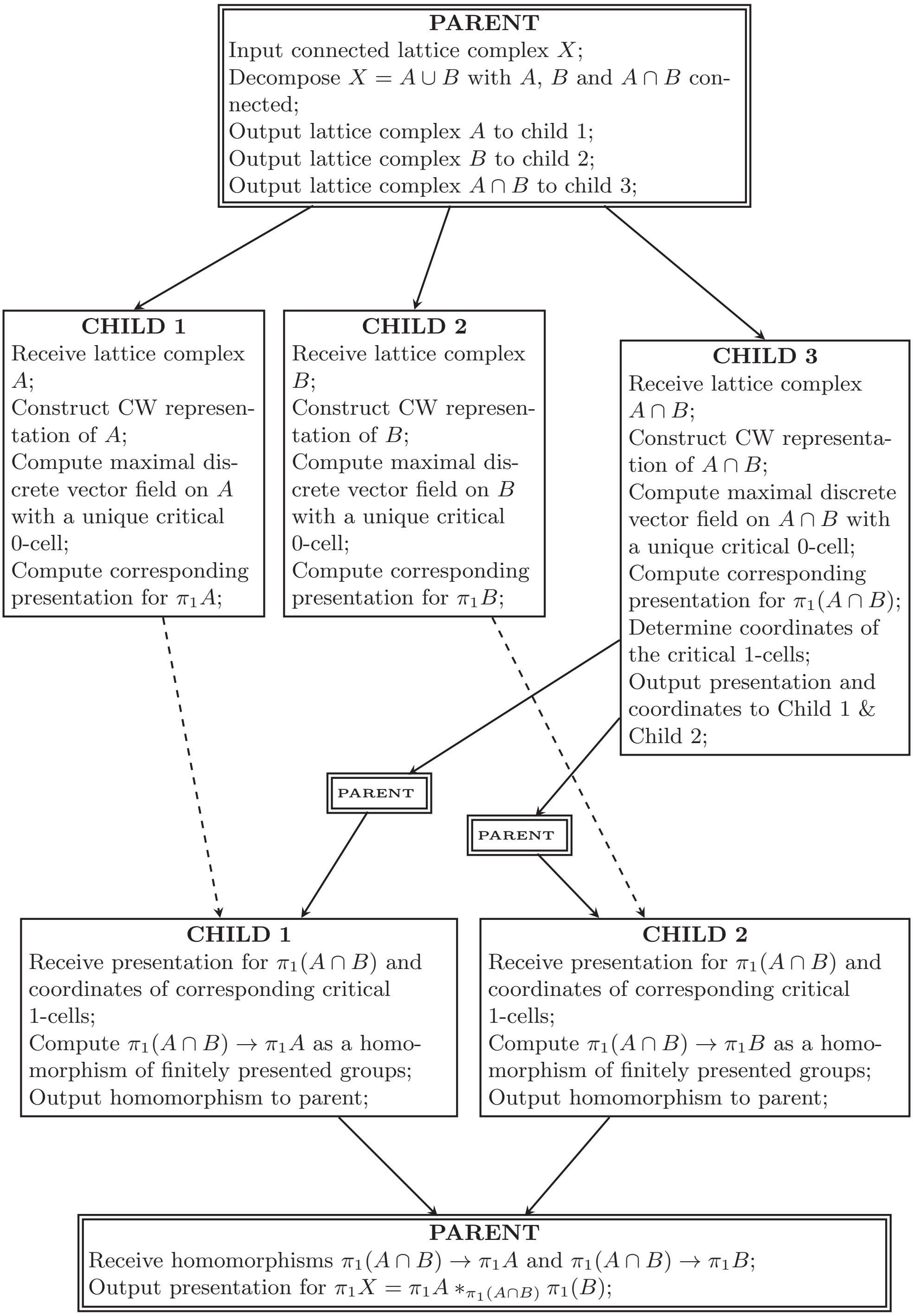
It is difficult to make meaningfully precise statements about the complexity of Steps 1–3. Step 2 involves the computation of a discrete vector field on a regular CW-space X. One could always deem every cell of X to be critical, in which case there is nothing to compute. At the other extreme one could aim for a vector field involving a minimal number of critical cells – but the construction of such a vector field is known to be an NP hard problem (see for instance [25]). In Step 1 the zig-zag homotopy reduction procedure is based on repeated applications of procedures which can be regarded as producing discrete vector fields on a lattice complex and so the comments for Step 2 also apply to Step 1. In Step 3 the 1- and 2-dimensional cohomology groups of a cochain complex need to be computed using some version of a Smith Normal Form algorithm and here again it is possible to produce examples where small input data can yield lengthy computations. Rather than attempt to make any precise complexity statement we simply provide the running times (on a Linux laptop with Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz) for each of the three steps applied to the example of Section 3: Step 1 took 5.7 seconds; Step 2 took 5.4 seconds; Step 3 took 0.004 seconds.

Step 1 involves repeatedly applying a basic homotopy deformation procedure which inputs a pure lattice complex M and outputs a homotopy equivalent pure lattice complex M' with potentially fewer voxels. The basic deformation procedure is applied until no size reduction occurs. The initial application of the procedure takes 1.1 seconds to reduce the number of voxels from 257851 to 25719 – a 90 percent reduction in size. The basic procedure has to be applied a total of six times in order to achieve a complex with 3728 voxels – a 98 percent reduction in size. So in this example Step 1 is best speeded up simply by applying the basic procedure only once rather than six times.

We thus focus on Step 2, with the aim of performing a distributed computation of a presentation $\mathcal{P}$ for $\pi_1 X$.

We start by supposing that in Step 2 we have some expression for $M' = A \cup B$ as a union of two pure permutahedral complexes A, B whose intersection $A \cap B$ is also a pure permutahedral complex. (For instance, we could choose two suitable integers $l < r \in \mathbb{R}$ and let A consist of those voxels of M' whose centre has x-coordinate $< r$, and B consist of those voxels whose centre has x-coordinate $> l$.) For the moment let us make an assumption which is mathematically convenient but which is not so realistic for many applied settings: assume that the spaces A, B and $A \cap B$ are all connected.

Given three computers PC_1, PC_2, PC_3 each directly connected to a parent computer we can send, from the parent computer, the data describing the pure permutahedral complex A to PC_1, the data describing B to PC_2, and the data describing $A \cap B$ to PC_3. We can then simultaneously construct the faces lattices for A, B, $A \cap B$ and represent these spaces as regular CW-spaces $X_A, X_B, X_{A\cap B}$ on PC_1, PC_2, PC_3 respectively. Then we can simultaneously compute maximal discrete vector fields on the 3-skeleta of X_A and X_B, and on the 2-skeleton of $X_{A\cap B}$. Because of the connectivity assumptions the computation can be done so that in each case only one 0-cell is critical. We can then read off on PC_1 the presentation $\mathcal{P}_A = \langle \underline{x}_A \mid \underline{r}_A \rangle$ for $\pi_1(X_A)$, and read off on PC_2 the presentation $\mathcal{P}_B = \langle \underline{x}_B \mid \underline{r}_B \rangle$ for $\pi_1(X_B)$. On PC_3 we have that the critical 1-cells of $X_{A\cap B}$ correspond to a generating set $\underline{x}_{A\cap B}$ for $\pi_1(X_{A\cap B})$. The identities of these critical 1-cells on $X_{A\cap B}$ can be sent from PC_3 to both PC_1 and PC_2. Then, on PC_1 we can use the discrete vector field on X_A to compute the homomorphism of free groups $f_A \colon F(\underline{x}_{A\cap B}) \to F(\underline{x}_A)$ induced by the inclusion $X_{A\cap B} \hookrightarrow X_A$ while, simultaneously on PC_2, we can use the discrete vector field on X_B to compute the corresponding homomorphism $f_B \colon F(\underline{x}_{A\cap B}) \to F(\underline{x}_B)$. Here $F(\underline{x}_A)$ denotes the free group on $\underline{x}_A$. The presentations for $\pi_1 A$, $\pi_1 B$, $\pi_1(A \cap B)$ and the homomorphisms f_A, f_B can now all be sent to the parent. The presentation $\mathcal{P}_{A\cup B} = \langle \underline{x}_A, \underline{x}_B \mid \underline{r}_A, \underline{r}_B, f_A(x) = f_B(x)\ (x \in \underline{x}_{A\cap B}) \rangle$ for the amalgamated product $\pi_1 A *_{\pi_1(A\cap B)} \pi_1 B$ can now be computed on the parent. By van Kampen's theorem $\mathcal{P}_{A\cup B}$ is a presentation for the fundamental group $\pi_1 X = \pi_1(A \cup B)$. The following is a flow chart of the parallel computation.



PARENT
Input connected lattice complex X;
Decompose $X = A \cup B$ with A, B and $A \cap B$ connected;
Output lattice complex A to child 1;
Output lattice complex B to child 2;
Output lattice complex $A \cap B$ to child 3;

CHILD 1
Receive lattice complex A;
Construct CW representation of A;
Compute maximal discrete vector field on A with a unique critical 0-cell;
Compute corresponding presentation for $\pi_1 A$;

CHILD 2
Receive lattice complex B;
Construct CW representation of B;
Compute maximal discrete vector field on B with a unique critical 0-cell;
Compute corresponding presentation for $\pi_1 B$;

CHILD 3
Receive lattice complex $A \cap B$;
Construct CW representation of $A \cap B$;
Compute maximal discrete vector field on $A \cap B$ with a unique critical 0-cell;
Compute corresponding presentation for $\pi_1(A \cap B)$;
Determine coordinates of the critical 1-cells;
Output presentation and coordinates to Child 1 & Child 2;

CHILD 1
Receive presentation for $\pi_1(A \cap B)$ and coordinates of corresponding critical 1-cells;
Compute $\pi_1(A \cap B) \to \pi_1 A$ as a homomorphism of finitely presented groups;
Output homomorphism to parent;

CHILD 2
Receive presentation for $\pi_1(A \cap B)$ and coordinates of corresponding critical 1-cells;
Compute $\pi_1(A \cap B) \to \pi_1 B$ as a homomorphism of finitely presented groups;
Output homomorphism to parent;

PARENT
Receive homomorphisms $\pi_1(A \cap B) \to \pi_1 A$ and $\pi_1(A \cap B) \to \pi_1 B$;
Output presentation for $\pi_1 X = \pi_1 A *_{\pi_1(A\cap B)} \pi_1(B)$;

This distributed computation of a presentation for $\pi_1 X$ is correct because the classical theorem of van Kampen can be invoked thanks to the connectivity assumptions on $A, B, A \cap B$. The distributed computation is practical because: (i) the digital images $A, B, A \cup B$ can be efficiently transmitted between computers simply by transmitting the lists of their voxel centres (there is no need to transmit the CW face lattice); (ii) the homotopy-theoretic information described by the critical cells of the CW-spaces $X_A, X_B, X_{A\cap B}$ in dimensions $\leqslant 2$ can be encoded as group presentations and then efficiently transmitted between computers. The method, under the connectivity assumptions, is implemented in the current version of HAP [8].

But one only has to think of a digital image representing a torus in $\mathbb{R}^3$ to realize that many of the most natural decompositions $M = A \cup B$ of a digital image M will have disconnected intersection $A \cap B$. To overcome this difficulty we pay attention to Alexander Grothendieck [19]:

> Ceci est lié notamment au fait que les gens s'obstinent encore, en calculant avec des groupes fondamentaux, à fixer un seul point base, plutôt que d'en choisir astucieusement tout un paquet qui soit invariant par les symétries de la situation, lesquelles sont donc perdues en route. Dans certaines situations (comme des théorèmes de descente à la Van Kampen pour groupes fondamentaux) il est bien plus élégant, voire indispensable pour y comprendre quelque chose, de travailler avec des groupoïdes fondamentaux par rapport à un paquet de points base convenable ...

Recall that a *groupoid* is a category $\mathcal{G}$ in which, for each morphism $f\colon a \to b$, there is a morphism $f^{-1}\colon b \to a$ such that $f \circ f^{-1} = 1_b$, $f^{-1} \circ f = 1_a$. A *morphism* of groupoids is just a functor. A groupoid whose set $Ob(\mathcal{G})$ of objects consists of a single object a is precisely a group; a morphism between single object groupoids is precisely a group homomorphism.

Let X be a space with subset $X_0 \subset X$ of *basepoints*. The *fundamental groupoid* $\pi_1(X, X_0)$ consists of homotopy classes rel X_0 of maps $p\colon [0,1] \to X$ with $p(0), p(1) \in X_0$. We write $[p]$ for the homotopy class rel X_0 of a map p. The composite $p * q$ of two such paths $p, q\colon [0,1] \to X$ is defined if $p(1) = q(0)$, and the composite is given by $p * q(t) = p(2t)$ for $t \in [0, 1/2]$ and $p * q(t) = q(2t - 1)$ for $t \in [1/2, 1]$. Groupoid composition is given by $[p] * [q] = [p * q]$. Given spaces X, X' with basepoint sets X_0, X_0' respectively, and given a map $f\colon X \to X'$ that sends X_0 into X_0', there is an *induced morphism* of groupoids $\pi_1 f\colon \pi_1(X, X_0) \to \pi_1(X', X_0'), [p] \mapsto [fp]$.

The following version of the van Kampen theorem is due to Ronnie Brown [5] and is a special case of a more general version in [4] for unions of more than two spaces.

Theorem 4.1. *[5, 4] Let X be a topological space with subspaces X_1, X_2. Let $\mathring{X}_i$ denote the interior of X_i. Let X_0 be a set of points in X that has non-empty intersection with each connected component of $X_1 \cap X_2$, each connected component of X_1, and each connected component of X_2. Then the induced square of fundamental groupoids*

$$\begin{tikzcd}
\pi_1(X_1 \cap X_2, X_1 \cap X_2 \cap X_0) \arrow[r] \arrow[d] & \pi_1(X_2, X_2 \cap X_0) \arrow[d] \\
\pi_1(X_1, X_1 \cap X_0) \arrow[r] & \pi_1(X, X_0)
\end{tikzcd}$$

is a pushout in the category of groupoids.

The basic definitions and constructions on finite presentations of groups extend routinely to finite presentations of groupoids (see [22, 23, 6] for details). A key observation for our computational goal is that if X is a regular CW-space endowed with a discrete vector field, and if X_0 is its set of critical 0-cells, then the collection of critical 0-cells, critical 1-cells and critical 2-cells determine a presentation of the fundamental groupoid $\pi_1(X, X_0)$ in a way that routinely generalizes the description in [1] for fundamental groups. Thus the above description of a distributed computation of a presentation for the fundamental group $\pi_1(X, x_0)$ of a regular CW-space $X = X_A \cup X_B$, under severe connectivity assumptions, extends to a computation of a presentation for the fundamental groupoid $\pi_1(X, X_0)$ where the connectivity assumptions are relaxed to the requirement that the set X_0 has non-empty intersection with each connected component of $X_{A\cap B}$, each connected component of X_A, and each connected component of X_B.

If a regular CW-space X is connected then there is a simple procedure, given in [22], for extracting a presentation $\mathcal{P}$ for $\pi_1(X, x_0)$ from a presentation for $\pi_1(X, X_0)$. This provides a practical method for distributing the computation of $\mathcal{P}$ over several computers. The first author, as part of her PhD work, is implementing a package of GAP functions for handling finitely presented groupoids and for performing this distributed computation of $\mathcal{P}$.

The above distributed computation extends to regular CW-spaces $X = X_1 \cup X_2 \cup \cdots \cup X_n$ arising as the union of CW-subspaces X_i for $1 \leqslant i \leqslant n$. Let X_0 denote a subset of the 0-skeleton X^0 that has non-empty intersection with each connected component of $X_i \cap X_j$ and non-empty intersection with each connected component of $X_i \cap X_j \cap X_k$ for all $1 \leqslant i, j, k \leqslant n$. There is a diagram of groupoid morphisms

$$\bigsqcup_{1\leqslant i<j\leqslant n} \pi_1(X_i \cap X_j, X_0 \cap X_i \cap X_j) \underset{\beta}{\overset{\alpha}{\rightrightarrows}} \bigsqcup_{1\leqslant i\leqslant n} \pi_1(X_i, X_0 \cap X_i) \xrightarrow{\gamma} \pi_1(X, X_0), \qquad (6)$$

where $\bigsqcup$ denotes disjoint union in the category of groupoids; α is induced by the inclusion $X_i \cap X_j \hookrightarrow X_i$; β is induced by the inclusion $X_i \cap X_j \hookrightarrow X_j$; γ is induced by the inclusion $X_i \hookrightarrow X$. The generalization of van Kampen's theorem given in [7] implies that in (6) the morphism γ is the coequalizer of α and β in the category of groupoids. Thus to compute a presentation for $\pi_1(X, X_0)$ one needs only compute presentations for each $\pi_1(X_i, X_0 \cap X_i)$ and the images under α and β of generators for each $\pi_1(X_i \cap X_j, X_0 \cap X_i \cap X_j)$. One can achieve this by first choosing a set X_0 of vertices that has non-empty intersection with each connected component of all double and triple intersections. Then compute admissible discrete vectors fields on the 2-skeleta of $X_i \cap X_j$ for all $1 \leqslant i < j \leqslant n$ such that all vertices in $X_0 \cap X_i \cap X_j$ are critical. Also compute admissible discrete vector fields on the 3-skeleta of the X_i such that all vertices in $X_0 \cap X_i$ are critical. From the vector fields on the $X_i \cap X_j$ and X_i one can construct the coequalizer (6) as a diagram of finitely presented groupoids and thus obtain a presentation for $\pi_1(X, X_0)$.

It is instructive to give an example showing the necessity for X_0 to have non-empty intersection with each connected component of three-fold intersections. Let X be the simplicial graph with vertex set $X^0 = \{1, 2, 3, 4, 5\}$ and edges $\{1, 4\}$, $\{2, 4\}$, $\{3, 4\}$, $\{1, 5\}$, $\{2, 5\}$, $\{3, 5\}$. For $1 \leqslant i \leqslant 3$ let X_i be the subgraph consisting of all vertices except i and those edges not incident with i. Let $X_0 = \{1, 2, 3\}$. In this case, for each

$1 \leqslant i < j \leqslant 3$ the groupoid $\pi_1(X_i \cap X_j, X_0 \cap X_i \cap X_j)$ consist of a single object and a single identity arrow. It follows that for $X = X_1 \cup X_2 \cup X_3$ the diagram (6) is not a coequalizer diagram.

5. Illustration: enriched Mapper

Low-dimensional cohomology and cup products can be used to enhance the output of the Mapper clustering technique [**29**] for modelling topological properties of metric spaces based on finite random samples. Our aim in this section is to provide an example which motivates the distributed computation of cup products in this context. Our example involves a relatively small data set and so we have performed the computations using the serial implementation of cup products available in [**8**]. Larger examples would require distributed computations, functions for which are being implemented by the first author as part of her PhD work.

In its basic form the Mapper clustering procedure inputs a finite sample S of points from a metric space X, together with a user defined function $f\colon S \to Z$ to some metric space Z and a user defined finite open cover $\mathcal{U} = \{U_p\}_{p\in P}$ of Z. It also inputs a user defined procedure *cluster* for clustering any finite subset $\tilde{U} \subset X$ into a number of distinct clusters V_i which yield a partition $\tilde{U} = V_1 \sqcup V_2 \sqcup \cdots \sqcup V_n$. The Mapper procedure outputs a finite simplicial complex K which is intended to serve as a model for X. The distance $d_X(x,y)$ must be known for all $x, y \in S$ but no further details of the metric space X are required in the construction of K. For simplicity we assume that the user has specified a finite subset $P \subset Z$ and a sufficiently large constant $r > 0$, and that the open cover $\mathcal{U}$ is defined by setting $U_p = \{z \in Z \ : \ d_Z(z,p) < r\}$. As part of the Mapper procedure each finite subset $\tilde{U}_p = \{x \in S \ : \ f(x) \in U_p\} \subseteq S$ is determined and partitioned $\tilde{U}_p = V_{p,1} \sqcup V_{p,2} \sqcup \cdots \sqcup V_{p,n_p}$ using the procedure *cluster*. The simplicial complex K is then defined to be the nerve of the cover $\mathcal{V} = \{V_{p,i}\}_{p\in P, 1\leqslant i \leqslant n_p}$ of S. Thus K has one k-simplex for each subset $\sigma \subset \mathcal{V}$ of size $k+1$ with $\cap_{V\in\sigma} V \neq \emptyset$. The above description of the Mapper clustering procedure can be encoded as a function

$$Mapper(S, d_X, f, d_Z, P, r, cluster),$$

which returns the simplicial complex K. Here d_X and d_Z denote the metrics on X and Z.

In many applications one chooses P, r and f so that the resulting simplicial complex K is 1-dimensional. Graph visualization software such as [**13**] can then be used to investigate K. Each vertex V of K corresponds to a finite subset of S. Information about f could be incorporated into the visualization using colours but our implementation omits this option.

The function f is called the *filter function* and can be chosen to reflect prior knowledge of the space X. If no prior knowledge is available then one can always choose a point $x_0 \in S$ that minimizes $b = Max_{x\in S} d_X(x_0, x)$, where d_Z be the standard Euclidean metric on $Z = [0, b] \subset \mathbb{R}$, and define $f\colon X \to Z$ by $f(x) = d_X(x_0, x)$.

One method of clustering a set $\tilde{U}$ involves choosing a fixed parameter $\epsilon > 0$, forming the graph $G(\tilde{U}, \epsilon)$ with vertex set equal to $\tilde{U}$ and with vertices $x, x' \in \tilde{U}$ connected by an edge if $d_X(x, x') \leqslant \epsilon$, and then taking all vertices in a single connected component of $G(\tilde{U}, \epsilon)$ to constitute a cluster. We refer to this naive method as *graph clustering*. It can be encoded as a function

$$cluster(\tilde{U}, \epsilon, d_X).$$

There are various heuristic approaches for determining a suitable value for ϵ in terms of U and d_X, and so the parameter ϵ can be omitted from the input if desired.

As an illustration we consider a sample $S \subset \mathbb{R}^7$ of 1331 points from Euclidean space and use Mapper to perform an heuristic qualitative investigation of whether these points have been sampled (possibly with errors) from some low-dimensional subspace $X \subset \mathbb{R}^3$. As filter function we take $f\colon S \to Z \subset \mathbb{R}^2$ to be projection of a point onto its first two coordinates. The image of f is contained in a square region of the plane of side 2. We take Z to be this region and take P to be the set of nine vertices of the tiling of Z by unit squares. We take d_X and d_Z to be the Euclidean metrics and use graph clustering. For suitable values of ϵ and r Mapper returns the simplicial graph of Figure 3 (left). A more detailed description of the simplicial graph is shown in Figure 3 (right) in which: the vertices of the graph are denoted by ellipses or circles and labelled by the number of sample points in the cluster corresponding to the vertex; the edges of the graph are labelled by a box containing the number of sample points in the cluster intersection corresponding to the edge.

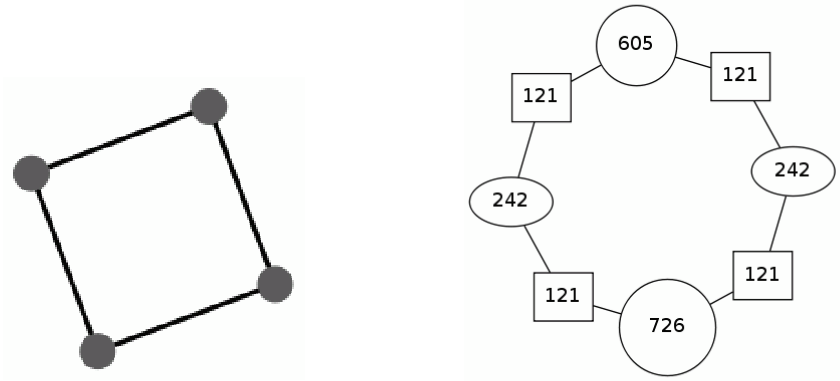


Figure 3: Output of Mapper from sample $S \subset \mathbb{R}^7$ of 1331 points.

As explained in [29], if the filter function f corresponds to a continuous map $f\colon X \to Z$ from the space X that we wish to model, then the open cover $\mathcal{U}$ on Z determines an open cover $f^{-1}\mathcal{U} = \{f^{-1}(U)\}_{U\in\mathcal{U}}$ on X. The homotopy type of X could be closely related to the homotopy type of the nerve of the open cover $f^{-1}\mathcal{U}$. For instance, this nerve will have the same homotopy type as X if X is paracompact and if every non-empty intersection of finitely many sets in $f^{-1}\mathcal{U}$ is contractible [21, Corollary 4G3].

We now describe a cohomological enhancement of the Mapper output aimed at detecting more homotopy-theoretic information.

Let $K = Mapper(S, d_X, f, d_Z, P, r, cluster)$ denote the simplicial complex arising from an application of Mapper. Each vertex V of K is a subset of S. In order to enhance K we first associate to each vertex V of K a regular CW-space X_V such that: (i) V is the vertex set of X_V, (ii) X_V is connected and, (iii) each cell of X_V is uniquely determined by the vertices in the cell boundary. We say that any regular CW-space satisfying property (iii) is *strongly regular*. One way to construct a strongly regular space X_V is to choose an $\epsilon > 0$ and let $X_V = X(V, d_X, \epsilon)$ be the simplicial complex with vertex set V and with k-simplices the subsets $\sigma \subset V$ of $k+1$ distinct elements such that $d_X(x, x') \leqslant \epsilon$ for all $x, x' \in V$. We refer to this particular construction of X_V as the *ϵ-clique complex* on V. For sufficiently large ϵ the ϵ-clique complex is connected. Having decided on some method for constructing the connected space X_V for each vertex V we then define the strongly regular CW-space

$$X_\emptyset = \bigcup_{V \in \mathcal{V}} X_V,$$

where $\mathcal{V}$ denotes the set of vertices of K. This space $X_\emptyset$ is intended to serve as a model for the space X from which the finite set S was sampled.

For each subset $\sigma \subset \mathcal{V}$ we define X_σ to be the CW-subspace of $X_\emptyset$ consisting of all cells whose boundary vertices are contained in $\bigcap_{V \in \sigma} V$. In particular, this associates a strongly regular CW-space X_σ to each simplex σ in K, and an inclusion of CW-spaces $\iota^\tau_\sigma \colon X_\sigma \hookrightarrow X_\tau$ to each subset $\tau \subseteq \sigma$. The spaces X_σ need not be connected for $|\sigma| \neq 1$. One approach to distilling the data contained in the collection $\{X_\sigma \hookrightarrow X_\tau\}_{\emptyset \neq \tau \subseteq \sigma \in K}$ of cellular inclusions is to choose a field $\mathbb{F}$ and integer $n \geqslant 0$ and to apply the cellular cohomology functor $H^n(\ ,\mathbb{F})$. This reduces the data to a collection of linear homomorphisms

$$\{H^n(\iota^\tau_\sigma, \mathbb{F}) \colon H^n(X_\tau, \mathbb{F}) \to H^n(X_\sigma, \mathbb{F})\}_{\emptyset \neq \tau \subseteq \sigma \in K}. \tag{7}$$

The homomorphisms (7) give information on the homology of $X_\emptyset$. For example, if $H^n(X_\sigma, \mathbb{F}) = 0$ for all $\emptyset \neq \sigma \in K$ and $n > 0$, then it is a classical result that $H^n(X_\emptyset, \mathbb{F}) = H^n(K, \mathbb{F})$ for $n \geqslant 0$. In the general case there is a classical spectral sequence involving the homomorphisms (7) which converges to the cohomology of $X_\emptyset$. This spectral sequence arises from a double cochain complex and it is this double cochain complex that needs to be implemented in order to obtain $H^n(X_\emptyset, \mathbb{F})$ in this way on a computer. To describe the double complex we order the vertex set $\mathcal{V} = \{V_0 < V_1 < \cdots < V_m\}$. For each k-simplex $\sigma \in K$ let $\partial_i \sigma$ denote the $(k-1)$-simplex obtained by removing the ith vertex in σ, $0 \leqslant i \leqslant k$. The inclusion $X_\sigma \hookrightarrow X_{\partial_i \sigma}$ induces a chain homomorphism $\partial_i \colon C_*(X_\sigma) \hookrightarrow C_*(X_{\partial_i \sigma})$ of cellular chain complexes. One defines the chain homomorphism

$$\partial = \sum_{i=0}^{k} (-1)^i \partial_i \colon C_*(X_\sigma) \hookrightarrow \bigoplus_{\tau \in K, |\tau| = k} C_*(X_\tau).$$

The chain homomorphism ∂ extends to a chain homomorphism

$$\partial = \sum_{i=0}^{k} (-1)^i \partial_i \colon \bigoplus_{\sigma \in K, |\sigma| = k+1} C_*(X_\sigma) \longrightarrow \bigoplus_{\tau \in K, |\tau| = k} C_*(X_\tau). \tag{8}$$

The homomorphisms (8) form a chain complex

$$\cdots \xrightarrow{\partial} D_k \xrightarrow{\partial} D_{k-1} \xrightarrow{\partial} \cdots$$

in the category of chain complexes, with $D_k = \bigoplus_{\sigma\in K, |\sigma|=k+1} C_*(X_\sigma)$. The double chain complex D_{**} can be dualized to a double cochain complex D^{**} where $D^{m,n} = D_{m,n}$ and where the boundary maps in D^{**} are the transpose of the corresponding maps in D_{**}. The double cochain complex D^{**} gives rise to the above mentioned spectral sequence. The total complex $T^* = T(D^{**})$ of the double cochain complex yields the cohomology of $X_\emptyset$:

$$H^n(T^*) \cong H^n(Hom_{\mathbb{F}}(C_*(X_\emptyset), \mathbb{F})). \tag{9}$$

See for instance [2] for further details on (9).

When the Mapper simplicial complex K is 1-dimensional, information about the linear homomorphisms (7) can be incorporated into the graphical visualization of K. For a given degree n and field $\mathbb{F}$ our implementation displays the dimensions of $H_n(X_\sigma, \mathbb{F})$ inside the circular nodes of the graph for $|\sigma| = 1$, and inside squares on the edges of the graph for $|\sigma| = 2$. The ranks of the linear maps $H^n(\iota_\sigma^\tau, \mathbb{F})$ could also be displayed for $|\sigma| = 2$ and $|\tau| = 1$, though this option is omitted from our implementation.

To illustrate this enhancement of Mapper we return to the sample $S \subset \mathbb{R}^7$ of 1331 points. The enhanced output is shown in Figure 4 for cohomology $H^n(-, \mathbb{F})$ with $\mathbb{F} = \mathbb{Z}/2\mathbb{Z}$ and $n = 1$ (left), $n = 2$ (right). This output is consistent with the possibility that $X_\emptyset$ has the homotopy type of $\mathbb{S}^1 \times Y$ where $H^n(Y, \mathbb{F}) = \mathbb{F}$ for $n = 0, 2$ and $H^1(Y, \mathbb{F}) = \mathbb{F} \oplus \mathbb{F}$. Two obvious candidates for the homotopy type of Y are $Y \simeq \mathbb{S}^1 \times \mathbb{S}^1$ and $Y \simeq \mathbb{S}^1 \vee \mathbb{S}^1 \vee \mathbb{S}^2$. On applying our cup product implementation to each vertex space $X_{\{i\}}$, $1 \leqslant i \leqslant 4$, we find that the cohomology on each of these spaces has non-trivial cup product (1). This suggests that $Y \not\simeq \mathbb{S}^1 \vee \mathbb{S}^1 \vee \mathbb{S}^2$.

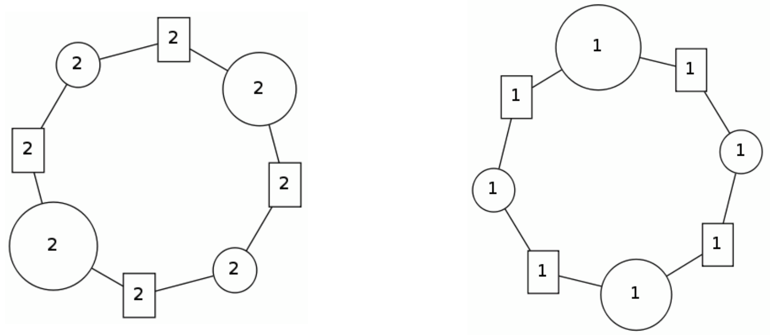


Figure 4: Output of Mapper enhanced with $\dim H^n(-, \mathbb{F})$ for $n = 1$ (left), $n = 2$ (right).

The next step in investigating the possibility of a homotopy equivalence $X_\emptyset \simeq \mathbb{S}^1 \times Y \simeq \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$ would be to compute the cup product on $X_\emptyset = X_{\{1\}} \cup X_{\{2\}} \cup X_{\{3\}} \cup X_{\{4\}}$. The distributed method described in Section 4 is aimed at such a computation. The need for groupoid methods is illustrated by the fact that $X_{\{1\}} \cup X_{\{2\}} \cup X_{\{3\}}$ has a non-connected intersection with $X_{\{4\}}$. Non-synthetic examples of this kind, involving substantially larger data sets, would require the groupoid computations to be distributed over several computers.

To conclude, we mention that the synthetic sample $S \subset \mathbb{R}^7$ of 1331 points was, in fact, chosen from the image of the function

$$\begin{array}{llll} \phi\colon & [0,2\pi]^3 & \longrightarrow & \mathbb{R}^7, \\ & (x,y,z) & \mapsto & (\cos x, \sin x, \cos y, -\sin y, \cos z, \sin z, \cos(x)\sin(y+z)). \end{array}$$

The sample S was constructed as the image of an $11 \times 11 \times 11$ cubical grid of evenly spaced points in the cube $[0,2\pi]^3$.

6. Conclusion

We have given two synthetic examples – one in digital image analysis and one in Mapper cluster analysis of point cloud data sets – to illustrate the use of the cup product in low-dimensional cohomology and to illustrate the practicality of computing this product from presentations of fundamental groups of connected cellular spaces. We have explained how the computation can be distributed over several processors using van Kampen's theorem for fundamental groups and groupoids. Distributed computations using the classical group-theoretic version of the theorem can be performed using the current version of the HAP package [8] for GAP, but require some stringent connectivity assumptions. We have explained how the groupoid version of van Kampen's theorem avoids these connectivity assumptions. More work now needs to be focused on developing GAP software routines for manipulating finitely presented fundamental groupoids in order to perform distributed computations on larger examples. As an aside, we have shown how the Mapper clustering technique can be enhanced by incorporating low-dimensional cohomological information.

References

[1] P. Brendel, P. Dłotko, G. Ellis, M. Juda, and M. Mrozek. Computing fundamental groups from point clouds. *Appl. Algebra Engrg. Comm. Comput.*, 26(1–2):27–48, 2015.

[2] K.S. Brown. *Cohomology of Groups*, volume 87 of *Grad. Texts in Math.* Springer-Verlag, New York, Berlin, 1982.

[3] R. Brown. Groupoids and van Kampen's theorem. *Proc. Lond. Math. Soc. (3)*, 17:385–401, 1967.

[4] R. Brown. *Elements of Modern Topology.* McGraw-Hill Book Co., New York, 1968.

[5] R. Brown. *Topology and Groupoids.* Booksurge LLC, S. Carolina, 2006. Available at http://groupoids.org.uk/

[6] R. Brown, P.J. Higgins, and R. Sivera. *Nonabelian Algebraic Topology.* Filtered spaces, crossed complexes, cubical homotopy groupoids, with contributions by Christopher D. Wensley and Sergei V. Soloviev. Volume 15 of *EMS Tracts Math*, European Mathematical Society (EMS), Zürich, 2011.

[7] R. Brown and A.R. Salleh. A van Kampen theorem for unions of nonconnected spaces. *Arch. Math. (Basel)*, 42(1):85–88, 1984.

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

[9] G. Ellis. GAP file containing the permutahedral complex of Figure 1, 2017. http://hamilton.nuigalway.ie/preprints/permutahedralcomplex.txt

[10] G. Ellis and F. Hegarty. Computational homotopy of finite regular CW-spaces. *J. Homotopy Relat. Struct.*, 9(1):25–54, 2014.

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

[12] R. Forman. A user's guide to discrete Morse theory. *Sém. Lothar. Combin.*, 48:Art. B48c, 35, 2002.

[13] E.R. Gansner and S.C. North. An open graph visualization system and its applications to software engineering. *Software Pract. Exp.*, 30(11):1203–1233, 2000.

[14] R. Gonzalez-Diaz, M.J. Jimenez, and B. Medrano. Cohomology ring of 3D cubical complexes. *Lecture Notes in Comput. Sci.*, 5852:139–150, 2009.

[15] R. Gonzalez-Diaz, J. Lamar, and R. Umble. Cup products on polyhedral approximations of 3D digital images. In *Combinatorial Image Analysis*, volume 6636 of *Lecture Notes in Comput. Sci.*, pages 107–119. Springer, Heidelberg, 2011.

[16] R. Gonzalez-Diaz, J. Lamar, and R. Umble. Computing cup products in $\mathbb{Z}_2$-cohomology of 3D polyhedral complexes. *Found. Comput. Math.*, 14(4):721–744, 2014.

[17] R. González-Díaz and P. Real. Computation of cohomology operations of finite simplicial complexes. *Homology Homotopy Appl.*, 5(2):83–93, 2003. Algebraic Topological Methods in Computer Science (Stanford, CA, 2001).

[18] R. González-Díaz and P. Real. On the cohomology of 3D digital images. *Discrete Appl. Math.*, 147(2–3):245–263, 2005.

[19] A. Grothendieck. Esquisse d'un programme. In *Geometric Galois Actions, 1*, volume 242 of *London Math. Soc. Lecture Note Ser.*, pages 5–48. Cambridge Univ. Press, Cambridge, 1997. With an English translation on pp. 243–283.

[20] S. Harker, K. Mischaikow, M. Mrozek, and V. Nanda. Discrete Morse theoretic algorithms for computing homology of complexes and maps. *Found. Comput. Math.*, 14(1):151–184, 2014.

[21] A. Hatcher. *Algebraic Topology.* Cambridge University Press, Cambridge, New York, 2002. Autre(s) tirage(s): 2003, 2004, 2005, 2006.

[22] P.J. Higgins. Presentations of groupoids, with applications to groups. *Proc. Cambridge Philos. Soc.*, 60:7–20, 1964.

[23] P.J. Higgins. *Notes on Categories and Groupoids*, volume 32 of *Van Nostrand Rienhold Math. Stud.* Van Nostrand Reinhold Co., London, New York, Melbourne, 1971 [Reprinted in Theory and Applications of Categories, No. 7 (2005) pp. 1–195].

[24] T. Kaczynski and M. Mrozek. The cubical cohomology ring: an algorithmic approach. *Found. Comput. Math.*, 13(5):789–818, 2013.

[25] T. Lewiner, H. Lopes, and G. Tavares. Optimal discrete Morse functions for 2-manifolds. *Comput. Geom.*, 26(3):221–233, 2003.

[26] W.S. Massey. *A Basic Course in Algebraic Topology*, volume 127 of *Grad. Texts in Math.* Springer-Verlag, New York, 1991.

[27] M. Morishita. *Knots and Primes: An Introduction to Arithmetic Topology.* Universitext. Springer, London, 2012.

[28] P. Pilarczyk and P. Real. Computation of cubical homology, cohomology, and (co)homological operations via chain contraction. *Adv. Comput. Math.*, 41(1):253–275, 2015.

[29] G. Singh, F. Memoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics.* The Eurographics Association, 2007.

[30] J.H.C. Whitehead. Simple homotopy types. *Amer. J. Math.*, 72:1–57, 1950.

Nisreen Alokbi nisreen.alokbi@gmail.com

School of Mathematics, National University of Ireland, Galway Ireland

Department of Mathematics, College of Computer Science and Mathematics, University of Wasit, Iraq

Graham Ellis graham.ellis@nuigalway.ie

School of Mathematics, National University of Ireland, Galway Ireland