

# Allowing cycles in discrete Morse theory



Aldo Gonzalez-Lorenzo<sup>a,b,\*</sup>, Alexandra Bac<sup>a</sup>, Jean-Luc Mari<sup>a</sup>, Pedro Real<sup>b</sup>

<sup>a</sup> Aix-Marseille Université, CNRS, LSIS UMR 7296, Marseille, France

<sup>b</sup> University of Seville, Institute of Mathematics IMUS, Seville, Spain

## ARTICLE INFO

### Article history:

Received 14 February 2017

Received in revised form 15 May 2017

Accepted 17 May 2017

Available online 24 May 2017

### MSC:

52C99

55U05

68R05

### Keywords:

Computational topology

Homology

CW complex

Discrete Morse theory

Homological discrete vector field

## ABSTRACT

Discrete gradient vector fields are combinatorial structures that can be used for accelerating the homology computation of CW complexes, such as simplicial or cubical complexes, by reducing their number of cells. Consequently, they provide a bound for the Betti numbers (the most basic homological information). A discrete gradient vector field can eventually reduce the complex to its minimal form, having as many cells of each dimension as its corresponding Betti number, but this is not guaranteed. Moreover, finding an optimal discrete gradient vector field is an NP-hard problem. We describe here a generalization, which we call Homological Discrete Vector Field (HDVF), which can overcome these limitations by allowing cycles under a certain algebraic condition. In this work we define the HDVF and its associated reduction, we study how to efficiently compute a HDVF, we establish the relation between the HDVF and other concepts in computational homology and we estimate the average complexity of its computation. We also introduce five basic operations for modifying a HDVF, which can also be applied to discrete gradient vector fields.

© 2017 Elsevier B.V. All rights reserved.

## 1. Introduction

Morse theory [1] is a tool in differential topology that deduces some information of the topology of a manifold by studying a differentiable function on it. In the late 90s, Robin Forman introduced a discrete version, the discrete Morse theory [2,3], which was defined for CW complexes and discrete functions. Several theorems of Morse theory were translated into the discrete context but, in our opinion, the most notable result was the simplification of a CW complex, which can be used to compute its homology groups.

Homology is an algebraic theory that formalizes the concept of “hole” present in an object. It associates a sequence of abelian groups to an object, whose elements correspond to sums of holes. Up to dimension three, these elements have an easy interpretation. Zero-dimensional holes (elements of the zeroth homology group) correspond to connected components, one-dimensional holes to tunnels or handles and two-dimensional

\* Corresponding author at: Aix-Marseille Université, CNRS, LSIS UMR 7296, Marseille, France.

E-mail address: [aldo.gonzalez-lorenzo@univ-amu.fr](mailto:aldo.gonzalez-lorenzo@univ-amu.fr) (A. Gonzalez-Lorenzo).

holes to cavities. When computing homology, we usually want to find the number of these holes (called the Betti numbers), which are the ranks of the homology groups, and a representative for each hole (called representative cycle of a homology generator).

Homology theory was born more than a century ago and, while all kinds of theoretical results in pure mathematics have been developed, its practical applications have not been exploited until the last 20 years due to its computational complexity. There are applications in dynamical systems [4,5], material science [6,7], electromagnetism [8,9], geometric modeling [10], image understanding [11–14] and sensor networks [15]. The general idea of these applications is that homology is used to analyze and understand high dimensional structures in a rigorous way.

The classical method for computing the homology groups is based on the Smith normal form (SNF) [16], which has super-cubical complexity [17]. Some advances in the computation of the SNF have been achieved, but the best results in computing the homology groups of a complex have been obtained by reducing the number of cells in the complex (see [18–20]).

Among other approaches, let us mention the following two, which are closely related to our work: effective homology theory [21] and discrete Morse theory [3]. Both of them are explained in Section 3.4 and 3.5. The former has the advantage that it “controls” the homology because it contains all the homological information [22]; the latter is very concise and easy to implement. Effective homology theory deals with linear maps which are typically encoded as enormous matrices; discrete Morse theory handles only graphs, but does not always allow us to reduce the complex to its minimal homological expression. The use of reductions (the main concept of effective homology theory) has proved to be successful in the context of image analysis [23,24,12,25] or in a more general setting providing more advanced topological information [26,22].

We aim at finding an intermediate solution, avoiding the respective drawbacks of both of these methods whilst maintaining their advantages. Roughly speaking, discrete Morse theory simplifies a CW complex by establishing arrows on it, hence providing a simpler (in terms of number of cells) complex having the same homological information as the original one. In this article we allow cycles in this “collection of arrows”, which is normally forbidden, so that we can go beyond the limits of the classical discrete Morse theory. Moreover, we can control when our approach produces the exact homological information. These allowed cycles must not be confused with the ideas found in [27]. The process of adding these arrows must be simultaneously accompanied by the computation of the linear maps of the effective homology theory, which is unnecessary when there are no cycles. The clearest advantage of our approach over effective homology theory is that we only use linear space for saving these maps, instead of quadratic. Also, our framework works for any dimension, any kind of CW complex and any ring of coefficients.

This article extends the ideas present in [28] under a different formalism, which allows us to find deeper results.

## 2. Previous works

This article somehow creates a new problem instead of solving an existing one. This justifies the shortness of this section.

Discrete Morse theory was introduced in [2,3]. It was then reformulated in terms of matchings in [29,30]. Discrete Morse theory is often used for simplifying a CW complex in order to accelerate the computation of its homology. Thus, it can be seen as an optimization problem, in which one wants to find a discrete gradient vector field (a matching in the Hasse diagram of a CW complex) with as many edges as possible. It was proved that this is an NP-hard problem (see [31,32]). Nevertheless, there has been an extensive research on this optimization problem without aiming at finding a perfect solution in the general case, such as in [31–34]. There has been a parallel and successful research about simplifying a CW complex in [18–20]. These works were recently related to discrete Morse theory in [35], which states that reductions and coreductions are particular strategies for establishing a discrete gradient vector field.

### 3. Preliminaries

#### 3.1. CW complex

Computational topology needs topological spaces that can be described through a finite representation. A rigorous presentation of CW complexes would be too long for this article, so we give an intuitive introduction and we let the reader satisfy its curiosity by consulting [36].

A *CW complex*, or *cell complex*, is a collection of closed unit balls (up to homeomorphism) of different dimensions, called *cells*, that are “glued” together by their boundary: every cell of dimension  $q \geq 1$  ( $q$ -cell) has a map from its boundary to the lower dimensional cells. A  $q$ -cell  $\sigma$  is denoted  $\sigma^{(q)}$  whenever its dimension is not clear from the context. We are certainly interested in the case where the number of cells is finite.

To be honest, we only use the notion of CW complex for comprehending simplicial complexes, cubical complexes [37, §2.1] or even polyhedra [38, §1.1]. We could have chosen to work with S-complexes [19] but we have preferred the CW complexes, as in [39].

We say that a cell  $\sigma$  is a *face* of another one  $\tau$  if it is contained in its boundary. A special case is when they have consecutive dimensions, in which we say that it is a *primary face* and we write  $\sigma < \tau$ .

Given a CW complex, we can define its *Hasse diagram*. It is a directed graph whose vertices represent the cells and whose arrows go from each cell to its primary faces. In this article we usually do not make the distinction between the vertices and the cells they represent, so we mix these terms.

#### 3.2. Chain complex and homology of a CW complex

Let  $(\mathfrak{R}, +, \cdot)$  be a ring, which we simply denote  $\mathfrak{R}$  if its operations are clear from the context. We usually consider  $\mathfrak{R}$  to be  $\mathbb{Z}_2 = \mathbb{Z}/2\mathbb{Z}$  or  $\mathbb{Z}$ . We call  $0_{\mathfrak{R}}$  and  $1_{\mathfrak{R}}$  its neutral elements for the addition and the multiplication respectively. We say that an element  $a$  of  $\mathfrak{R}$  is a *unit* if it is invertible for the multiplication, that is if there exists  $b \in \mathfrak{R}$  such that  $a \cdot b = b \cdot a = 1_{\mathfrak{R}}$ , and we write  $a^{-1} := b$ . We denote by  $\mathfrak{R}^*$  the set of the units of  $\mathfrak{R}$ . For instance,  $\mathbb{Z}^* = \{-1, 1\}$ ,  $\mathbb{Z}_2^* = \{1\}$  and more generally,  $\mathfrak{R}^* = \mathfrak{R} \setminus \{0_{\mathfrak{R}}\}$  if  $\mathfrak{R}$  is a field.

A *chain complex*  $(C, d)$  is a sequence of  $\mathfrak{R}$ -modules  $C_0, C_1, \dots$  (called *chain groups*) and homomorphisms  $d_1 : C_1 \rightarrow C_0, d_2 : C_2 \rightarrow C_1, \dots$  (called *differential* or *boundary operators*) such that  $d_{q-1}d_q = 0$ , for all  $q > 0$ , where  $\mathfrak{R}$  is some ring, called the *ring of coefficients* or *ground ring*.

Given a CW complex  $K$ , we define its associated chain complex  $C(K)$  as follows:

- $C_q$  is  $\mathfrak{R}[K_q]$ , the free  $\mathfrak{R}$ -module generated by the  $q$ -dimensional cells of  $K$ ;
- $d_q$  gives the “algebraic” boundary, which is the linear operator that maps every cell to the sum of its primary faces with specific coefficients. These coefficients, which are not unique, can be computed with the algorithm present in [39, §3.1].

We will usually use the term *complex* for the CW complex or for its associated chain complex. This chain complex can be seen as a sequence of matrices that express the relation of inclusion between the cells. However, note that not every chain complex is the chain complex associated to a CW complex.

The elements of the chain group  $C_q$ , which are formal linear combinations of cells, are called  *$q$ -chains*. If  $x = \sum_{i \in I} \lambda_i \sigma_i$  then  $\langle x, \sigma_i \rangle := \lambda_i$  denotes the coefficient of  $\sigma_i$  in the chain  $x$ . A  $q$ -chain  $x$  is a cycle if  $d_q(x) = 0$ , and a boundary if  $x = d_{q+1}(y)$  for some  $(q+1)$ -chain  $y$ . We do not write the subscripts when it is clear from the context. By the property  $d_{q-1}d_q = 0$ , every boundary is a cycle, but the reverse is not true: a cycle which is not a boundary contains a “hole”. The  $q$ -th homology group of the chain complex  $(C, d)$  contains the  $q$ -dimensional “holes”:  $H(C)_q = \ker(d_q) / \text{im}(d_{q+1})$ . This set is a finitely generated group, so there is generating set (a “base”) typically formed by the holes of the complex. By the Fundamental Theorem of Finitely Generated Abelian Groups [40, §5.2], there are two different “normalizations”:

1. The group is isomorphic to  $\mathbb{Z}^{\beta_q} \times \mathbb{Z}/\lambda_1\mathbb{Z} \times \mathbb{Z}/\lambda_2\mathbb{Z} \times \dots$ , where each  $\lambda_i$  divides  $\lambda_{i+1}$ . This is called the *invariant factor decomposition*.
2. The group is isomorphic to  $\mathbb{Z}^{\beta_q} \times \mathbb{Z}/\lambda_1\mathbb{Z} \times \mathbb{Z}/\lambda_2\mathbb{Z} \times \dots$ , where each  $\lambda_i$  is a power of some prime number. This is called the *primary decomposition*.

As most of the literature about computational homology, we use the first decomposition. The number  $\beta_q$  is called  $q$ -th *Betti number* and  $\lambda_1, \dots, \lambda_t$  are the *torsion coefficients* of dimension  $q$ . Let us recall that if the CW complex is embedded in  $\mathbb{R}^3$ , it has no torsion coefficients.

These homology groups depend on the ring of coefficients, but they can all be deduced from the homology groups with coefficients in  $\mathbb{Z}$  by the Universal Coefficient Theorem [41, §3.A].

The Hasse diagram of a CW complex is actually a weighted graph, whose values over the arrows are given by the boundary operators.

There is a dual theory of homology which defines the *cohomology* groups [16, §5]. Roughly speaking, it considers the dual of the differential operators.

### 3.3. Homological information

Given the definition of the homology groups, we could ask ourselves how much information we want to obtain. If we want to use homology as a topological invariant, it should be enough to know its Euler–Poincaré characteristic or, more generally, its Betti numbers and torsion coefficients. Moreover, if we want to use homology to better understand the shape of the complex, we could be interested in knowing a representative of each class of homology that is a generator. These representatives, which we directly call *homology generators*, are not unique at all, and it is an interesting and ill-defined problem to find a set of well-shaped generators. Furthermore, we can decompose a given cycle onto the computed homology generators.

Since not all works in computational homology try to obtain the same information, we propose the following classification of homological information:

**Level 0:** The Euler–Poincaré characteristic [41, p. 146].

**Level 1:** The Betti numbers. They are the rank of the free part of the homology groups.

**Level 2:** Invariant factor decomposition of the homology groups.

**Level 3:** Homology group with generators:  $(\mathbb{Z}[z_1] \times \dots \times \mathbb{Z}[z_{\beta_q}]) \times \mathbb{Z}[t_1]/\lambda_1\mathbb{Z}[c_1] \times \mathbb{Z}[t_2]/\lambda_2\mathbb{Z}[c_2] \times \dots$ .

**Level 4:** Homology group with generators and decomposition of cycles.

Each level of homological information can be trivially deduced from the higher ones. We have decided to start from level 0 since the Euler–Poincaré characteristic is the easiest computable homological information. Persistent homology usually works at level 1 (in each complex of the filtration), which is equivalent to level 2 because the ring of coefficients considered is usually a field; Munkres’ original theorem/method arrives to level 2 and the *modified-SNF* [42] reaches the third level. Effective homology theory arrives to the fourth level whenever we have a perfect reduction (see Section 3.4), since for a given cycle  $x \in \ker(d_q)$ ,  $f(x)$  decomposes it onto a linear combination of generators.

This classification could be extended with (co)homology operations, the cohomology ring or even the homotopy groups.

### 3.4. Reduction

A *reduction* is a strong relation between two chain complexes which guarantees that they have isomorphic homology groups. This is the main tool in *effective homology theory* [21]. We typically reduce the initial chain complex to another one much smaller (called *reduced complex*).

Formally, a reduction between two chain complexes  $(C, d)$  and  $(C', d')$  is a triplet of graded homomorphisms  $\rho = (h, f, g)$  such that:

- $h_q : C_q \rightarrow C_{q+1}$  for every  $q \geq 0$
- $f_q : C_q \rightarrow C'_q$  is a chain map:  $f_{q-1}d_q = d'_q f_q$
- $g_q : C'_q \rightarrow C_q$  is also a chain map:  $g_{q-1}d'_q = d_q g_q$
- $gf = 1_C - dh - hd$
- $fg = 1_{C'}$
- $hh, fh, hg = 0$

For instance, consider the following chain complexes  $(C, d)$  and  $(C', d')$ , whose chain groups are freely generated with  $\mathfrak{R} = \mathbb{Z}$ :

$$\begin{aligned}
 C_0 &= \langle \sigma_1, \sigma_2, \sigma_3 \rangle, & C_1 &= \langle \sigma_4, \sigma_5, \sigma_6 \rangle, & C_2 &= \langle \sigma_7 \rangle, & C_3 &= 0, & \dots \\
 d_1 &= \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 1 \\ 1 & 1 & 0 \end{bmatrix}, & d_2 &= \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix} \\
 C'_0 &= \langle \tau_1, \tau_2 \rangle, & C'_1 &= \langle \tau_3 \rangle, & C'_2 &= 0, & \dots \\
 d'_1 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}, & d'_2 &= 0
 \end{aligned}$$

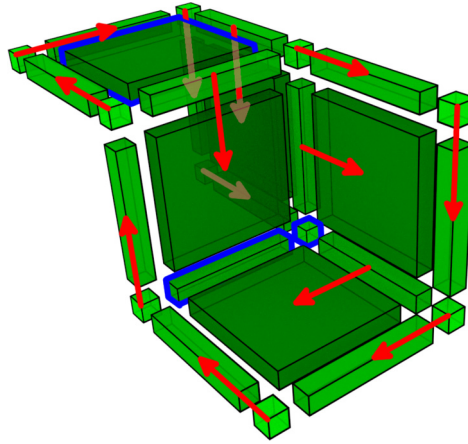
Hence  $(h, f, g)$  is a reduction, where

$$\begin{aligned}
 h_0 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, & h_1 &= \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} \\
 f_0 &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & f_1 &= \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \\
 g_0 &= \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, & g_1 &= \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}
 \end{aligned}$$

A reduction is *perfect* if  $d' = 0$ . In such case,  $H(C) \cong H(C') = C'$  and thus the homology groups are directly obtained. Moreover,  $g(C')$  is a basis for  $H(C)$ . Also, given a cycle  $x \in C_q$ , it is a boundary if  $f_q(x) = 0$  and in that case  $x = d_{q+1}(y)$  for the chain  $y = h_q(x)$ . This should justify the interest of having a perfect reduction.

Let us point out that if the homology groups of a chain complex have a torsion subgroup then there is no perfect reduction, since a perfect reduction involves homology groups freely generated, and hence of the form  $\mathbb{Z}^\beta$ . Also, a reduction can always be obtained via the Smith normal form computation as described in [43, p. 48]. This reduction is perfect if the homology groups are torsion-free. Otherwise, its reduced boundary matrices are in the Smith normal form.

If  $\rho = (h, f, g)$  is a reduction from  $(C, d)$  to  $(C', d')$ , it is easy to prove that  $\rho^* = (h^*, g^*, f^*)$  is a reduction between the cochain complexes  $(C, d^*)$  and  $(C', (d')^*)$ . Consequently, a perfect reduction also provides a basis for the cohomology groups, namely  $f^*(C)$ .



**Fig. 1.** A DGVF over a cubical complex. The critical cells are highlighted in blue. The integral arrows are shown in red. The differential arrows are omitted. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

### 3.5. Discrete Morse theory

Discrete Morse theory was introduced by Robin Forman as a discretization of Morse theory [3]. One of the main ideas is to obtain some homological information by means of a function defined on a complex. This function is equivalent to a discrete gradient vector field and we rather use this notion.

A *discrete vector field* (DVF) on a CW complex is a matching on its Hasse diagram, that is a collection of edges such that no two of them have a common vertex. From a Hasse diagram and a discrete vector field we can define a *Morse graph*: it is a graph similar to the Hasse diagram except for the arrows contained in the matching, which are reversed. These arrows are called *integral arrows*, and the other ones, *differential arrows*.

Given a DVF over a CW complex  $K$ , its cells can be partitioned into the following classes:

**Primary (P)** The cells having an out-going integral arrow.

**Secondary (S)** The cells having an in-going integral arrow.

**Critical (C)** The cells not incident to any integral arrow.

Since a DVF is a matching, it is immediate that  $K = P \sqcup S \sqcup C$ . This notation is inspired by [34, Def. 1], but this classification was previously introduced in [35, Def. 3.1] and [44, §5] with a different notation.

A  $\mathcal{V}$ -*path* is a path on the Morse graph that alternates between integral and differential arrows. Its *length* is the number of integral arrows contained. A *discrete gradient vector field* (DGVF) is a discrete vector field that does not contain any closed  $\mathcal{V}$ -path. As mentioned above, a *critical vertex* (or critical cell) is a vertex that is not paired by the matching. Fig. 1 shows the usual representation of a DGVF over a cubical complex.

One of the main results of discrete Morse theory is that the number of critical  $q$ -cells is greater than or equal to the  $q$ -th Betti number. When they are equal, we say that the DGVF is *perfect*. An *optimal* DGVF contains the least possible number of critical cells. Every perfect DGVF is obviously optimal, but the converse is false. Therefore, a DGVF gives an estimation of the Betti numbers without using any algebraic method. We could say that it is a “combinatorial” tool.

Given a DGVF  $\mathcal{V}$  on a CW complex, we define its *Morse complex*  $(\mathfrak{R}[C], d_M)$ , where  $\mathfrak{R}[C]$  denotes the graded free  $\mathfrak{R}$ -module generated by the critical cells of  $\mathcal{V}$  and  $d_M$  is the linear map that sends each critical  $(q + 1)$ -cell  $\sigma$  to the sum of critical  $q$ -cells which are connected to  $\sigma$  by a  $\mathcal{V}$ -path. An accurate definition of this map (called reduced boundary) will be given in Section 5.

Let us point out that starting from a DGVF  $\mathcal{V}$ , an associated reduction  $(h, f, g) : (C, d) \Rightarrow (f(C), d)$  can be defined [2,34]. Firstly, let us define a linear operator  $V$  which maps vertices of the Morse graph containing an outward integral arrow to the head of this arrow with its sign. Formally,

$$V(\sigma) = \begin{cases} \langle d(\tau), \sigma \rangle \cdot \tau, & (\sigma, \tau) \in \mathcal{V} \\ 0, & \text{otherwise} \end{cases}$$

where  $\langle \cdot, \cdot \rangle$  is the inner product associated to the basis of cells. In other words,  $V$  maps each primary cell to its secondary cell. Then,

$$\begin{aligned} h(\sigma) &= \sum_{k \geq 0} V(1 - dV)^k(\sigma) \\ f(\sigma) &= (1 - dh - hd)(\sigma) \\ g(\sigma) &= \sigma \end{aligned}$$

Notice that the sum in the equation for  $h$  is actually finite since a DGVF has no cycles. The image of  $h$  can be interpreted as the sum of the secondary cells in all the  $\mathcal{V}$ -paths leaving a primary cell. Furthermore, the map  $f$  coincides with the *stabilization map*  $\Phi^\infty$  introduced in [2].

Let us point out that this reduction can be encoded as a matching in the Hasse diagram instead of using a sequence of matrices. Thus, the DGVF actually “compresses” the reduction as it can be described in linear space, which obviously involves a computational cost for its “decompression”. Actually, this is not a general property of reductions, since not all reductions can be represented with a DGVF, as the following example shows.

Consider the simplicial complex with one 1-cell (and its two 0-faces) whose boundary matrix is

$$d_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Then there is a reduction  $(h, f, g) : (C, d) \Rightarrow (f(C), d|_{f(C)})$ , where

$$h_0 = \begin{bmatrix} 2 & 3 \end{bmatrix}, \quad f_0 = \begin{bmatrix} 3 & 3 \\ -2 & -2 \end{bmatrix}, \quad f_1 = \begin{bmatrix} 0 \end{bmatrix}, \quad g = inc$$

In this case we cannot find how to “compress” the reduction as a matching, so we can only explicitly encode it as a sequence of matrices, hence needing quadratic space.

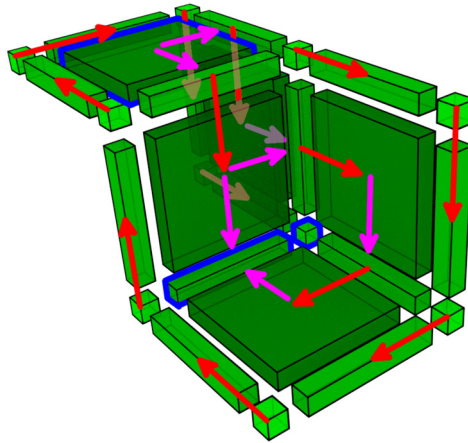
#### 4. Motivation

The discrete Morse theory approach has a strong interest as it addresses the computation of homology as a purely combinatorial problem rather than an algebraic one. The associated reduction can be encoded just as a list of pairs of cells. It also provides an approximation of the Betti numbers that can sometimes be accurate (depending on the choice of the integral arrows) but that is always wrong for some well known spaces as, for instance, the Bing’s house [45] (also called house with two rooms) or the dunce hat [46].

We can increase a DGVF (and thus improve the approximation) by canceling pairs of critical cells: find two critical cells  $\tau^{(q+1)}$  and  $\sigma^{(q)}$  connected by only one  $\mathcal{V}$ -path and exchange the integral and differential arrows in this path. This can be seen as reversing the direction of the  $\mathcal{V}$ -path. Note that, even though this transformation is expressed in combinatorial terms, computing the number of  $\mathcal{V}$ -paths is equivalent to compute the associated reduction.



**Fig. 2.** Left: an iterated Morse decomposition, where the red arrow belongs to the first DGVF and the purple one, to the second DGVF. Right: a (standard) DGVF inducing the same reduction. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 3.** The same DGVF depicted in Fig. 1. Some differential arrows are shown in purple. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Another approach for reducing the number of critical cells is to compute the Morse complex and to establish a new DGVF  $\mathcal{V}'$  on it, which is useful when there is no unique  $\mathcal{V}$ -path between the critical cells. This is known as *iterated Morse decomposition* [47]. Regarding the associated reduction, reversing the only  $\mathcal{V}$ -path between two critical cells is equivalent to adding an integral arrow between them in the Morse complex. Fig. 2 illustrates this.

Thus, reversing a  $\mathcal{V}$ -path can be seen as pushing an integral arrow from the Morse complex back to the original one. However, not all the integral arrows on the Morse complex are equivalent to reverse a  $\mathcal{V}$ -path: this is the case when there are several  $\mathcal{V}$ -paths between two critical cells. Fig. 3 shows an example where there are three  $\mathcal{V}$ -paths between two critical cells. However, the 1-cell is a face of the 2-cell in the associated Morse complex, so we can add an integral arrow which does not correspond to a unique  $\mathcal{V}$ -path. The motivation for our work was to push all the integral arrows in the Morse complex back to the original one.

## 5. Introducing the HDVF

In the context of discrete Morse theory, we always try to set a DGVF with the maximum number of integral arrows (or equivalently, with the minimum number of critical cells) in order to obtain the best possible approximation of the Betti numbers. In the language of effective homology theory, the induced reduction greatly “reduces” the original chain complex.

Given a DGVF, we can improve it by increasing the number of integral arrows. If we find two critical cells  $\sigma < \tau$ , such that inserting an integral arrow between them does not create a cycle, adding this integral arrow reduces by two the number of critical cells.

More generally, if there is only one  $\mathcal{V}$ -path between one cell  $\sigma'$  belonging to the boundary of a critical cell  $\tau$  and another critical cell  $\sigma$ , we can reverse it and add the arrow  $(\sigma', \tau)$ . This means that the integral



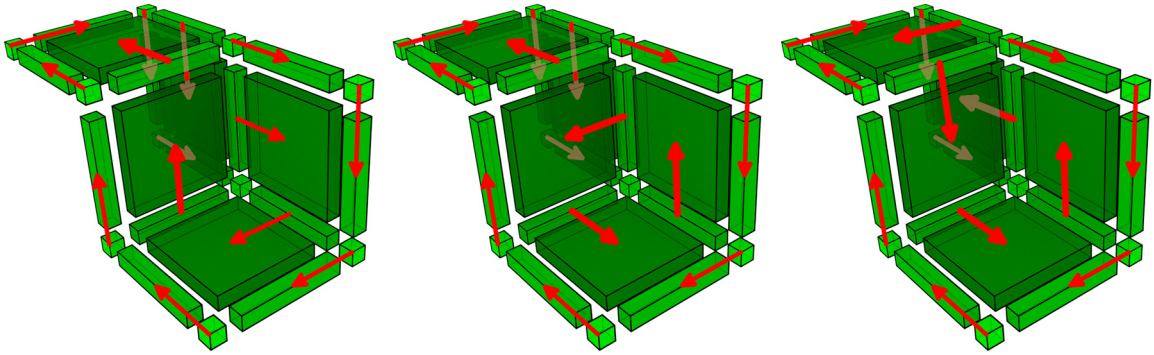


Fig. 4. Three different matchings inducing the same HDVF.

and differential arrows in the  $\mathcal{V}$ -path are exchanged. This can be considered as the general method for improving a DGVF (actually, in the previous case, the  $\mathcal{V}$ -path has length zero so there is no reversing). However, depending on the order in which we cancel the critical cells and on the CW complex itself, we can create several  $\mathcal{V}$ -paths between the other pairs of critical cells, so that we cannot cancel them anymore. This gives an intuition on why this optimization problem is NP-hard [31].

In order to avoid this situation, we propose to allow cycles in the DVF, provided that we create them “smartly”, so a reduction can still be defined. We cancel pairs of critical cells independently of the number of  $\mathcal{V}$ -paths, but considering the information given by the associated reduction. This means that the reduction must be known at every step, but do not panic: finding a  $\mathcal{V}$ -path amounts also to compute a reduction.

We recall that a DVF induces a partition  $K = P \sqcup S \sqcup C$  of a CW complex.

**Definition.** A homological discrete vector field (HDVF) on a CW complex  $K$  is a pair of disjoint sets of cells  $X = (P, S)$  of  $K$  such that  $d(S_{q+1})|_{P_q}$  is an invertible matrix (in  $\mathfrak{R}$ ) for every  $q \geq 0$ , where  $P_q$  and  $S_q$  denote the restrictions of  $P$  and  $S$  to the  $q$ -cells and  $d(S_{q+1})|_{P_q}$  is the submatrix of the boundary matrix  $d_{q+1}$  consisting in the columns associated to the secondary  $(q + 1)$ -cells and the rows associated to the primary  $q$ -cells.

Note that the DVF is not explicit in the definition of the HDVF. When  $X$  is a DGVF, there is a unique DVF inducing its partition, but this is not the case for a HDVF. For instance, Fig. 4 depicts three different DVFs inducing the same HDVF, since the primary and secondary cells in each complex are the same.

Deducing a DVF from a HDVF requires to find a perfect matching in a bipartite graph. The existence of this perfect matching, when the partition is a HDVF, follows from Proposition 5.1.

**Proposition 5.1.** Let  $K$  be a CW complex endowed with a HDVF  $X = (P, S)$ . Then there exists a discrete vector field  $\mathcal{V}$  that induces the partition  $K = P \sqcup S \sqcup C$ .

**Proof.** In this proof we do not use the fact that  $d(S_{q+1})|_{P_q}$  is invertible, but that  $\det(d(S_{q+1})|_{P_q}) \neq 0$ .

Let us fix a dimension  $q$ . By the Laplace expansion formula, there is a pair of cells  $(\sigma, \tau)$  such that  $\langle d(\tau), \sigma \rangle \neq 0$  and  $\det(d(S_{q+1} \setminus \tau)|_{P_q \setminus \sigma}) \neq 0$ . Thus, the discrete vector field  $\mathcal{V}$  can be found recursively.  $\square$

The DVF can be computed using the Hopcroft–Karp algorithm [48] in  $\mathcal{O}(m\sqrt{n})$  time, where  $n$  and  $m$  denote the number of vertices and edges in the Hasse diagram. It is interesting as it allows us to visualize the HDVF and its computation.

Let us now present the reduction induced by a HDVF. We showed in Section 3.5 a reduction induced by a DGVF. Since a DGVF has no cycles, the chain  $(1 - dV)$  is nilpotent and hence the sum  $\sum_{k \geq 0} V(1 - dV)^k$  is well defined. This does not hold for the HDVF, and therefore we must consider an appropriate reduction.

Note that all the operators of a reduction are linear, so they can be represented by matrices. An appropriate choice of bases can provide nice matrices and we have found a very good one: the basis  $\mathcal{B} = \langle P_q, S_q, C_q \rangle$  for every chain group  $C_q$ . In the following we omit the subscripts to facilitate readability.

**Theorem 5.2.** *Let  $K$  be a CW complex endowed with a HDVF  $X$ . Then  $X$  induces the reduction  $(h, f, g) : (C, d) \Rightarrow (\mathfrak{R}[C], d')$ , where  $\mathfrak{R}[C]$  is the chain group generated by the critical cells of the HDVF and the operators  $h, f, g$  and the reduced boundary  $d'$  are given by*

$$\begin{array}{c}
 \begin{array}{ccc}
 & P & S & C \\
 P & \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline \end{array} \\
 S & \begin{array}{|c|c|c|} \hline H & 0 & 0 \\ \hline \end{array} \\
 C & \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline \end{array}
 \end{array}
 &
 f = C \begin{array}{ccc}
 & P & S & C \\
 \begin{array}{|c|c|c|} \hline F & 0 & I \\ \hline \end{array}
 &
 g = \begin{array}{ccc}
 & C \\
 P & \begin{array}{|c|} \hline 0 \\ \hline \end{array} \\
 S & \begin{array}{|c|} \hline G \\ \hline \end{array} \\
 C & \begin{array}{|c|} \hline I \\ \hline \end{array}
 \end{array}
 &
 d' = C \begin{array}{ccc}
 & C \\
 \begin{array}{|c|} \hline D \\ \hline \end{array}
 \end{array}
 \end{array}$$

where

$$\begin{aligned}
 H &= (d(S)|_P)^{-1} \\
 F &= -d(S)|_C \cdot (d(S)|_P)^{-1} \\
 G &= -(d(S)|_P)^{-1} \cdot d(C)|_P \\
 D &= d(C)|_C + F \cdot d(C)|_P = d(C)|_C + d(S)|_C \cdot G
 \end{aligned}$$

**Proof.** Let us see that these linear operators satisfy the conditions of a reduction. By developing the matrix products by blocks, we can easily check that  $hh = 0, fh = 0, hg = 0$  and  $fg = 1_C$ . The rest of the conditions require more detail.

◆  $gf = 1_C - dh - hd$ : By developing the matrix product, we obtain

$$\left[ \begin{array}{c|c|c} 0 & 0 & 0 \\ \hline GF & 0 & G \\ \hline F & 0 & I \end{array} \right] = \left[ \begin{array}{c|c|c} I - d(S)|_P H & 0 & 0 \\ \hline -d(S)|_S H - Hd(P)|_P & I - Hd(S)|_P & -Hd(C)|_P \\ \hline -d(S)|_C H & 0 & I \end{array} \right]$$

All the equalities can be deduced directly from the definition of  $H, F$  and  $G$ . The equality  $GF = -d(S)|_S H - Hd(P)|_P$  is more difficult to see. Let us call

$$\begin{aligned}
 X &= GF + d(S)|_S H + Hd(P)|_P \\
 &= Hd(C)|_P d(S)|_C H + d(S)|_S H + Hd(P)|_P
 \end{aligned}$$

Then,

$$\begin{aligned}
 d(S)|_P X d(S)|_P &= [d(S)|_P H] d(C)|_P d(S)|_C [Hd(S)|_P] \\
 &\quad + d(S)|_P d(S)|_S [Hd(S)|_P] + [d(S)|_P H] d(P)|_P d(S)|_P \\
 &= d(C)|_P d(S)|_C + d(S)|_P d(S)|_S + d(P)|_P d(S)|_P
 \end{aligned}$$

Then, the reader can check that  $d(S)|_P X d(S)|_P = (dd)(S)|_P = 0$ , so  $X = 0$ .

We need now some properties whose proof is direct by developing the matrix product:

$$d' = fdg = fd \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} = \left[ 0 \mid 0 \mid I \right] dg \tag{1}$$

$$f = \left[ 0 \mid 0 \mid I \right] \cdot (1_C - dh) \tag{2}$$

$$g = (1_C - hd) \cdot \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \tag{3}$$

◆  $d'f = fd$ . Using (1) and (2),

$$\begin{aligned} d'f &= \left[ 0 \mid 0 \mid I \right] dgf \\ &= \left[ 0 \mid 0 \mid I \right] d(1_C - dh - hd) \\ &= \left[ 0 \mid 0 \mid I \right] (1_C - dh)d = fd \end{aligned}$$

◆  $gd' = dg$ . Symmetrically,

$$\begin{aligned} gd' &= gfd \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \\ &= (1_C - dh - hd)d \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \\ &= d(1_C - hd) \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} = dg \end{aligned}$$

◆  $d'd' = 0$ . Using (2) and (3),

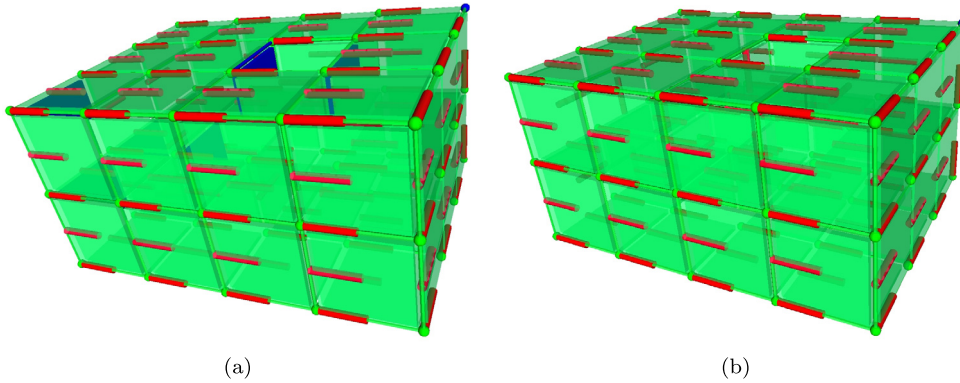
$$d'd' = (fdg)(fdg) = fdg(d'f)g = f(dd)gfg = 0 \quad \square$$

The previous theorem allows us to prove the desired property that the number of critical cells approximates the Betti numbers also in the HDVF.

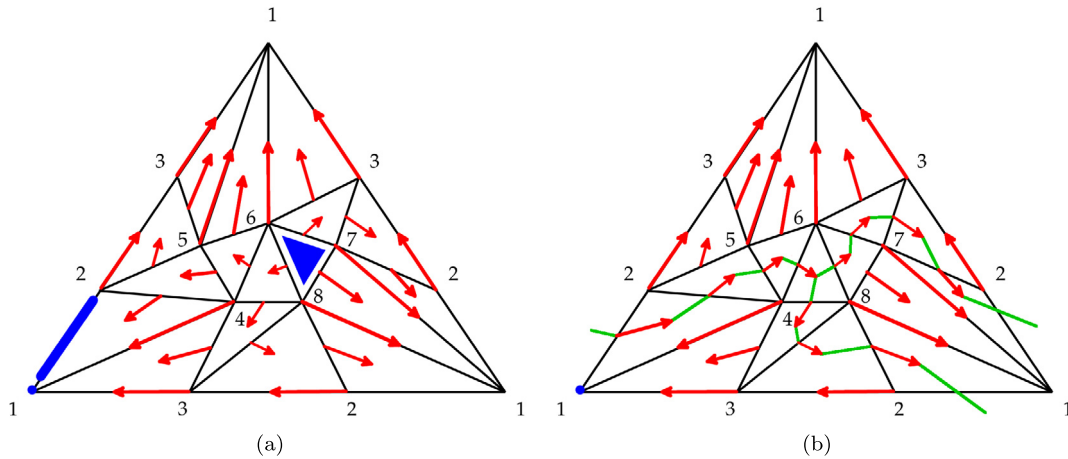
**Theorem 5.3.** *Let  $K$  be a CW complex endowed with a HDVF  $X$ . Then, for every  $q \geq 0$ , the number of  $q$ -critical cells is greater than or equal to its  $q$ -th Betti number.*

**Proof.** A HDVF induces a reduction to a chain complex  $C'$  with isomorphic homology groups, whose rank in each dimension  $q$  is the number of critical  $q$ -cells. This proves the theorem.  $\square$

Let us point out that this reduction is not directly a generalization of the reduction introduced in [34]. Though, it has a similar form if we consider the reduction  $\rho' = (h', f', g') = (h, 1 - dh - hd, \iota)$  between  $(C, d)$  and  $(f'(C), d)$ .



**Fig. 5.** (a) A DGVF over the Bing's house. (b) A perfect HDVF obtained on the Bing's house. There is only one critical 0-cell (in blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 6.** (a) A DGVF over the dunce hat with three critical cells in blue. (b) The HDVF obtained after improving the DGVF. The only critical cell is the 0-cell denoted by 1. The two cycles in the Morse graph are displayed in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We say that a HDVF is *perfect* if its associated reduction is perfect. Using the same language as discrete Morse theory, the HDVF allows us to find the correct number of critical cells in complexes which do not admit a perfect DGVF, such as the Bing's house or the dunce hat [49]. Instead of providing the explicit (and enormous) description of each complex and its HDVF, we prefer to show illustrations and to comment the construction of those HDVFs.

The cubical complex version of the Bing's house has been created by the authors. It contains 60 0-cells, 129 1-cells and 70 2-cells. The first DGVF defined on it contains 13 critical cells (see Fig. 5-(a)): 1 0-cell, 6 1-cells and 6 2-cells. Let us comment that it is not the best DGVF possible. Starting from this DGVF, and after canceling pairs of critical cells by reversing  $\mathcal{V}$ -paths, it remains only 1 critical 0-cell, which corresponds to the Betti numbers of the complex. Obviously, these  $\mathcal{V}$ -paths were chosen to preserve the HDVF structure. Consequently, the Morse graph contains two cycles.

For the dunce hat we used a simplicial complex from [50] consisting of 8 0-cells, 24 1-cells and 17 2-cells. We can set a DGVF containing 3 critical cells (see Fig. 6-(a)): one of each dimension. After reversing one  $\mathcal{V}$ -path between the critical cells of dimension 1 and 2, we obtain a HDVF with only 1 critical 0-cell, which is in accordance with the Betti numbers of the complex. The two cycles created in the homological DVF are shown in green.

### 6. Computing a HDVF

We explain in this section how we can compute a HDVF and its reduction efficiently.

#### 6.1. Computing the reduced complex

Our first proposition states when we can add a pair of cells to a HDVF so that the matrix  $d(S)|_P$  is still invertible.

**Proposition 6.1.** *Let  $K$  be a CW complex endowed with a HDVF  $X = (P, S)$ . Let  $\sigma^{(q)}$  and  $\tau^{(q+1)}$  be two critical cells. If  $\langle d'(\tau), \sigma \rangle$  is a unit then  $X' = (P \cup \{\sigma\}, S \cup \{\tau\})$  is a HDVF.*

**Proof.** We only need to prove that the matrix  $d(S')|_{P'}$  is invertible, where  $S' = S \cup \{\tau\}$  and  $P' = P \cup \{\sigma\}$ . This matrix has the form

$$d(S')|_{P'} = \begin{array}{c} \begin{array}{cc} & \begin{array}{c} S \qquad \tau \end{array} \\ \begin{array}{c} P \\ \sigma \end{array} & \begin{array}{|c|c|} \hline & \\ \hline d(S)|_P & v \\ \hline u & w \\ \hline \end{array} \end{array} \end{array}$$

where  $u = d(S)|_\sigma$ ,  $v = d(\tau)|_P$  and  $w = d(\tau)|_\sigma$ .

We know that  $d(S)|_P$  is invertible. Let us prove that  $w - u(d(S)|_P)^{-1}v$  is also invertible. By hypothesis,  $\langle d'(\tau), \sigma \rangle = \pm 1$ . Since

$$D = d(C)|_C - d(S)|_C \cdot (d(S)|_P)^{-1} \cdot d(C)|_P \quad (\text{cf. Theorem 5.2})$$

then, by Lemma A.1 (without specifying the indices),

$$\begin{aligned} \langle d'(\tau), \sigma \rangle &= d(\tau)|_\sigma - d(S)|_\sigma \cdot (d(S)|_P)^{-1} \cdot d(\tau)|_P \\ &= w - u \cdot (d(S)|_P)^{-1} \cdot v \end{aligned}$$

Consequently, by the Schur determinant formula (cf. Lemma A.3),

$$\det(d(S')|_{P'}) = \det(d(S)|_P) \cdot \det(w - u(d(S)|_P)^{-1}v)$$

is a unit, so  $d(S)|_P$  is invertible.  $\square$

Once we have added two critical cells to a HDVF, we do not need to compute a new DVF inducing the expanded HDVF. Instead of this, we can deduce the corresponding DVF by inverting one of the  $\mathcal{V}$ -paths connecting both critical cells. The following proposition proves that such  $\mathcal{V}$ -path exists.

**Proposition 6.2.** *Let  $K$  be a CW complex endowed with a HDVF  $X$ . Let  $\sigma^{(q)}$  and  $\tau^{(q+1)}$  be two critical cells. If  $\langle d'(\tau), \sigma \rangle$  is a unit then there is a  $\mathcal{V}$ -path between them.*

**Proof.** Let  $V$  denote the matrix associated with the DVF introduced in Section 3.5. Thus,

$$\begin{aligned} d'(C) &= d(C)|_C - d(S)|_C \cdot (d(S)|_P)^{-1} \cdot d(C)|_P \\ &= d(C)|_C - d(S)|_C \cdot V(P)|_S \cdot (V(P)|_S)^{-1} \cdot (d(S)|_P)^{-1} \cdot d(C)|_P \\ &= d(C)|_C - dV(P)|_C \cdot (dV(P)|_P)^{-1} \cdot d(C)|_P \end{aligned}$$

Hence  $\langle d'(\tau), \sigma \rangle = d'(\tau)|_\sigma = -dV(P)|_\sigma \cdot (dV(P)|_P)^{-1} \cdot d(\tau)|_P + d(\tau)|_\sigma$ . If  $\sigma < \tau$  then it is obvious that there is a  $\mathcal{V}$ -path between them. Otherwise,  $\langle d'(\tau), \sigma \rangle = -dV(P)|_\sigma \cdot (dV(P)|_P)^{-1} \cdot d(\tau)|_P$ . As this term is non-zero, there must be some  $\sigma_i, \sigma_j$  in  $P$  such that

$$\begin{aligned} dV(\sigma_i)|_\sigma &\neq 0 \\ d(\tau)|_{\sigma_j} &\neq 0 \\ (dV(P)|_P)^{-1}_{(i,j)} &\neq 0 \end{aligned}$$

The first two inequalities imply that there exist two  $\mathcal{V}$ -paths  $\sigma_i \nearrow b \searrow \sigma$  and  $\tau \searrow \sigma_j$ . The third one implies that there is a path from  $\sigma_j$  to  $\sigma_i$ . Let us see a short proof of this in a more general context.

Let  $A$  be the adjacency matrix of a weighted digraph such that  $\det(A) \neq 0$ . We know that  $A^k_{(i,j)} \neq 0, i \neq j$  implies that there is a path from the vertex  $j$  to  $i$ . Following the Cayley–Hamilton theorem [51, §4.4, Thm. 2],

$$\begin{aligned} A^n + c_{n-1}A^{n-1} + \dots + c_1A + (-1)^n \det(A)I_n &= 0 \\ \Rightarrow A^{-1} &= \frac{(-1)^{n+1}}{\det(A)} (A^{n-1} + c_{n-1}A^{n-2} + \dots + c_1I_n) \end{aligned}$$

Thus,  $A^{-1}_{(i,j)} \neq 0 \Rightarrow \exists k \geq 0, A^k_{(i,j)} \neq 0$ .

Then, there is a  $\mathcal{V}$ -path  $\sigma_j \nearrow b_1 \searrow a_2 \dots \searrow \sigma_i$ . By concatenating these paths, we obtain

$$\tau \searrow \sigma_j \nearrow b_1 \searrow a_2 \dots \searrow \sigma_i \nearrow b \searrow \sigma \quad \square$$

Algorithm 1 gives a general pipeline for computing a HDVF.

---

**Algorithm 1:** Compute a HDVF.

---

**Input:** A CW complex  $K$   
**Output:** A HDVF  $X$

```

1  $X \leftarrow (\emptyset, \emptyset)$ ;
2 repeat
3   Find two critical cells  $\sigma, \tau$  such that  $\langle d'(\tau), \sigma \rangle$  is a unit;
4   Add  $(\sigma, \tau)$  to  $X$ ;
5   Update the boundary matrices  $D$  of the reduced chain complex;
6 until idempotency;
```

---

If we also want to obtain the DVF, for each pair of cells  $(\sigma, \tau)$  that we add to the HDVF we have to find a  $\mathcal{V}$ -path between them and reverse it.

The core of Algorithm 1 lies at line 5. We now present three methods for updating the matrix  $D_{q+1}$  after adding a pair of critical cells  $(\sigma^{(q)}, \tau^{(q+1)})$  and we study their complexity.

Let us now point out an important aspect about the complexity. We denote by  $n$  the number of cells in the CW complex  $K$ . If  $K$  is a simplicial complex and  $D_q$  is its initial (not reduced) boundary matrix, then the number of non-zero entries in each column of  $D_q$  is  $q + 1$  (since a  $q$ -simplex has  $q + 1$  faces of dimension  $q - 1$ ). Also, if  $K$  is a cubical complex embedded in  $\mathbb{R}^d$  then each column of  $D_q$  has  $2q$  non-zero entries and

each row contains at most  $2(d - q)$  non-zero entries. Therefore, it is interesting not only to consider dense boundary matrices, but also sparse ones along their columns or along their rows and columns. We denote these three types of matrices *dense*, *d-bounded* and *dd\**-*bounded* respectively.

A common point between the following three methods is that we must remove the row of  $\tau$  in  $D_{q+2}$  and the column of  $\sigma$  in  $D_q$ .

*Method I: inverting H*

Given the equations for the reduction associated to a HDVF  $X$  (cf. [Theorem 5.2](#)), the most trivial way to update the boundary matrices is to invert the new matrix  $d(S'_{q+1})|_{P'_q}$  and to compute

$$D'_{q+1} = d(C'_{q+1})|_{C'_q} - d(S'_{q+1})|_{C'_q} \cdot (d(S'_{q+1})|_{P'_q})^{-1} \cdot d(C'_{q+1})|_{P'_q}$$

We estimate the complexity of this operation. Remark that all these matrices have at most  $n$  columns and  $n$  rows. Inverting the matrix  $d(S'_{q+1})|_{P'_q}$  can be done in matrix multiplication time, so it requires  $\mathcal{O}(n^\omega)$  operations, where  $\omega \leq 2.374$  [\[52\]](#).

In order to understand the complexity of this method in the context of the three types of boundary matrices, we introduce the following notation. A matrix  $A \in M_{n \times n}(\mathbb{Z})$  is called  $[r, c]$ -matrix if each row contains at most  $r$  non-zero entries and each column contains at most  $c$  non-zero entries. Thus, dense matrices are  $[n, n]$ -matrices, *d-bounded* matrices are  $[n, c]$ -matrices and *dd\**-*bounded* are  $[r, c]$ -matrices for some constants  $r$  and  $c$ . [Lemma A.6](#) states the complexity of the sum and the multiplication of  $[r, c]$ -matrices.

Let us suppose that the three matrices  $d(C'_{q+1})|_{C'_q}$ ,  $d(S'_{q+1})|_{C'_q}$  and  $d(C'_{q+1})|_{P'_q}$  are  $[r, c]$ -matrices, where  $K = P' \sqcup S' \sqcup C'$  denotes the new partition and  $D'$  is the new reduced boundary after adding  $(\sigma, \tau)$  to  $X$ . Thus we can obtain  $D'_{q+1}$  by performing  $\mathcal{O}(n^2 \cdot (c + r))$  operations:

$$\begin{aligned}
 D'_{q+1} &= \underbrace{d(C'_{q+1})|_{C'_q}}_{[r,c]} - \underbrace{d(S'_{q+1})|_{C'_q}}_{[r,c]} \cdot \underbrace{(d(S'_{q+1})|_{P'_q})^{-1}}_{[n,n]} \cdot \underbrace{d(C'_{q+1})|_{P'_q}}_{[r,c]} \\
 & \qquad \qquad \qquad [r, c] + ([r, c] \cdot [n, n]) \cdot [r, c] \\
 & \qquad \qquad \qquad [r, c] + [n, n] \cdot [r, c] \quad (n^2c \text{ operations}) \\
 & \qquad \qquad \qquad [r, c] + [n, n] \quad (n^2r \text{ operations}) \\
 & \qquad \qquad \qquad [n, n] \quad (n^2 \text{ operations})
 \end{aligned}$$

Consequently, if the boundary matrices are dense, *d-bounded* or *dd\**-*bounded*, the complexity of this method is  $\mathcal{O}(n^3)$ ,  $\mathcal{O}(n^3)$  and  $\mathcal{O}(n^{2.374})$  respectively.

*Method II: using the Banachiewicz formula for H*

Let us see how we can obtain  $(d(S)|_P)^{-1}$  without inverting the matrix. In the following we omit the subscripts whenever they are clear from the context.

**Proposition 6.3.** *After adding the pair of critical cells  $(\sigma, \tau)$ , the matrix  $(d(S')|_{P'})^{-1}$  can be obtained within  $\mathcal{O}(n^2)$  operations.*

**Proof.** We write

$$\begin{aligned}
 H_q &= [A]^{-1} & & =: H \\
 F_q &= - \begin{bmatrix} u \\ B \end{bmatrix} \cdot H_q = - \begin{bmatrix} uH \\ BH \end{bmatrix} & & =: \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix}
 \end{aligned}$$

$$\begin{aligned}
G_{q+1} &= -H_q \cdot [v \mid C] = -[Hv \mid HC] & =: [G_{11} \mid G_{12}] \\
D_{q+1} &= -\left[\frac{uH}{BH}\right] \cdot [v \mid C] + \left[\frac{w \mid s}{r \mid E}\right] = \\
&= \left[\frac{-uHv + w \mid -uHC + s}{-BHv + r \mid -BHC + E}\right] & =: \left[\frac{D_{11} \mid D_{12}}{D_{21} \mid D_{22}}\right]
\end{aligned}$$

where

$$\begin{aligned}
A &= d(S_{q+1})|_{P_q} & r &= d(\tau)|_{C'_q} \\
u &= d(S_{q+1})|_{\sigma} & B &= d(S_{q+1})|_{C'_q} & s &= d(C'_{q+1})|_{\sigma} \\
v &= d(\tau)|_{P_q} & C &= d(C'_{q+1})|_{P_q} & E &= d(C'_{q+1})|_{C'_q} \\
w &= d(\tau)|_{\sigma}
\end{aligned}$$

Remark that  $D_{11} = \langle d'(\tau), \sigma \rangle$  is a unit, so  $D_{11}^{-1}$  exists. By the Banachiewicz identity (cf. [Lemma A.4](#)),

$$\begin{aligned}
H'_q &= \left[\frac{A \mid v}{u \mid w}\right]^{-1} \\
&= \left[\frac{H + HvD_{11}^{-1}uH \mid -HvD_{11}^{-1}}{-D_{11}^{-1}uH \mid D_{11}^{-1}}\right] \\
&= \left[\frac{H + G_{11}D_{11}^{-1}F_{11} \mid G_{11}D_{11}^{-1}}{D_{11}^{-1}F_{11} \mid D_{11}^{-1}}\right]
\end{aligned}$$

The complexity of this method is dominated by the computation of the upper-left block, which requires  $\mathcal{O}(n^2 + n(c+r)) = \mathcal{O}(n^2)$  operations:

$$\begin{aligned}
&H + HvD_{11}^{-1}uH \\
&[n, n] + ([n, n] \cdot [1, c]) \cdot ([r, 1] \cdot [n, n]) \\
&[n, n] + [1, n] \cdot [n, 1] \quad (nc + nr \text{ operations}) \\
&[n, n] + [n, n] \quad (n^2 \text{ operations}) \\
&[n, n] \quad (n^2 \text{ operations}) \quad \square
\end{aligned}$$

Thus, the boundary matrices can be obtained within  $\mathcal{O}(n^2 + n^2(c+r)) = \mathcal{O}(n^2(c+r))$  operations. Consequently, if the boundary matrices are dense,  $d$ -bounded or  $dd^*$ -bounded, the complexity of this method is  $\mathcal{O}(n^3)$ ,  $\mathcal{O}(n^3)$  and  $\mathcal{O}(n^2)$  respectively. This means that this method is theoretically better for a cubical complex.

*Method III: continuing that idea*

Following the notation in [Proposition 6.3](#), we can directly obtain  $D_{q+1}$  without computing  $H_q$ .

**Proposition 6.4.** *The matrix  $D_{q+1}$  can be obtained within  $\mathcal{O}(n^2)$  operations.*



**Proof.** Following the notation of the proof of Proposition 6.3,

$$\begin{aligned}
 D'_{q+1} &= d(C'_{q+1})|_{C'_q} - d(S'_{q+1})|_{C'_q} \cdot (d(S'_{q+1})|_{P'_q})^{-1} \cdot d(C'_{q+1})|_{P'_q} \\
 &= E - \left[ \begin{array}{c|c} B & r \end{array} \right] \left[ \begin{array}{c|c} H + HvD_{11}^{-1}uH & -HvD_{11}^{-1} \\ \hline -D_{11}^{-1}uH & D_{11}^{-1} \end{array} \right] \left[ \begin{array}{c} C \\ \hline s \end{array} \right] \\
 &= E - BHC - BHvD_{11}^{-1}uHC + rD_{11}^{-1}uHC + BHvD_{11}^{-1}s - rD_{11}^{-1}S \\
 &= (E - BHC) - (r - BHv)D_{11}^{-1}(s - uHC) \\
 &= D_{22} - D_{21}D_{11}^{-1}D_{12}
 \end{aligned}$$

Consequently,  $D'_{q+1}$  can be obtained within  $\mathcal{O}(n^2)$  operations.  $\square$

Table 1 compares the three methods against the three possible types of boundary matrices.

**Table 1**

Comparison of the three methods.

	Method I	Method II	Method III
Dense	$n^3$	$n^3$	$n^2$
$d$ -bounded	$n^3$	$n^3$	$n^2$
$dd^*$ -bounded	$n^{2.373}$	$n^2$	$n^2$

Thus the third method outperforms the two others for each type of boundary matrix. We can now formulate the complexity of Algorithm 1.

**Theorem 6.5.** Algorithm 1 can be computed within  $\mathcal{O}(n^3)$  operations.

**Proof.** Let us consider the worst case in which we found a perfect HDVF and no critical cell remains (which is impossible since there is at least one connected component). Thus we add  $n/2$  pairs of cells. Finding a pair of cells consists in choosing a unit in the boundary matrices, so it can be done within  $\mathcal{O}(n^2)$  operations. Then, by using the third method, the complexity of the algorithm is  $\mathcal{O}(\frac{n}{2}(n^2 + n^2)) = \mathcal{O}(n^3)$ .  $\square$

Note that this result does not depend (and does not take advantage) of the boundary matrices type. Computing also the DVF does not affect the complexity of the algorithm theoretically. We can find and reverse a  $\mathcal{V}$ -path in  $\mathcal{O}(n^2)$  time, so obtaining the DVF associated to the HDVF requires also at most  $\mathcal{O}(n^3)$  operations.

In Algorithm 1 we do not propose any rule for choosing the pair of critical cells. It could be convenient to choose a pair  $(\sigma, \tau)$  such that  $D_{12} = 0$  or  $D_{21} = 0$ , so updating the reduced boundary  $D$  is simpler. This corresponds to an elementary reduction ( $\tau$  is the only coface of  $\sigma$ ) or an elementary coreduction ( $\sigma$  is the only face of  $\tau$ ) [19]. It has been noted that it is preferable to look for elementary coreductions than for elementary reductions [20,39,35,53].

### 6.2. Computing also the reduction

At some point, it may be interesting to obtain the reduction induced by the HDVF. We have seen with the second method that it is better not to invert  $d(S)|_P$ , so  $H$  may be computed at each step using the formula in Proposition 6.3. Then,  $F$  and  $G$  may be computed at the end of the algorithm using the formula in Theorem 5.2, which needs  $\mathcal{O}(n^3)$  operations if the boundary matrices are not  $dd^*$ -bounded. However, if we want to know the operators  $f$  and  $g$  throughout the computation of the HDVF it is better to use the following result.

**Proposition 6.6.** *After adding the pair of critical cells  $(\sigma^{(q)}, \tau^{(q+1)})$  to a HDVF, the matrices  $F'_q$  and  $G'_{q+1}$  can be obtained within  $\mathcal{O}(n^2)$  operations.*

**Proof.** Using the notation introduced in Proposition 6.3, it is easy to prove that

$$\begin{aligned} F'_q &= - \left[ B \mid r \right] \cdot H'_q \\ &= \left[ F_{21} - D_{21}D_{11}^{-1}F_{11} \mid -D_{21}D_{11}^{-1} \right] \\ G'_{q+1} &= -H'_q \cdot \begin{bmatrix} C \\ s \end{bmatrix} \\ &= \begin{bmatrix} G_{12} - G_{11}D_{11}^{-1}D_{12} \\ -D_{11}^{-1}D_{12} \end{bmatrix} \end{aligned}$$

Thus, we can update  $f$  and  $g$  within  $\mathcal{O}(n^2)$  operations.  $\square$

Consequently, a HDVF and its reduction can be computed within  $\mathcal{O}(n^3)$  operations.

### 6.3. Some questions about the algorithm

We partially answer some questions concerning Algorithm 1 in this section. For each question, we treat separately the case where  $\mathfrak{R}$  is a field and where it is not.

*Question 1: does Algorithm 1 return a perfect HDVF?*

If  $\mathfrak{R}$  is a field, the answer is yes.

**Proposition 6.7.** *Let  $K$  be a CW complex. Then Algorithm 1 returns a perfect HDVF whenever the ring of coefficients is a field.*

**Proof.** Note that, if  $\mathfrak{R}$  is a field, every non-zero element is a unit. Thus, Algorithm 1 only stops when the reduced boundary matrices are zero, in which case the returned HDVF is perfect.  $\square$

This means that we can always obtain a perfect reduction using a HDVF, even for complexes like the Bing’s house or the dunce hat, which do not admit a perfect DGVF. Hence, Proposition 6.7 is a fundamental property of HDVFs.

If  $\mathfrak{R}$  is not a field, the answer is no. First of all, we recall that a CW complex with a torsion subgroup in one of its homology groups does not admit a perfect HDVF, since a perfect reduction involves homology groups of the form  $\mathfrak{R}^\beta$ .

Moreover, let us show that Algorithm 1 can return a non-perfect HDVF even when the homology groups are torsion-free. We consider the ring of coefficients  $\mathfrak{R} = \mathbb{Z}$  and the chain complex

$$0 \xrightarrow{0} A \xrightarrow{d} B \xrightarrow{0} 0$$

where  $A = \langle a_1, \dots, a_5 \rangle \cong \mathbb{Z}^5$ ,  $B = \langle b_1, \dots, b_5 \rangle \cong \mathbb{Z}^5$  and the linear operator  $d$  is defined by the matrix

$$\begin{bmatrix} 1 & 1 & -1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & -1 \\ -1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

If [Algorithm 1](#) adds the pairs  $(b_4, a_1)$ ,  $(b_3, a_5)$  and  $(b_5, a_4)$  then it stops since the reduced boundary matrix is

$$D = \begin{bmatrix} 4 & -3 \\ 3 & 0 \end{bmatrix}$$

and it does not contain any unit. However, if [Algorithm 1](#) adds the pairs  $(b_1, a_1)$ ,  $(b_2, a_4)$ ,  $(b_5, a_3)$ ,  $(b_3, a_5)$  and  $(b_4, a_2)$  then it does find a perfect HDVF. Nevertheless, this counterexample consists of a chain complex, so the question remains open for simplicial or cubical complexes.

*Question 2: can [Algorithm 1](#) compute every HDVF?*

If  $\mathfrak{R}$  is a field, the answer is yes.

**Proposition 6.8.** *Any HDVF can be obtained with [Algorithm 1](#) whenever the ring of coefficients is a field.*

**Proof.** Let  $X$  be a HDVF. Using the Laplace expansion for the determinant along the first row of  $A = d(S)|_P$ , we get  $\det(d(S)|_P) = \sum_j \lambda_{1,j} \det(A_{1,j}) \neq 0$ . Thus there exists some  $j$  such that  $\det(A_{1,j}) \neq 0$ , so  $X$  can be obtained by adding a pair  $(\sigma, \tau)$  to a smaller HDVF. The result follows from induction over the size of the matrix  $d(S)|_P$ .  $\square$

[Proposition 6.8](#) proves that [Algorithm 1](#) not only computes a perfect HDVF, but any possible perfect HDVF over a given CW complex. Hence, if we are looking for a particular perfect HDVF (possibly with some property on its associated homology or cohomology generators), we are sure that we can find it by choosing the pairs of critical cells in the correct order.

If  $\mathfrak{R}$  is not a field, the answer is no. We provide again a counterexample. Let  $\mathfrak{R} = \mathbb{Z}$  and consider the chain complex  $0 \xrightarrow{0} \mathbb{Z}^6 \xrightarrow{d} \mathbb{Z}^6 \xrightarrow{0} 0$  where the linear operator  $d$  is defined by the matrix

$$\begin{bmatrix} -1 & 1 & 0 & 1 & -1 & 1 \\ 1 & 0 & 1 & -1 & 0 & -1 \\ 1 & 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

One can prove by exhaustion over all the possible sequences of pairs  $(b_i, a_j)$  that [Algorithm 1](#) never finds the unique perfect HDVF, which contains all the elements of the bases. Hence, this is also a counterexample for the first question.

*Question 3: existence of a perfect HDVF*

If  $\mathfrak{R}$  is a field, there exists a perfect HDVF for every CW complex since [Algorithm 1](#) always returns one.

If  $\mathfrak{R}$  is not a field, the answer is unknown. The two first questions have been partially answered, since we do not have any counterexample involving a simplicial or cubical complex with  $\mathfrak{R} = \mathbb{Z}$ . Indeed, they are strongly related to the existence of a perfect HDVF. If a CW complex does not admit a perfect HDVF, the first question has a negative answer. On the other hand, if a CW complex admits a perfect HDVF and the second question has a positive answer, then [Algorithm 1](#) can find a perfect HDVF, though it may not find it always.

We have already seen that a CW complex whose homology groups have torsion coefficients does not admit a perfect HDVF. In addition, as a consequence of [Proposition 6.7](#), every CW complex admits a perfect HDVF whenever  $\mathfrak{R}$  is a field. Nevertheless, we have not addressed what happens when  $\mathfrak{R} = \mathbb{Z}$  and the homology groups are torsion-free. In order to find a counterexample, we executed [Algorithm 1](#) for all the torsion-free simplicial complexes in Benedetti and Lutz’s library of triangulations [54] and we always

found a perfect HDVF. Moreover, the HDVFs returned for the simplicial complexes with just one torsion coefficient per dimension (i.e., `Hom_C5_K4`, `RP4`, `RP4#K3_17`, `RP4#11S2xS2` and `RP5_24`) had their reduced boundary matrix already in SNF. Hence, even if they are not perfect HDVFs, the homology groups can be directly read from them.

We point out that the simplicial complex `hyperbolic_dodecahedral_space` presented an interesting behavior. Its 1-dimensional homology group is  $H_1 = (\mathbb{Z}_5)^3$ . Due to its small size (718 cells), we executed [Algorithm 1](#) 500 000 times with random choices of pairs of cells and we only found 72 HDVFs whose reduced boundary matrices were in SNF. The other simplicial complex with more than one torsion coefficient, `PG128_PG128P7`, is much larger (13 462 cells) and we still have not found any HDVF whose reduced boundary matrix is in SNF.

#### 6.4. Another algorithm for computing a HDVF

[Algorithm 1](#) consists in iteratively adding a pair of critical cells to the HDVF. Nevertheless, we can also add several pairs of cells to a HDVF at the same time.

Let  $X$  be a HDVF and  $\Sigma = \{\sigma_1, \dots, \sigma_r\}$  and  $T = \{\tau_1, \dots, \tau_r\}$  be two sets of critical cells of codimension 1 (that is,  $\dim(\sigma_i) = \dim(\tau_i) - 1$ ). If the matrix  $d'(T)|_{\Sigma}$  is invertible in  $\mathfrak{R}$  then  $X' = (P \cup \Sigma, S \cup T)$  is a HDVF. The proof is similar to that of [Proposition 6.1](#).

As a consequence, [Algorithm 1](#) is not the unique way of computing a HDVF. However, we prefer it for its simplicity and we do not study in this work the above alternative approach.

Let us point out that, if we can add several pairs of cells at the same time, then the second question of the previous section is true since we can add all the pairs of cells in a HDVF at once.

## 7. Deforming a HDVF

In [Section 6](#) we described how the reduction changes after adding a pair of critical cells to the HDVF. This can be seen as a basic operation on a HDVF, in which two critical cells  $\gamma$  and  $\gamma'$  are transformed into a primary and a secondary cell respectively. In this section we extend this idea to define five basic operations that allow us to modify a HDVF.

### 7.1. Basic operations

Let  $K$  be a CW complex endowed with a HDVF  $X = (P, S)$ . Let  $\sigma \in P$ ,  $\tau \in S$  and  $\gamma, \gamma' \in C$ . Thus,

- $X' = \mathbf{A}(X, \gamma, \gamma') = (P \cup \{\gamma\}, S \cup \{\gamma'\})$  is a HDVF identical to  $X$  except for  $\gamma$ , which is a primary cell, and  $\gamma'$ , which is a secondary cell
- $X' = \mathbf{R}(X, \sigma, \tau) = (P \setminus \{\sigma\}, S \setminus \{\tau\})$  is a HDVF identical to  $X$  except for  $\sigma$  and  $\tau$ , which are critical cells
- $X' = \mathbf{M}(X, \sigma, \gamma) = ((P \setminus \{\sigma\}) \cup \{\gamma\}, S)$  is a HDVF identical to  $X$  except for  $\sigma$ , which is a critical cell, and  $\gamma$ , which is a primary cell
- $X' = \mathbf{W}(X, \tau, \gamma) = (P, (S \setminus \{\tau\}) \cup \{\gamma\})$  is a HDVF identical to  $X$  except for  $\tau$ , which is a critical cell, and  $\gamma$ , which is a secondary cell
- $X' = \mathbf{MW}(X, \sigma, \tau) = ((P \setminus \{\sigma\}) \cup \{\tau\}, (S \setminus \{\tau\}) \cup \{\sigma\})$  is a HDVF identical to  $X$  except for  $\tau$ , which is a primary cell, and  $\sigma$ , which is a secondary cell

The operation **A** has been largely explained in [Section 6](#) and **R** consists in removing a pair of cells from the HDVF. **M** exchanges a primary cell with a critical one, while **W** exchanges a secondary cell with a critical one. **MW** is like combining **M** and **W** except that no critical cell is needed.

Let us see the conditions under which we can perform each operation.

**Proposition 7.1.** *Let  $K$  be a CW complex endowed with a HDVF  $X$ . Let  $\sigma \in P$ ,  $\tau \in S$  and  $\gamma, \gamma' \in C$ . Thus,*

1.  $\mathbf{A}(X, \gamma, \gamma')$  is a HDVF if  $\langle d'(\gamma'), \gamma \rangle$  is a unit
2.  $\mathbf{R}(X, \sigma, \tau)$  is a HDVF if  $\langle h(\sigma), \tau \rangle$  is a unit
3.  $\mathbf{M}(X, \sigma, \gamma)$  is a HDVF if  $\langle f(\sigma), \gamma \rangle$  is a unit
4.  $\mathbf{W}(X, \tau, \gamma)$  is a HDVF if  $\langle g(\gamma), \tau \rangle$  is a unit
5.  $\mathbf{MW}(X, \sigma, \tau)$  is a HDVF if  $\langle dh(\sigma), \tau \rangle$  and  $\langle hd(\tau), \sigma \rangle$  are units

**Proof.** The first statement only rephrases [Proposition 6.1](#).

For the second statement we need to prove that  $d_q(S'_{q+1})|_{P'_q}$  is invertible after removing the two cells. In the following we omit the subscripts. We write

$$d(S)|_P = \left[ \begin{array}{c|c} A & B \\ \hline C & d(S')|_{P'} \end{array} \right], \quad M = \left[ \begin{array}{c|c} 1 & 0 \\ \hline C & d(S')|_{P'} \end{array} \right]$$

where  $A = d(\tau)|_\sigma$ ,  $B = d(S')|_\sigma$  and  $C = d(\tau)|_{P'}$ . Note that  $\det(M) = \det(d(S')|_{P'})$ . Then

$$\begin{aligned} \det(d(S')|_{P'}) &= \det(M) \\ &= \det\left(d(S)|_P + \left[ \begin{array}{c|c} 1-A & -B \\ \hline 0 & 0 \end{array} \right]\right) \\ &= \det\left(d(S)|_P + \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] \cdot \left[ \begin{array}{c|c} 1-A & -B \end{array} \right]\right) \\ &= \det(d(S)|_P) \cdot \left(1 + \left[ \begin{array}{c|c} 1-A & -B \end{array} \right] \cdot H \cdot \left[ \begin{array}{c} 1 \\ 0 \end{array} \right]\right) \quad (\text{cf. Lemma A.7}) \\ &= \det(d(S)|_P) \cdot \left(1 + \left[ \begin{array}{c|c} 1 & 0 \end{array} \right] \cdot H \cdot \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] - \left[ \begin{array}{c|c} A & B \end{array} \right] \cdot H \cdot \left[ \begin{array}{c} 1 \\ 0 \end{array} \right]\right) \\ &= \det(d(S)|_P) \cdot (1 + H_{11} - 1) = \det(d(S)|_P) \cdot H_{11} \end{aligned}$$

where  $H_{11}$  denotes  $h(\sigma)|_\tau = \langle h(\sigma), \tau \rangle$ .

The third statement is also proved using [Lemma A.7](#). We write

$$d_q(S)|_P = \left[ \begin{array}{c} a \\ \hline M \end{array} \right], \quad d_q(S)|_{P'} = \left[ \begin{array}{c} b \\ \hline M \end{array} \right]$$

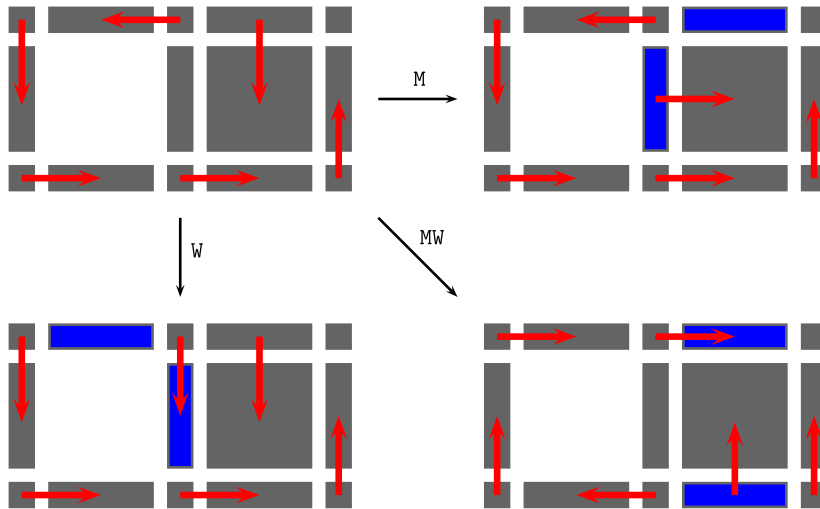
where  $a = d(S)|_\sigma$  and  $b = d(S)|_\gamma$ . We note that

$$F = - \left[ \begin{array}{c} b \\ \hline N \end{array} \right] \cdot H$$

where  $N = d(S)|_{C \setminus \gamma}$  and thus

$$\langle f(\sigma), \gamma \rangle = -b \cdot h$$

where  $h = h(\sigma)|_S$ . Then



**Fig. 7.** A HDVF on a cubical complex and the result after applying M, W and MW. Blue cells are those which are exchanged. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\begin{aligned}
 \det(d_q(S)|_{P'}) &= \det\left(d_q(S)|_P + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot (b - a)\right) \\
 &= \det(d_q(S)|_{P'}) \cdot \left(1 + (b - a) \cdot H \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \\
 &= \det(d_q(S)|_{P'}) \cdot (1 + (b - a) \cdot h) \\
 &= \det(d_q(S)|_{P'}) \cdot (1 + b \cdot h - a \cdot h) \\
 &= \det(d_q(S)|_{P'}) \cdot (1 - \langle f(\sigma), \gamma \rangle - 1) \\
 &= -\det(d_q(S)|_{P'}) \cdot \langle f(\sigma), \gamma \rangle
 \end{aligned}$$

$d_q(S)|_{P'}$  is thus invertible.

We omit the proof of the last two statements since they are similar to the third one.  $\square$

All these operations can be applied in terms of the DVF by reversing a  $\mathcal{V}$ -path between the two cells considered. This  $\mathcal{V}$ -path is not unique, but its existence can be proved using the same argument present in Proposition 6.2. In the case of MW, there are two  $\mathcal{V}$ -paths to reverse. Fig. 7 illustrates the operations M, W and MW on a cubical complex.

Some of these operations were introduced in [34]. Namely, the *arrow reversing* is the operation M between 0-cells, the *edge rotation* is MW between 1-cells and the *face rotation* is MW between 2-cells. These three operations were announced as local deformations of a DGVF but, since no condition was given, this is in general false: applying an *edge rotation* or a *face rotation* to a DGVF can produce a non-gradient discrete vector field. Only the *arrow reversing* preserved the structure of DGVF since, in dimension 0, the existence of a  $\mathcal{V}$ -path between a primary cell  $\sigma$  and a critical cell  $\gamma$  implies that  $\langle f(\sigma), \gamma \rangle$  is a unit.

### 7.2. Delineating (co)homology generators

Given a perfect HDVF, the operations M, W and MW are interesting since they change the reduction and thus the generators of the homology and the cohomology groups. The next proposition specifies how a generator changes when the operations M or W affect its associated critical cell.

**Proposition 7.2.** *Let  $K$  be a CW complex endowed with a perfect HDVF  $X = (P, S)$  with  $\mathfrak{R} = \mathbb{Z}_2$ . Let  $\sigma \in P$ ,  $\tau \in S$  and  $\gamma \in C$ . Then,*

1. *If  $\langle f(\sigma), \gamma \rangle$  is a unit, the cohomology generators associated to  $\gamma$  in  $X$  and  $\sigma$  in  $\mathbb{M}(X, \sigma, \gamma)$  are the same.*
2. *If  $\langle g(\gamma), \tau \rangle$  is a unit, the homology generators associated to  $\gamma$  in  $X$  and  $\tau$  in  $\mathbb{W}(X, \tau, \gamma)$  are the same.*

**Proof.** The proof of these statements is quite lengthy, but it provides a partial description of the reduction after perturbing the HDVF.

For the first statement we write

$$\begin{aligned}
 H_q &= \left[ \begin{array}{c} u \\ A \end{array} \right]^{-1} && =: \left[ H_1 \mid H_2 \right] \\
 F_q &= - \left[ \begin{array}{c} v \\ B \end{array} \right] \cdot H_q = - \left[ \begin{array}{c|c} vH_1 & vH_2 \\ \hline BH_1 & BH_2 \end{array} \right] && =: \left[ \begin{array}{c|c} F_{11} & F_{12} \\ \hline F_{21} & F_{22} \end{array} \right]
 \end{aligned}$$

where  $u = d(S)|_\sigma$ ,  $A = d(S)|_{P \setminus \sigma}$ ,  $v = d(S)|_\tau$  and  $B = d(S)|_{C \setminus \gamma}$

Then,

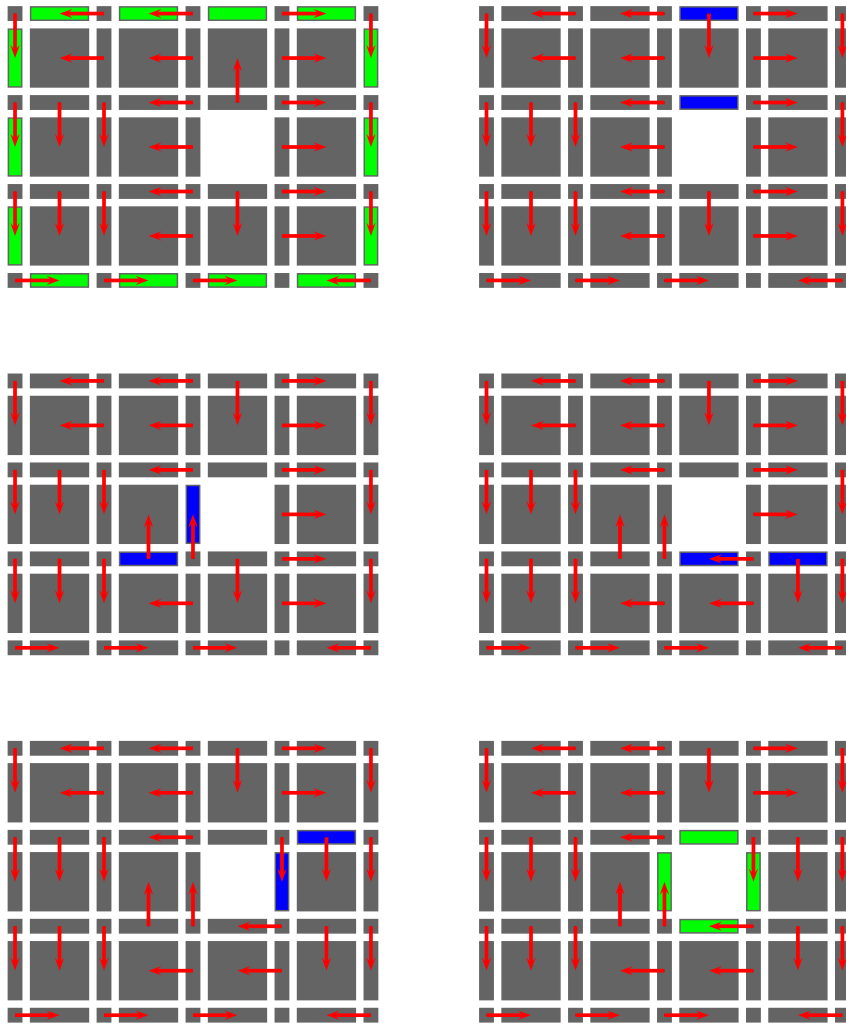
$$\begin{aligned}
 H' &= (d(S)|_{P'})^{-1} \\
 &= \left( d(S)_P + \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] \cdot (v - u) \right)^{-1} \quad (\text{cf. Lemma A.5}) \\
 &= H + F_{11}^{-1} \left( H \cdot \left[ \begin{array}{c} 1 \\ 0 \end{array} \right] \cdot (v - u) \cdot H \right) \\
 &= H \cdot \left( I - F_{11}^{-1} \left[ \begin{array}{c|c} F_{11} & F_{12} \\ \hline 0 & 0 \end{array} \right] - F_{11}^{-1} \left[ \begin{array}{c|c} 1 & 0 \\ \hline 0 & 0 \end{array} \right] \right) \\
 &= H \cdot \left[ \begin{array}{c|c} -F_{11}^{-1} & -F_{11}^{-1}F_{12} \\ \hline 0 & I \end{array} \right] \\
 &= \left[ -H_1F_{11}^{-1} \mid H_2 - H_1F_{11}^{-1}F_{12} \right]
 \end{aligned}$$

Consequently,

$$\begin{aligned}
 F' &= - \left[ \begin{array}{c} u \\ B \end{array} \right] \cdot \left[ -H_1F_{11}^{-1} \mid H_2 - H_1F_{11}^{-1}F_{12} \right] \\
 &= \left[ \begin{array}{c|c} F_{11}^{-1} & F_{11}^{-1}F_{12} \\ \hline F_{21}F_{11}^{-1} & F_{22} - F_{21}F_{11}^{-1}F_{12} \end{array} \right]
 \end{aligned}$$

The proof of the second statement is similar. We write

$$\begin{aligned}
 H_q &= \left[ u \mid A \right]^{-1} && =: \left[ \begin{array}{c} H_1 \\ H_2 \end{array} \right] \\
 G_{q+1} &= -H_q \cdot \left[ v \mid B \right] = - \left[ \begin{array}{c|c} H_1v & H_1B \\ \hline H_2v & H_2B \end{array} \right] && =: \left[ \begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right]
 \end{aligned}$$



**Fig. 8.** Example of multiple applications of the operations on a cubical complex. Blue cells are those that have changed. The one-dimensional homology generator is depicted in green at the beginning and at the end. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where  $u = d(\tau)|_P$ ,  $A = d(S \setminus \tau)|_P$ ,  $v = d(\gamma)|_P$  and  $B = d(C \setminus \gamma)|_P$ . Then it is easy to prove that

$$H' = \begin{bmatrix} -G_{11}^{-1}H_1 \\ H_2 - G_{11}^{-1}G_{21}H_1 \end{bmatrix}$$

$$G' = \left[ \begin{array}{c|c} G_{11}^{-1} & G_{11}^{-1}G_{12} \\ \hline G_{11}^{-1}G_{21} & G_{22} - G_{11}^{-1}G_{21}G_{12} \end{array} \right] \quad \square$$

We can thus use these operations to change the shape of the generators. Fig. 8 shows a cubical complex endowed with a HDVF. We want to have a one-dimensional homology generator around the hole. For doing this, it suffices that all the 1-cells are secondary except for one which is critical. Thus, we use  $M$  on the top 1-cell to put there the critical cell. Then, for the other three 1-cells, we use  $MW$  to make them secondary. At the end, the homology generator induced by the HDVF stands at the desired location.

It is unclear whether this application is computationally feasible. The problem is: given a perfect HDVF  $X$  and a set of cycles  $S$ , can we find a perfect HDVF  $X'$  whose homology generators contain this set? We may first check that the cycles are linearly independent. This can be done by computing the rank of the



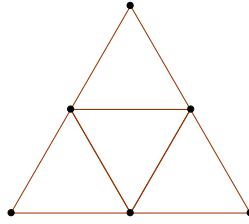


Fig. 9. There exists no HDVF on this simplicial complex whose homology generators are the four triangles consisting of three 1-cells.

matrix  $f(S)$ . If the rank is maximal, then the cycles can be part of a homology basis. But even if the cycles are linearly independent, the HDVF  $X'$  does not exist in general, and Fig. 9 provides a counterexample. Thus, this problem must be studied further in order to find conditions under which such HDVF exists. A possible hint to follow is that every cycle must have a cell not included in any other cycle, which is the intuition that led to our counterexample.

Assuming that the HDVF  $X'$  exists, it is possible to find a sequence of operations that transform one HDVF into the other: it suffices to successively apply **R** to  $X$  for removing all the pairing in the DVF, and then build the other HDVF using **A** (this is guaranteed if  $\mathfrak{R}$  is a field by Proposition 6.8). Thus, the interesting question is to find a minimal sequence of operations that transform  $X$  into  $X'$ .

### 7.3. Connectivity between HDVFs

The new definitions let us state that Algorithm 1 can compute any HDVF which is the result of applying only the operation **A** to an empty HDVF. We explained in Section 6.3 that we have not addressed if every HDVF on a simplicial or cubical complex can be found through Algorithm 1. Thus it is natural to wonder if any HDVF on a simplicial or cubical complex can be obtained by a sequence of operations (not only **A**) on an empty HDVF. This can also be formulated as follows: are all the HDVF on a simplicial or cubical complex connected via a sequence of operations? This is still an open question.

## 8. Relation with other methods in computational homology

There are several methods for computing homology in the literature which seem to be equivalent. The simple formulation of the HDVFs allows us to clearly see these equivalences.

### 8.1. Iterated Morse decomposition

First, let us prove that the HDVF generalizes the notion of DGVF.

**Proposition 8.1.** *Every DGVF is a HDVF.*

**Proof.** We need to prove that the matrices  $d(S_{q+1})|_{P_q}$  are invertible for each  $q \geq 0$ . In the following we omit the subscripts since the proof is the same for every dimension  $q$ .

Let  $\mathcal{V} = \{(\sigma_i, \tau_i)\}_{i=1}^m$  be a DGVF. Consider the weighted digraph whose vertices are the primary cells and where arrows connect two vertices whenever there is a  $\mathcal{V}$ -path of length 1 between them. Formally,  $(\sigma_i, \sigma_j)$  is an arrow if  $\langle d(\tau_i), \sigma_j \rangle \neq 0$  and  $\sigma_i \neq \sigma_j$ . Its weight is the value  $-\langle d(\tau_i), \sigma_i \rangle \cdot \langle d(\tau_i), \sigma_j \rangle$ . It is immediate to see that the matrix associated to this graph is  $I - d(S)|_{P}V$ , where  $S = \{\tau_i\}_{i=1}^m$ ,  $P = \{\sigma_i\}_{i=1}^m$  and the diagonal matrix  $V = (v_{i,j})$  is such that for every  $i$ ,  $v_{i,i} = \langle d(\tau_i), \sigma_i \rangle$  and zero elsewhere. Note that  $V$  is invertible. Since there are no closed  $\mathcal{V}$ -paths in the DGVF, the matrix  $I - d(S)|_{P}V$  is nilpotent, and thus  $d(S)|_{P}V$  is invertible. As  $V$  is invertible, we deduce that  $d(S)|_{P}$  is invertible.  $\square$

A more elaborate tool for computing the homology groups using discrete Morse theory is the iterated Morse decomposition [47], which consists in iteratively: (1) computing a DGVF and (2) considering the resulting Morse complex for a next DGVF. We prove now that every iterated Morse decomposition is also a HDVF.

**Proposition 8.2.** *Every iterated Morse decomposition is a HDVF.*

**Proof.** For clarity, we assume that the iterated Morse decomposition consists of only two DGVFs  $\mathcal{V}^1$  and  $\mathcal{V}^2$ . We recall that the second DGVF is defined on the chain complex consisting of the critical cells of  $\mathcal{V}^1$  and the boundary operator

$$d' = d(C^1)|_{C^1} - d(S^1)|_{C^1} \cdot H \cdot d(C^1)|_{P^1}.$$

If we write

$$d(S^1 \cup S^2)|_{P^1 \cup P^2} = \left[ \begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]$$

then

$$d(S^1)|_{P^1} = A \quad \text{and} \quad d'(S^2)|_{P^2} = D - CA^{-1}B$$

Given the previous proposition, these two matrices are invertible. Thus, by Lemma A.3,

$$\det(d(S^1 \cup S^2)|_{P^1 \cup P^2}) = \det(d(S^1)|_{P^1}) \cdot \det(d'(S^2)|_{P^2})$$

which is a unit.  $\square$

It is easy to see that if a HDVF has been created using Algorithm 1, then the list of pairs of cells  $[(\sigma_i, \tau_i)]_{i=1}^m$  is an iterated Morse decomposition. As we showed in Section 6, it is not true in general that every HDVF can be computed with Algorithm 1, so we cannot deduce that every HDVF is an iterated Morse decomposition. However, this does not mean that it is false. This question remains open.

## 8.2. The Smith normal form

The classic algorithm for computing homology groups computes the Smith normal form (SNF) [16]. We prove in this section that the reduced boundary matrices obtained in Algorithm 1 are similar to the diagonalization performed for the computation of the SNF.

Let  $K$  be a CW complex and  $X$  a trivial HDVF ( $P = S = \emptyset$ ). Let us choose some pivot in some boundary matrix  $D_q$ . For simplicity, we assume that the pivot is the element  $D_q(1, 1) = \langle d_q(\tau), \sigma \rangle$  and we omit the subscript. In order to make all the other entries in its row and column into zeros we perform

$$\begin{aligned} \forall j \neq 1, \quad D(\cdot, j) &\leftarrow D(\cdot, j) - D(1, j)D(1, 1)^{-1}D(\cdot, 1) \\ \forall i \neq 1, \quad D(i, \cdot) &\leftarrow D(i, \cdot) - D(i, 1)D(1, 1)^{-1}D(i, \cdot) \end{aligned}$$

Using the notation of Proposition 6.3, this is equivalent to

$$D' = D - \begin{bmatrix} D_{11} \\ D_{21} \end{bmatrix} D_{11}^{-1} \begin{bmatrix} 0 & D_{12} \end{bmatrix}$$

$$D'' = D' - \left[ \frac{0}{D'_{21}} \right] D_{11}^{-1} \left[ D'_{11} \mid D'_{12} \right]$$

By developing both equations we obtain that the pseudo-diagonalized boundary matrix is

$$\left[ \begin{array}{c|c} D_{11} & 0 \\ \hline 0 & D_{22} - D_{21}D_{11}^{-1}D_{12} \end{array} \right],$$

where the bottom-right block is the reduced boundary computed in Algorithm 1 after inserting the pair of cells  $(\sigma, \tau)$ .

**Proposition 8.3.** *Let  $K$  be a CW complex. Then, Algorithm 1 performs a partial diagonalization of the boundary matrices of  $K$ .*

**Proof.** The proof is direct from the previous argument.  $\square$

We have just seen that computing a HDVF is equivalent to compute the SNF of the boundary matrices using only the pivot operation, that is, given an invertible entry in the matrix, we make all the entries in its row and column into zeros. Computing the SNF needs also another type of operation: if there is no entry dividing all the others, we make elementary operations on the rows and columns so such an entry appears. For this reason, Algorithm 1 cannot always return a perfect HDVF if  $\mathfrak{R} = \mathbb{Z}$ , since in the computation of the SNF we can arrive to a matrix without units even if the SNF contains only units in its diagonal (see Section 6.3 for an example).

### 8.3. Persistent homology

Proposition 8.3 also implies that persistent homology can be computed with a variation of Algorithm 1. The classical algorithm for persistent homology [55, §4.2] is based on the Smith normal form. The main difference with a standard homology computation is that cells are considered in the order given by the filtration. Therefore, Algorithm 2 computes the persistence intervals of a filtration using the same calculations as Algorithm 1.

---

**Algorithm 2:** Compute a HDVF associated to a filtration.

---

**Input:** A CW complex  $K$ , a filtration  $\text{deg}$  and an ordering  $\{\sigma^j\}_{j=1}^n$

**Output:** The persistence intervals of  $\text{deg}$

**for**  $k = 0$  **to**  $\dim(K)$  **do**

$L_k \leftarrow \emptyset;$

$X \leftarrow (\emptyset, \emptyset);$

**for**  $j = 1$  **to**  $n$  **do**

**if**  $d'(\sigma^j) \neq 0$  **then**

$i \leftarrow \max \{j' : \langle d'(\sigma^j), \sigma^{j'} \rangle = 1\};$

$X \leftarrow \mathbf{A}(X, \sigma^i, \sigma^j)$  (and update the boundary matrices  $D$ );

$L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \{(\text{deg}(\sigma^i), \text{deg}(\sigma^j))\};$

**for**  $j = 1$  **to**  $n$  **do**

**if**  $\sigma^j$  **is critical** **then**

$L_{\dim(\sigma^j)} \leftarrow L_{\dim(\sigma^j)} \cup \{(\text{deg}(\sigma^j), \infty)\};$

---

Let  $\text{deg} : K \rightarrow \mathbb{Z}$  be a map on a CW complex defining a filtration, that is for any two cells  $\sigma < \tau$ ,  $\text{deg}(\sigma) \leq \text{deg}(\tau)$ . Let  $\{\sigma^j\}_{j=1}^n$  be an ordering of the cells of  $K$  respecting the order induced by  $\text{deg}$  such

that  $\{\sigma^j\}_{j=1}^i$  is a subcomplex of  $K$  for each  $1 \leq i \leq n$ .  $L_k$  denotes the set of the  $k$ -dimensional persistence intervals of the map  $\text{deg}$ .

**Algorithm 2** is a mere translation of the algorithm described in [55] into the HDVF framework. The purpose of doing so is to show that we can obtain a reduction for every step of the filtration and that we can apply the conclusions of Section 9 to the persistent homology theory.

## 9. Experimental complexity

We fix in this section  $\mathfrak{R} = \mathbb{Z}_2$ , so we are sure that we obtain a perfect HDVF and thus we compute the homology of the CW complex.

Computing the homology groups of a CW complex is considered in general a problem with  $\mathcal{O}(n^3)$  time complexity. Only [56] proves that it can be computed in matrix multiplication time, but there is no implementation of this algorithm. Nevertheless, it has been noticed that in practice the execution time is linear for homology [39, §4] and persistent homology [55, §4]. We estimated in Theorem 6.5 the complexity of our algorithm by bounding the number on non-zero entries in rows and columns by  $n$ , obtaining that Algorithm 1 can find a HDVF within  $(n/2) \cdot n^2 = n^3/2$  operations or  $(n/2) \cdot (n^2 + n^2 + n^2 + n^2) = 2n^3$  if we also want to obtain the associated reduction. Since these bounds are not tight, it should not be surprising that the complexity in practice is lower than cubic.

One advantage of the HDVF framework is that we can easily count the number of operations that we perform along its computation. At each step of Algorithm 1, updating the matrices  $H, F, G$  and  $D$  requires  $|F_{11}||G_{11}|$ ,  $|D_{21}||F_{11}|$ ,  $|G_{11}||D_{12}|$  and  $|D_{21}||D_{12}|$  operations respectively, where  $|v|$  denotes the number of non-zero entries in the vector  $v$ . Thus, updating the reduced boundary requires

$$|D_{21}||D_{12}|$$

operations (plus some operations to remove rows and columns). Moreover, updating all the reduction requires

$$|F_{11}||G_{11}| + |D_{21}||F_{11}| + |G_{11}||D_{12}| + |D_{21}||D_{12}| = (|F_{11}| + |D_{12}|)(|G_{11}| + |D_{21}|)$$

operations.

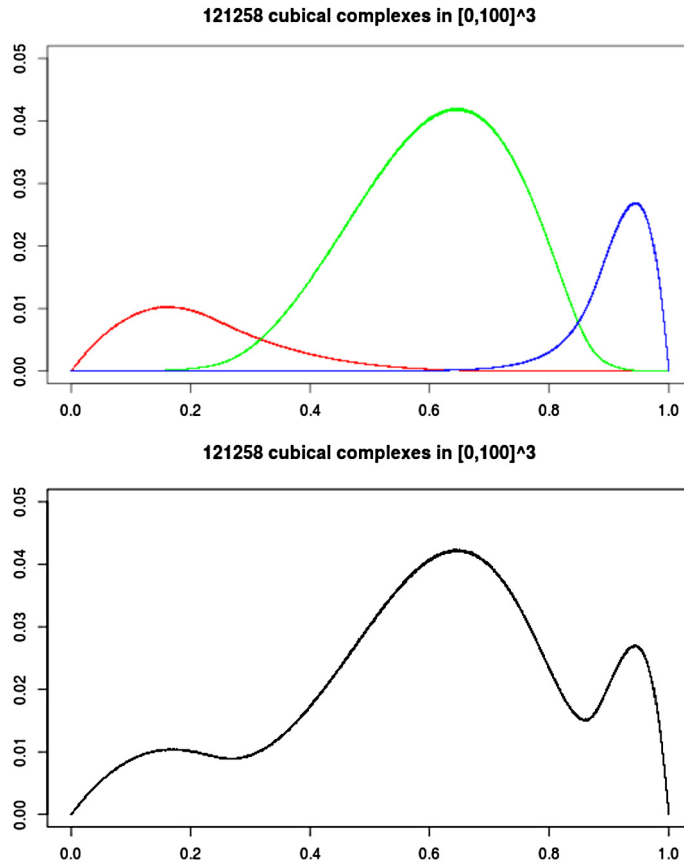
Let us study now the average complexity for two random models.

*Random cubical complexes.* We introduce a random model for constructing cubical complexes. We denote it by  $K(p, m)$  and it is similar to the *closed faces model* introduced in [57]. Let  $m \in \mathbb{Z}^+$  and  $p \in \mathbb{R}$ ,  $0 \leq p \leq 1$ . A cubical complex in  $K(p, m)$  is built by adding each cubical cell  $\sigma \subset [0, m]^3$  (with its faces) to the complex with probability  $p$ . Note that each cell  $\sigma$  belongs to the cubical complex with probability  $1 - (1 - p)^c$ , where  $c$  denotes the number of cofaces (in the full cubical complex  $[0, m]^3$ ) of  $\sigma$ , including itself. Thus, lower-dimensional cells are more frequent.

For each cubical complex  $K$  we denote by  $X(K)$  its number of cells. Also,  $Y^d(K)$  denotes the number of operations performed for computing a HDVF (which is the sum of  $|D_{21}||D_{12}|$  along its computation) and  $Y^r(K)$  denotes the number of operations needed for computing a HDVF and its reduction. We thus know that for each  $K$ ,

$$Y^d(K) \leq \frac{1}{2}X(K)^3 \quad \text{and} \quad Y^r(K) \leq 2X(K)^3$$

Let us point out two concerns:

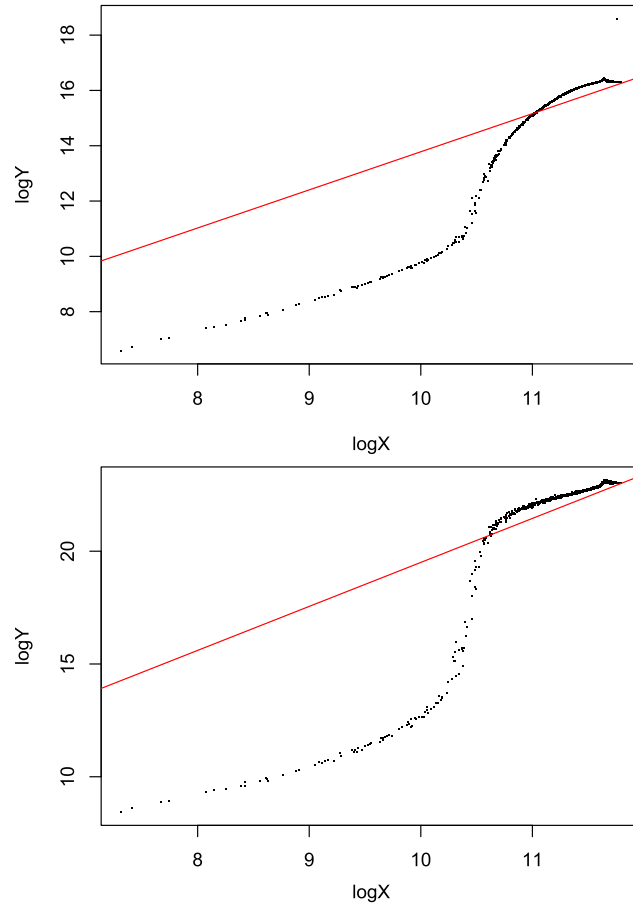


**Fig. 10.** We have computed the Betti numbers for 121258 cubical complexes in  $K(p, 100)$  with  $p \sim U(0, 1)$ . Top: for each cubical complex  $K$  we plot the points  $(X(K), \beta_0)$  (red),  $(X(K), \beta_1)$  (green) and  $(X(K), \beta_2)$  (blue) divided by the total number of cells  $(2 \cdot 100 + 1)^3$ . Bottom: for each cubical complex  $K$  we plot the point  $(X(K), \beta_0 + \beta_1 + \beta_2)$  divided by the total number of cells. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1. If  $K$  has large Betti numbers then there are several critical cells in the perfect HDVF, so we perform less than  $n/2$  steps. The relation between the parameter  $p$  and the number of critical cells in a perfect HDVF is unknown, and this could help to understand our experiments. Fig. 10 shows the Betti numbers for a large number of cubical complexes in  $K(p, 100)$  and their sum. We appreciate that the number of critical cells is at most 5% of the size of the complex, so it does not seem to be significant.
2. We do not make any smart choice for the pair  $(\sigma, \tau)$  in each step of the algorithm. This means that the quantities  $Y_d(K)$  and  $Y_r(K)$  are not optimized. We could have implemented a better choice that tries to minimize these values, but we have chosen not to do it since we want to obtain a really general result that can be also applied to compute persistent homology.

In our experiment we fix  $m = 25$  and we build 2217 cubical complexes with probability  $p$  uniformly distributed in  $[0, 1]$ . We want to show that the average complexity of Algorithm 1 is  $\mathcal{O}(n^\alpha)$  for some  $\alpha \in \mathbb{R}$ . Note, however, that our sample does not seem to fit to a polynomial function. For achieving this we fit our sample  $\{(\log(X_i), \log(Y_i^d))\}$  to a linear function  $y = b \cdot x$ . Using R [58] we obtain the 99.99% confidence interval  $[1.372086, 1.384346]$  for  $b$ . Thus, we can (statistically) affirm that  $b < 1.4$ . Consequently,  $Y^d < X^{1.4}$  and the average-case complexity is  $\mathcal{O}(n^{1.4})$ . Fig. 11-(top) shows the plot of  $\{(\log(X_i), \log(Y_i^d))\}$  together with the fitted linear function passing by the origin.

We repeat this study for  $Y^r$  (thus computing also the reduction). The 99.99% confidence interval for  $b$  is  $[1.940787, 1.959928]$  so we can affirm that in average, computing a perfect HDVF with its reduction requires



**Fig. 11.** Top: plot of  $\{(\log(X_i), \log(Y_i^d))\}$  for cubical complexes in  $K(25, p)$  together with the linear regression model passing by the origin. Bottom: the same for  $\{(\log(X_i), \log(Y_i^r))\}$ .

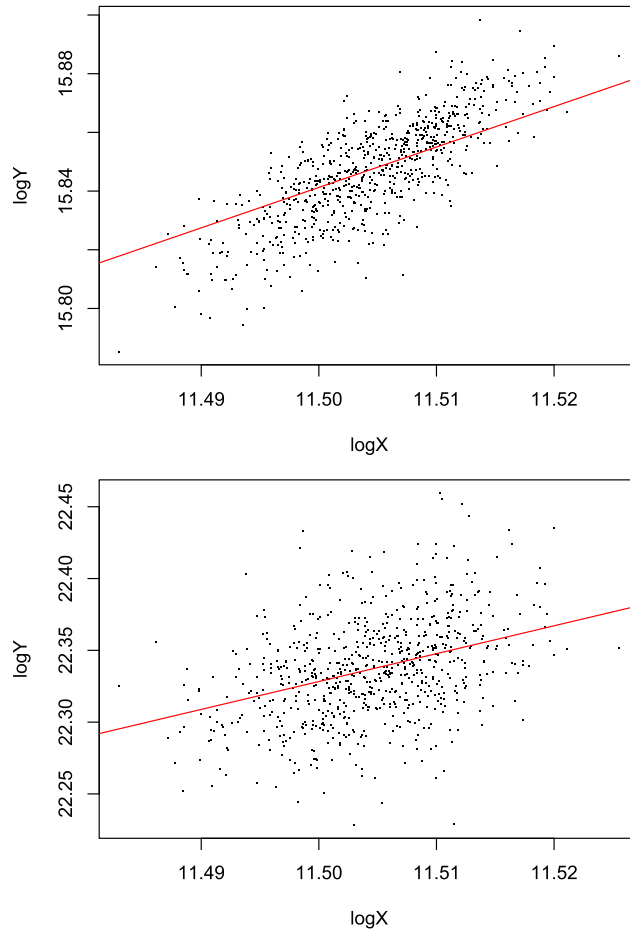
$\mathcal{O}(n^2)$ . Fig. 11-(bottom) shows the plot of  $\{(\log(X_i), \log(Y_i^r))\}$  together with the fitted linear function passing by the origin.

*Random volumes.* We may be interested in studying cubical complexes which come from binary volumes. We thus introduce the following random model. Let be  $m \in \mathbb{Z}^+$ . We consider an empty binary volume of size  $m \times m \times m$  and we add  $\lfloor m/10 \rfloor^3$  blocks of voxels of size  $\lfloor m/10 \rfloor \times \lfloor m/10 \rfloor \times \lfloor m/10 \rfloor$  at random (uniform) position. We can see this process as cutting a volume in small pieces and shuffling them. This binary volume can be transformed into a cubical complex by substituting each voxel for a cubical 3-dimensional cell and its faces. We denote this model by  $V(m)$ .

When building 1675 cubical complexes with the random model  $V(25)$  we obtain very similar results. By fitting  $\{(\log(X_i), \log(Y_i^d))\}$  to a linear function  $y = b \cdot x$  we obtain the 99.99% confidence interval  $[1.377406, 1.377593]$  for  $b$ . For  $Y^r$ , the interval is  $[1.941323, 1.941854]$ . Both fitted linear functions are depicted in Fig. 12.

## 10. Conclusion and future work

We have introduced a combinatorial structure that can be interpreted as a new class of discrete vector field, namely the homological discrete vector field, together with the theorems providing homological results for such extended DGVF. We have shown that this extended class of DGVF successfully reaches the correct



**Fig. 12.** Top: plot of  $\{(\log(X_i), \log(Y_i^d))\}$  for cubical complexes in  $V(25)$  together with the linear regression model passing by the origin. Bottom: the same for  $\{(\log(X_i), \log(Y_i^r))\}$ .

number of critical cells in complexes for which standard DGVPs cannot, such as the Bing’s house or the dunce hat.

We provide a simple sequential algorithm for computing a HDVP in Section 6. The reduction is updated by using formulas that depend on the previous step reduction, without inverting the matrix  $d(S)|_P$ . The worst-case complexity of this algorithm is  $\mathcal{O}(n^3)$ . Finally, we partially answer some questions about the algorithm and the existence of perfect HDVPs.

Section 7 introduces five basic operations that allow us to switch pairs of cells belonging to the sets  $P$ ,  $S$  or  $C$  of a HDVP. This extends and corrects a similar idea present in [34]. These operations allow us to change the shape of the homology (or cohomology) generators of a perfect HDVP. However, we do not explore how to do this in practice since there are some theoretical problems that must be solved before designing an algorithm, namely the existence of a sequence of operations that transform one HDVP into any other.

Section 8 is devoted to the relation between the HDVP framework and other homology algorithms. We prove that computing a HDVP is equivalent to compute a DGVP, an iterated Morse decomposition and the classical homology algorithm using the Smith normal form. It remains as an open question if every iterated Morse decomposition can be computed through a HDVP (the converse is proved). We also show how to compute persistence intervals using the HDVP framework. A very interesting task is to do the same for zigzag persistent homology [59].

In Section 9 we study the average-case complexity of our algorithm through an experimental approach. The validity of this study can be questioned, since we consider only two random models for cubical complexes with fixed parameters which do not sample the whole space of CW complexes. However we show that we can use the HDVF framework for giving a more concrete sense to the well accepted idea that homology and persistent homology can be computed in practice in almost linear time. We show, using simple linear regression, that a perfect HDVF (which provides the Betti numbers of the complex) can be obtained within  $\mathcal{O}(n^{1.4})$  operations in average. If we also want the reduction (and thus the homology groups), it requires  $\mathcal{O}(n^2)$  operations in average. However, we have not used any optimization technique such as reduction and coreductions [19], which should give even better estimations.

## Appendix A. Some matrix properties

The proofs in this work use several matrix properties that may not be trivial for the reader. We prefer to recall some of them in order to ease the reading of the proofs.

**Lemma A.1.** *Let be  $A = BCD$  the product of three matrices. Then,*

$$a_{i,j} = B_{i,\cdot}CD_{\cdot,j}$$

**Proof.**  $a_{i,j} = L_iAR_j$ , where  $L_i$  is a row vector with zeros everywhere except for the  $i$ -th position, and  $R_j$  is a column vector with zeros everywhere except for the  $j$ -th position. Therefore,  $a_{i,j} = L_iAR_j = L_iBCDR_j = B_{i,\cdot}CD_{\cdot,j}$ .  $\square$

**Lemma A.2.** *Let be  $A \in M_{m \times n}(\mathbb{Z})$ ,  $B \in M_{n \times n}(\mathbb{Z})$ ,  $B$  invertible and  $C = A \cdot B^{-1}$ . Then,*

$$c_{(i,j)} = \frac{1}{\det(B)} \det(\tilde{B}),$$

where  $\tilde{B}$  is the matrix identical to  $B$  except for the  $j$ -th row, which has been replaced by the  $i$ -th row of  $A$ .

**Lemma A.3.** (Schur determinant formula) *Let be*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ( $A \in M_{n \times n}(\mathbb{Z})$ ,  $B \in M_{n \times k}(\mathbb{Z})$ ,  $C \in M_{k \times n}(\mathbb{Z})$  and  $D \in M_{k \times k}(\mathbb{Z})$ ). If  $A$  is invertible then

$$\det(M) = \det(A) \cdot \det(D - CA^{-1}B).$$

**Lemma A.4.** (The Banachiewicz identity) *Let be*

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

a block matrix ( $A \in M_{n \times n}(\mathbb{Z})$ ,  $B \in M_{n \times k}(\mathbb{Z})$ ,  $C \in M_{k \times n}(\mathbb{Z})$  and  $D \in M_{k \times k}(\mathbb{Z})$ ). If  $A$  and  $D - CA^{-1}B$  are invertible, then  $M$  is invertible and

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$



We recall that the transpose of a matrix  $A$  is denoted  $A^\top$ .

**Lemma A.5** (Sherman–Morrison formula). *Let  $A \in M_{n \times n}(\mathbb{Z})$  and  $u, v \in \mathbb{Z}^n$ . If  $A$  is invertible and  $1 + v^\top Au \neq 0$  then*

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1}uv^\top A^{-1}}{1 + v^\top Au}.$$

**Lemma A.6.** *Let  $A \in M_{m \times n}(\mathbb{Z})$ , we say that it is an  $[r, c]$ -matrix if each row contains at most  $r$  non-zero entries and each column contains at most  $c$  non-zero entries. Thus,*

- *If  $A \in M_{m \times n}(\mathbb{Z})$  is an  $[r, c]$ -matrix and  $B \in M_{m \times n}(\mathbb{Z})$  is an  $[r', c']$ -matrix, then  $A + B$  is an  $[r + r', c + c']$ -matrix and it can be computed within  $\mathcal{O}(\min(m \cdot (r + r'), (c + c') \cdot n))$  operations.*
- *If  $A \in M_{m \times n}(\mathbb{Z})$  is an  $[r, c]$ -matrix and  $B \in M_{n \times p}(\mathbb{Z})$  is an  $[r', c']$ -matrix, then  $A \cdot B$  is an  $[r r', c c']$ -matrix and it can be computed within  $\mathcal{O}(m \cdot \min(r, c') \cdot p)$  operations.*

**Lemma A.7.** (Matrix inversion lemma) *Let  $A \in M_{n \times n}(\mathbb{Z})$  and  $u, v \in \mathbb{Z}^n$ . If  $A$  is invertible then*

$$\det(A + uv^\top) = \det(A) \cdot (1 + v^\top A^{-1}u).$$

**Proof.** We write

$$M = \left[ \begin{array}{c|c} 1 & -v \\ \hline u & A \end{array} \right]$$

Since  $\det(M) = \det(M^t)$ , by the Schur determinant formula (see Lemma A.3),

$$\begin{aligned} \det(M) &= \det(M^t) \\ \det(1) \cdot \det(A + uv) &= \det(A) \cdot \det(1 + vA^{-1}u) \\ \det(A + uv) &= (1 + vA^{-1}u) \cdot \det(A) \quad \square \end{aligned}$$

## References

- [1] J. Milnor, *Morse Theory*, Princeton University Press, 1962.
- [2] R. Forman, Morse theory for cell complexes, *Adv. Math.* 134 (1) (1998) 90–145, <http://dx.doi.org/10.1006/aima.1997.1650>.
- [3] R. Forman, A user's guide to discrete Morse theory, *Sémin. Lothar. Comb.* 48 (2002).
- [4] K. Mischaikow, Conley index theory, in: R. Johnson (Ed.), *Dynamical Systems*, in: *Lecture Notes in Mathematics*, vol. 1609, Springer, Berlin Heidelberg, 1995, pp. 119–207.
- [5] M. Mrozek, Index pairs algorithms, *Found. Comput. Math.* 6 (4) (2006) 457–493, <http://dx.doi.org/10.1007/s10208-005-0182-1>.
- [6] S. Day, W.D. Kalies, T. Wanner, Verified homology computations for nodal domains, *Multiscale Model. Simul.* 7 (4) (2009) 1695–1726, <http://dx.doi.org/10.1137/080735722>.
- [7] T. Teramoto, Y. Nishiura, Morphological characterization of the diblock copolymer problem with topological computation, *Jpn. J. Ind. Appl. Math.* 27 (2) (2010) 175–190, <http://dx.doi.org/10.1007/s13160-010-0014-9>.
- [8] P.W. Gross, P.R. Kotiuga, *Electromagnetic Theory and Computation*, Cambridge University Press, 2004, Cambridge Books Online.
- [9] P. Dlotko, R. Specogna, Efficient cohomology computation for electromagnetic modeling, *Comput. Model. Eng. Sci.* 60 (3) (2010) 247–278, <http://dx.doi.org/10.3970/cmcs.2010.060.247>.
- [10] H. Edelsbrunner, J. Harer, *Computational Topology – an Introduction*, American Mathematical Society, 2010.
- [11] M. Allili, D. Corriveau, Topological analysis of shapes using Morse theory, *Comput. Vis. Image Underst.* 105 (3) (2007) 188–199, <http://dx.doi.org/10.1016/j.cviu.2006.10.004>.
- [12] R. González-Díaz, M.J. Jiménez, B. Medrano, Cubical cohomology ring of 3D photographs, *Int. J. Imaging Syst. Technol.* 21 (1) (2011) 76–85, <http://dx.doi.org/10.1002/ima.20271>.

- [13] M. Mrozek, M. Zelawski, A. Gryglewski, S. Han, A. Krajniak, Homological methods for extraction and analysis of linear features in multidimensional images, *Pattern Recognit.* 45 (1) (2012) 285–298, <http://dx.doi.org/10.1016/j.patcog.2011.04.020>.
- [14] O. Delgado-Friedrichs, V. Robins, A.P. Sheppard, Skeletonization and partitioning of digital images using discrete Morse theory, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (3) (2015) 654–666, <http://dx.doi.org/10.1109/TPAMI.2014.2346172>.
- [15] P. Dlotko, R. Ghrist, M. Juda, M. Mrozek, Distributed computation of coverage in sensor networks by homological methods, *Appl. Algebra Eng. Commun. Comput.* 23 (1–2) (2012) 29–58, <http://dx.doi.org/10.1007/s00200-012-0167-7>.
- [16] J.R. Munkres, *Elements of Algebraic Topology*, Addison-Wesley, 1984.
- [17] A. Storjohann, Near optimal algorithms for computing Smith normal forms of integer matrices, in: *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation, ISSAC '96*, Zurich, Switzerland, July 24–26, 1996, 1996, pp. 267–274.
- [18] T. Kaczyński, M. Mrozek, M. Ślusarek, Homology computation by reduction of chain complexes, *Comput. Math. Appl.* 35 (4) (1998) 59–70, [http://dx.doi.org/10.1016/S0898-1221\(97\)00289-7](http://dx.doi.org/10.1016/S0898-1221(97)00289-7).
- [19] M. Mrozek, B. Batko, Coreduction homology algorithm, *Discrete Comput. Geom.* 41 (1) (2009) 96–118, <http://dx.doi.org/10.1007/s00454-008-9073-y>.
- [20] M. Mrozek, T. Wanner, Coreduction homology algorithm for inclusions and persistent homology, *Comput. Math. Appl.* 60 (10) (2010) 2812–2833, <http://dx.doi.org/10.1016/j.camwa.2010.09.036>.
- [21] F. Sergeraert, Effective homology, a survey, <http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Survey.pdf>, 1992, online; accessed 11 June, 2014.
- [22] P. Pilarczyk, P. Real, Computation of cubical homology, cohomology, and (co)homological operations via chain contraction, *Adv. Comput. Math.* 41 (1) (2015) 253–275, <http://dx.doi.org/10.1007/s10444-014-9356-1>.
- [23] R. González-Díaz, P. Real, On the cohomology of 3D digital images, *Discrete Appl. Math.* 147 (2–3) (2005) 245–263, <http://dx.doi.org/10.1016/j.dam.2004.09.014>.
- [24] R. González-Díaz, M.J. Jiménez, B. Medrano, P. Real, A tool for integer homology computation:  $\lambda$ -AT-model, *Image Vis. Comput.* 27 (7) (2009) 837–845, <http://dx.doi.org/10.1016/j.imavis.2008.10.001>.
- [25] A. Berciano, H. Molina-Abril, P. Real, Searching high order invariants in computer imagery, *Appl. Algebra Eng. Commun. Comput.* 23 (1–2) (2012) 17–28, <http://dx.doi.org/10.1007/s00200-012-0169-5>.
- [26] R. González-Díaz, M.J. Jiménez, B. Medrano, P. Real, Chain homotopies for object topological representations, *Discrete Appl. Math.* 157 (3) (2009) 490–499, <http://dx.doi.org/10.1016/j.dam.2008.05.029>.
- [27] R. Forman, Combinatorial vector fields and dynamical systems, *Math. Z.* 228 (4) (1998) 629–681, <http://dx.doi.org/10.1007/PL00004638>.
- [28] A. Gonzalez-Lorenzo, A. Bac, J. Mari, P. Real, Computing homological information based on directed graphs within discrete objects, in: *16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2014*, Timisoara, Romania, September 22–25, 2014, 2014, pp. 571–578.
- [29] M.K. Chari, On discrete Morse functions and combinatorial decompositions, *Discrete Math.* 217 (1–3) (2000) 101–113, [http://dx.doi.org/10.1016/S0012-365X\(99\)00258-7](http://dx.doi.org/10.1016/S0012-365X(99)00258-7).
- [30] D.N. Kozlov, *Combinatorial Algebraic Topology, Algorithms and Computation in Mathematics*, vol. 21, Springer, 2008.
- [31] T. Lewiner, H. Lopes, G. Tavares, Toward optimality in discrete Morse theory, *Exp. Math.* 12 (3) (2003) 271–285, <http://dx.doi.org/10.1080/10586458.2003.10504498>.
- [32] M. Joswig, M.E. Pfetsch, Computing optimal Morse matchings, *SIAM J. Discrete Math.* 20 (1) (2006) 11–25, <http://dx.doi.org/10.1137/S0895480104445885>.
- [33] A. Engström, Discrete Morse functions from Fourier transforms, *Exp. Math.* 18 (1) (2009) 45–53, <http://dx.doi.org/10.1080/10586458.2009.10128886>.
- [34] H. Molina-Abril, P. Real, Homological spanning forest framework for 2D image analysis, *Ann. Math. Artif. Intell.* 64 (4) (2012) 385–409, <http://dx.doi.org/10.1007/s10472-012-9297-7>.
- [35] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, Discrete Morse theoretic algorithms for computing homology of complexes and maps, *Found. Comput. Math.* 14 (1) (2014) 151–184, <http://dx.doi.org/10.1007/s10208-013-9145-0>.
- [36] A.T. Lundell, S. Weubgram, *The Topology of CW Complexes*, Springer, New York, 1969.
- [37] T. Kaczyński, K. Mischaikow, M. Mrozek, *Computational Homology, Applied Mathematical Sciences*, vol. 157, Springer-Verlag, New York, 2004.
- [38] J.R. Stallings, *Lectures on Polyhedral Topology*, Tata Institute of Fundamental Research, Bombay, 1968.
- [39] P. Dlotko, T. Kaczyński, M. Mrozek, T. Wanner, Coreduction homology algorithm for regular CW-complexes, *Discrete Comput. Geom.* 46 (2) (2011) 361–388, <http://dx.doi.org/10.1007/s00454-010-9303-y>.
- [40] D.S. Dummit, R.M. Foote, *Abstract Algebra*, John Wiley & Sons, Hoboken, NJ, 2004.
- [41] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, New York, 2002.
- [42] S. Peltier, S. Alayrangues, L. Fuchs, J. Lachaud, Computation of homology groups and generators, in: *Discrete Geometry for Computer Imagery, 12th International Conference, DGCI 2005*, Poitiers, France, April 13–15, 2005, *Proceedings*, 2005, pp. 195–205.
- [43] D. Boltcheva, S. Merino Aceitunos, J.-C. Léon, F. Hétyroy, Constructive Mayer–Vietoris Algorithm: Computing the Homology of Unions of Simplicial Complexes, *Research Report RR-7471*, INRIA, December 2010.
- [44] K.S. Brown, R. Geoghegan, An infinite-dimensional torsion-free  $FP_\infty$  group, *Invent. Math.* 77 (2) (1984) 367–381, <http://dx.doi.org/10.1007/BF01388451>.
- [45] R.H. Bing, Some aspects of the topology of 3-manifolds related to the Poincaré conjecture, in: *Lectures on Modern Mathematics*, vol. 2, 1964.
- [46] E. Zeeman, On the dunce hat, *Topology* 2 (4) (1963) 341–358, [http://dx.doi.org/10.1016/0040-9383\(63\)90014-4](http://dx.doi.org/10.1016/0040-9383(63)90014-4).
- [47] P. Dlotko, H. Wagner, Computing homology and persistent homology using iterated Morse decomposition, preprint, [arXiv:1210.1429](https://arxiv.org/abs/1210.1429), 2012, pp. 1–31.

- [48] J.E. Hopcroft, R.M. Karp, An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* 2 (4) (1973) 225–231, <http://dx.doi.org/10.1137/0202019>.
- [49] R. Ayala, D. Fernández-Ternero, J.A. Vilches, Perfect discrete Morse functions on 2-complexes, *Pattern Recognit. Lett.* 33 (11) (2012) 1495–1500, <http://dx.doi.org/10.1016/j.patrec.2011.08.011>.
- [50] M. Hachimori, Simplicial complex library, [http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index\\_eng.html](http://infoshako.sk.tsukuba.ac.jp/~hachi/math/library/index_eng.html).
- [51] F.R. Gantmacher, *The Theory of Matrices*, vol. 1, Chelsea Publishing Company, New York, N.Y., 1959.
- [52] A.M. Davie, A.J. Stothers, Improved bound for complexity of matrix multiplication, *Proc. R. Soc. Edinb., Sect. A, Math.* 143 (2013) 351–369, <http://dx.doi.org/10.1017/S0308210511001648>.
- [53] M. Juda, M. Mrozek, CAPD::RedHom v2 – Homology software based on reduction algorithms, in: *Mathematical Software – ICMS 2014 – 4th International Congress*, Seoul, South Korea, August 5–9, 2014, *Proceedings*, 2014, pp. 160–166.
- [54] B. Benedetti, F.H. Lutz, Random discrete Morse theory and a new library of triangulations, *Exp. Math.* 23 (1) (2014) 66–94, <http://dx.doi.org/10.1080/10586458.2013.865281>.
- [55] A. Zomorodian, G.E. Carlsson, Computing persistent homology, *Discrete Comput. Geom.* 33 (2) (2005) 249–274, <http://dx.doi.org/10.1007/s00454-004-1146-y>.
- [56] N. Milosavljevic, D. Morozov, P. Skraba, Zigzag persistent homology in matrix multiplication time, in: *Proceedings of the 27th ACM Symposium on Computational Geometry*, Paris, France, June 13–15, 2011, 2011, pp. 216–225.
- [57] M. Werman, M.L. Wright, Intrinsic volumes of random cubical complexes, *Discrete Comput. Geom.* 56 (1) (2016) 93–113, <http://dx.doi.org/10.1007/s00454-016-9789-z>.
- [58] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>, 2015.
- [59] G.E. Carlsson, V. de Silva, Zigzag persistence, *Found. Comput. Math.* 10 (4) (2010) 367–405, <http://dx.doi.org/10.1007/s10208-010-9066-0>.