

Parallel decomposition of persistence modules through interval bases

Original

Parallel decomposition of persistence modules through interval bases / DE GREGORIO, Alessandro; Guerra, Marco; Scaramuccia, Sara; Vaccarino, Francesco. - ELETTRONICO. - (2021).

Availability:

This version is available at: 11583/2942232 since: 2021-12-01T19:05:15Z

Publisher:

Published

DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Parallel decomposition of persistence modules through interval bases

Alessandro De Gregorio ¹, Marco Guerra ¹, Sara Scaramuccia ¹, and Francesco Vaccarino ¹

¹Department of Mathematical Sciences, Politecnico di Torino, Italy

June 23, 2021

Abstract

We introduce an algorithm to decompose any finite-type persistence module with coefficients in a field into what we call an *interval basis*. This construction yields both the standard persistence pairs of Topological Data Analysis (TDA), as well as a special set of generators inducing the interval decomposition of the Structure theorem. The computation of this basis can be distributed over the steps in the persistence module. This construction works for general persistence modules on a field \mathbb{F} , not necessarily deriving from persistent homology. We subsequently provide a parallel algorithm to build a persistent homology module over \mathbb{R} by leveraging the Hodge decomposition, thus providing new motivation to explore the interplay between TDA and the Hodge Laplacian.

Keywords— Graded module decomposition; persistent homology; Hodge decomposition; Hodge Laplacian

Mathematics Subject Classification (2020)— 13P20, 55N31, 62R40

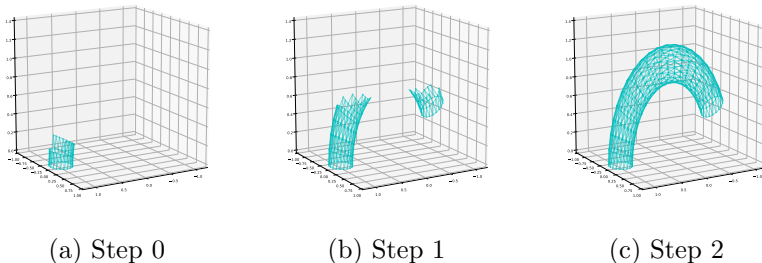


Figure 1: A 3-step filtration by sublevel sets, with respect to the the z coordinate, of a half torus.

1 Introduction

Consider a monotone persistence module $\mathcal{M} = \{M_i, \varphi_i : M_i \rightarrow M_{i+1}\}_i$ over a field \mathbb{F} (PM), indexed over the natural numbers where each M_i is called i^{th} step and each φ_i is called i^{th} structure map. Assume that the module is of *finite type*, i.e. that each \mathbb{F} -vector space M_i is finite dimensional and only a finite number of maps among the φ_i 's are not isomorphisms.

This is the typical setting of persistent homology, the subject in which the name persistence module originally arose. The two fundamental properties of finite type persistent modules imply that each $\{M_i, \varphi_i\}$ can be decomposed, up to ordering, into a direct sum of interval modules so that the module is completely described by its *persistence diagram* or *barcode*, and furthermore that this decomposition is stable with respect to noise if the PM is made of the homology groups of a geometric sampling.

These properties largely account for the success of Topological Data Analysis, a field that leverages computational techniques in algebraic topology to extract information from data.

More explicitly, a persistence module can be written uniquely, up to permutations of summands, as a direct sum of persistence (sub)modules

$$\mathcal{M} \cong \bigoplus_m I_{[b_m, d_m]}, \tag{1}$$

where each term $I_{[b_m, d_m]}$, called *interval*, is an indecomposable persistence submodule depending on the indexes $b_m \leq d_m$. The multiset of pairs $\{(b_m, d_m)\}$

alessandro.degregorio@polito.it; marco.guerra@polito.it; sara.scaramuccia@polito.it; francesco.vaccarino@polito.it

forms the barcode.

Under the isomorphism in Eq.(1), for each m , one might search for an element v_m in \mathcal{M} which, through the application of the structure maps, generates the submodule $I(v_m)$ corresponding to $I_{[b_m, d_m]}$.

In this work, we call a set of generators $\{v_m\}_m$ of \mathcal{M} an *interval basis* if it satisfies

$$\mathcal{M} = \bigoplus_m I(v_m) \cong \bigoplus_m I_{[b_m, d_m]}. \quad (2)$$

The classical algorithm to construct the barcode of a finite type persistence module [5] works at the level of the chain complexes and thus is limited to persistence module induced by homology. The algorithm yields the pairs $\{(b_m, d_m)\}$, and can be used to extract a set of homology generators, as done in [13]. However, the extracted set of generators does not generally satisfy Eq. (2). For example, consider a filtration of the half torus such as the one depicted in Fig. 1, which will be our running example throughout the paper. The classical algorithm correctly identifies the persistence pairs, but returns the generators depicted in Fig. 2, in accordance with the so-called *elder rule*. These generators do not induce a decomposition such as the one in Eq. (2), because their images in the final step of the filtration become homologous. Algebraically, this entails a linear relation between non-zero elements of \mathcal{M} and thus the sum of interval modules in Eq. (2) is not direct.

A principal contribution of this work is to provide a procedure to build an interval basis. From an interval basis, it is trivial to compute the persistence pairs. As an example, for the case of 1-homology of the half torus, a set of generators forming an interval basis is depicted in Fig. 3.

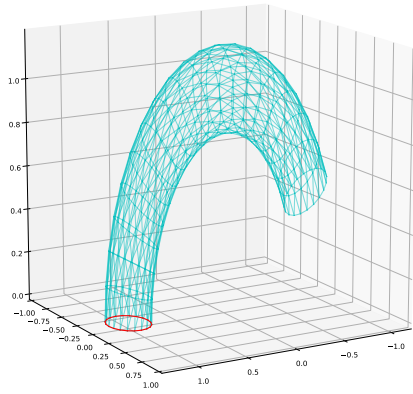
With the classical approach one is only allowed to compute the decomposition of a persistence module that is the homology of a filtration of simplicial complexes. Moreover, the algorithm intrinsically works with objects (chain groups and chain maps) whose dimension is typically much larger than that of their homology module.

Our proposed method, instead, addresses the case of finding a decomposition of a general, i.e. not necessarily a homology, persistence module. As such, we work directly at the level of $\{M_i, \varphi_i\}$. As a consequence, the bulk of our algorithm can be performed independently for each M_i , i.e. is largely parallelizable.

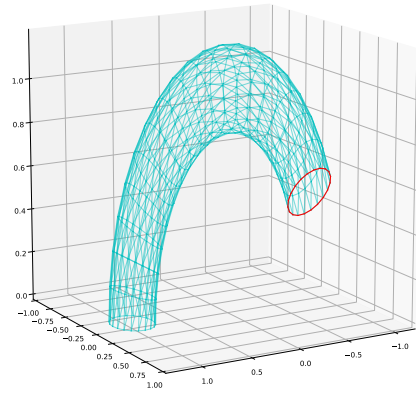
We can summarize the main contribution with the following result

Theorem. *Given a finite-type \mathbb{F} -persistence module, the output of Algorithm 3 provides an interval basis.*

After giving a specific instance of our algorithm for the case of *k-persistent homology modules*, i.e., persistence modules obtained through the *k*-homology functor



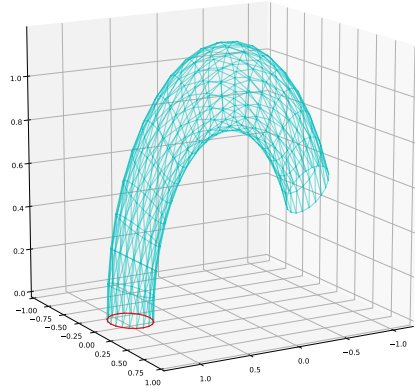
(a) Generator 1



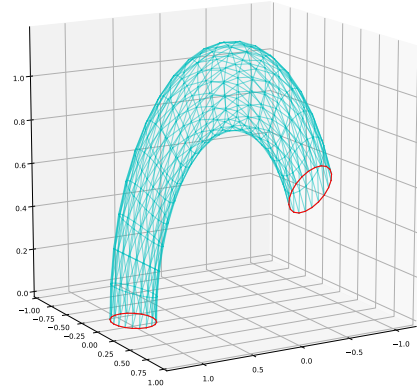
(b) Generator 2

Figure 2: Generators, classical algorithm.

applied to a filtration, we specialize our procedure to the case where the chain spaces are Euclidean, i.e. have a scalar product. For example, we address the case of k -persistent homology modules with coefficients in \mathbb{R} , which can be built from the kernel of the k Hodge Laplacian operator with respect to a filtration of simplicial complexes. We construct the induced maps between kernels and, by the known properties of the Hodge Laplacian, prove it to be equivalent to the k -persistent homology module. We then apply our decomposition to obtain an interval basis, which is shown in Fig. 4. This furthers the recent trend of exploring the interplay between topological data analysis and the properties of the Hodge Laplacian [10, 25, 26].

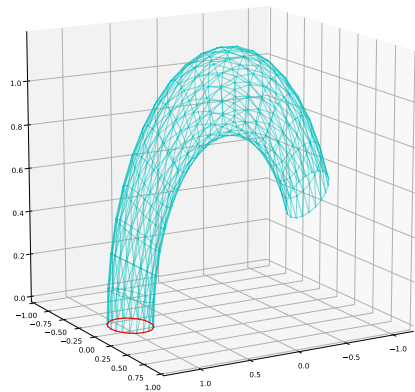


(a) Generator 1

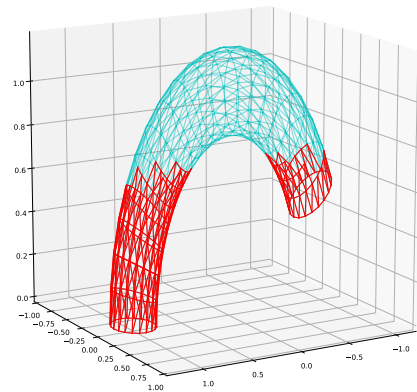


(b) Generator 2

Figure 3: Interval basis generators for the k -persistent homology module.



(a) Generator 1



(b) Generator 2

Figure 4: Interval basis generators for the harmonic persistence module

Contributions and related work

The key results of this paper can be summarized as:

1. An algorithm for the decomposition of any monotone persistence module on any field. We do not assume the module is induced by homology. As such, the module might derive from any sequence of simplicial maps.
2. The generators we provide are such that a decomposition of the persistence module is given by the interval modules they each generate. We call this an *interval basis*.
3. Our algorithm is parallelizable for each step of the module.

We further provide a parallel algorithm to construct a persistence module from a general collection of simplicial maps. In the special case of real coefficients, we use the Hodge Laplacian as a construction method for homology groups and maps between them. We therefore specialize the decomposition algorithm.

The original approach to the decomposition of persistence modules arising from filtrations of simplicial complexes was introduced in [13], based on the simplex pairing approach of [7]. These algorithms were limited to working with \mathbb{Z}_2 coefficients and simplicial maps deriving from inclusions of simplicial complexes. Later, this framework was extended to arbitrary coefficient fields in [5], which established the notorious correspondence between persistence modules and graded modules over a PID. It is immediate to construct a set of generators from the output of these algorithms. However, they do not induce an interval basis decomposition of the module. A different approach to attempt a principled choice of homology generators is to introduce a notion of geometric minimality, as it was done in [9, 16, 19]. The distributed computation of the persistence pairs has been addressed in [1], where arguments from discrete Morse theory are employed to observe that the reduction of a boundary matrix obtained from a filtration of simplicial complexes can be performed in *chunks*. The chunk algorithm can, like the one proposed here, be run in parallel for each filtration step, and is incremental in that it only considers cells that are born within the step itself. Its objective, however, is to compute the persistence pairs of a filtration, whereas our proposal works for general (albeit monotone) simplicial maps, needs not consider the (typically much larger) space of k -chains and computes an interval basis.

Subsequently, the case of a monotone sequence of general simplicial maps was addressed in [8], which relies on the notion of simplex annotation. In this way, and by interpreting each simplicial map as a sequence of inclusions and vertex collapses, a consistent homology basis can be maintained efficiently. Later in [20] a variation

of the coning approach of [8] is proposed, which takes a so-called *simplicial tower* and converts it into a filtration, while preserving its barcode, with asymptotically small overhead. Our approach to the problem is different in that we do not need to work at the level of simplicial complexes or maps, and in fact our decomposition is independent from the persistence module being induced by a filtration of simplicial complexes.

To our knowledge, the first paper to deal with persistence module decompositions without assuming it arises from a family of inclusion maps was an instance of topological persistence whose underlying structure is not monotone: the so-called *zigzag persistence* ([4]). Zigzag persistence is based on a sequence of structure maps whose direction is not necessarily left to right. They provide an incremental algorithm which computes a decomposition of the zigzag module, without focusing on its generators. Our proposal differs in that we are restricted to the case of monotone persistence, but in turn our focus is on the construction of an interval basis, and on a parallel algorithm.

An analogous point of view is taken in a recent paper by Carlsson et al. ([3]). The decomposition of the general zigzag module is tackled from the matrix factorization viewpoint. Starting from a type-A quiver representation, their goal is to compute a basis for each homogenous component of a persistence module in such a way that the structure maps are represented as echelon form matrices. In other words, the matrix associated to the quiver representation is in canonical form. Our algorithm can decompose the module while avoiding the change of basis operations that are needed for the construction of the echelon form. Furthermore, by specializing to monotone persistence modules we can avoid the divide and conquer approach and instead perform computations in parallel across all degrees.

Structure of the paper

In the background Section 2 we review some concepts about persistence modules, not necessarily made by the homology of some space, and introduce our notion of interval basis in connection with the Structure Theorem. Section 3 introduces the main algorithm (Algorithm 3) to compute an interval basis of a general persistence module. In Section 4 we give a constructive method, acting independently over each step (Algorithm 6) and structure map (Algorithm 7), for constructing a k -persistent homology module. Moreover, we see how our constructive method can be adapted to the case of harmonic persistence module where each step is obtained as the kernel of the Hodge Laplacian operator, and each structure map is easily obtained via Algorithm 8. The Appendices contain bonus background material to support the proofs in Section 4 and to recall the construction from simplicial complexes to chain complexes in the usual TDA pipeline.

2 Background

In order to formalize the problem addressed in this work, we introduce the *persistence module*.

We define a persistence module \mathcal{M} as pair $\{M_i, \varphi_i\}_{i \in \mathbb{N}}$ consisting of, for any $i \in \mathbb{N}$:

- a finite-dimensional \mathbb{F} -vector space M_i called *step*
- a linear map $\varphi_i : M_i \rightarrow M_{i+1}$ called *structure map*, such that there exists $n \in \mathbb{N}$ for which φ_i is an isomorphism for all $i \geq n$.

We denote by $\varphi_{i,j} : M_i \rightarrow M_j$ with $i < j$ the composition $\varphi_{j-1} \circ \dots \circ \varphi_i$. We remark that the reader may find in the literature more general objects defined as persistence modules. Our definition corresponds to what is normally referred to as finite type persistence module [6].

It is useful to underline that any persistence module \mathcal{M} can be thought of as a graded module $\bigoplus_i M_i$ over the ring of polynomials $\mathbb{F}[x]$, or a graded $\mathbb{F}[x]$ -module, for short. Indeed, the homogeneous part of degree i is the step M_i and the action over M_i under the indeterminate x is given by applying the structure map φ_i . Being that correspondence an equivalence of categories, we can transpose to persistence modules some notions of graded modules such as, isomorphisms, homogeneous elements, direct sums, generators, and submodules.

Let $I(v) = \{I_i(v), \psi_i(v)\}_{i \in \mathbb{N}}$ be the submodule of \mathcal{M} generated by $v \in M_b$ (i.e., v is homogeneous in M_b of degree b). Observe that we have

$$I_i(v) = \begin{cases} \langle \varphi_{b,i}(v) \rangle & \text{if } i \geq b, \\ 0 & \text{otherwise,} \end{cases} \quad \psi_i(v) = \begin{cases} \varphi_i|_{\langle \varphi_{b,i}(v) \rangle} & \text{if } i \geq b, \\ 0 & \text{otherwise,} \end{cases}$$

where the brackets $\langle \cdot \rangle$ denotes the linear space spanned by their argument.

Now, define an (*integer*) *interval* $[b, d]$ with $b \leq d$ to be the limited set $\{b, b+1, \dots, d\}$ and $[b, \infty]$ to be the unlimited set $\{b, b+1, \dots\}$. The *interval module* $I_{[b,d]}$ relative to the interval $[b, d]$, possibly with $d = \infty$, is the persistence module $\{I_i, \psi_i\}_{i \in \mathbb{N}}$ such that

$$I_i = \begin{cases} \mathbb{F} & \text{if } b \leq i \leq d, \\ 0 & \text{otherwise,} \end{cases} \quad \psi_i = \begin{cases} \text{id}_{\mathbb{F}} & \text{if } b \leq i < d, \\ 0 & \text{otherwise.} \end{cases}$$

Remark 1. Fix a degree b . For each $v \in M_b$, there exists $d \in \mathbb{N}$ or $d = \infty$ such that

$$I(v) \cong I_{[b,d]}.$$

Indeed, each step in $I(v)$ is either isomorphic to the vector space \mathbb{F} or to 0 and the structure maps are either isomorphisms or the null map. If there exists an integer r , such that $I_r(v) = 0$, we take d to be the minimum of such r 's. Otherwise, we set $d = \infty$.

Definition 1. Given a persistence module $\mathcal{M} = \{M_i, \varphi_i\}_{i \in \mathbb{N}}$, a finite family $\{v_1, \dots, v_N\} \subseteq \bigoplus_i M_i$ of homogeneous elements is an *interval basis* for \mathcal{M} if and only if

$$\bigoplus_{m=1}^N I(v_m) = \mathcal{M}$$

Proposition 1. Every persistence module $\mathcal{M} = \{M_i, \varphi_i\}_{i \in \mathbb{N}}$ admits an interval basis $\{v_1, \dots, v_N\} \subseteq \bigoplus_i M_i$.

Proof. The existence of an interval basis for each \mathcal{M} follows from the interval decomposition corresponding to the Structure Theorem [5] for finitely generated graded $\mathbb{F}[x]$ -modules. Indeed, the interval decomposition implies that \mathcal{M} decomposes into a direct sum of interval modules of the form

$$\mathcal{M} \cong \bigoplus_{m=1}^N I_{[b_m, d_m]}, \quad (3)$$

where the intervals $[b_m, d_m]$ with $b_m \leq d_m \leq \infty$ are uniquely determined up to reorderings. Let $\Psi = \{\Psi_i\}_{i \in \mathbb{N}}$ be the graded isomorphism of the interval decomposition. Then, for each summand $I_{[b_m, d_m]}$, the map Ψ_{b_m} detects a vector $v_m \in M_{b_m}$. By Remark 1, we have that $I(v_m) \cong I_{[b_m, r_m]}$ for some $b_m \leq r_m \leq \infty$. Observe now that, for all indexes $i \in \mathbb{N}$, the decomposition isomorphism satisfies $\varphi_i \circ \Psi_i = \Psi_{i+1} \circ \psi_i$, where ψ_i is the structure map of $I_{[b_m, d_m]}$. This implies that $r_m = d_m$ for all indexes $i \in \mathbb{N}$. \square

Formula (3) defines the *barcode* $B(\mathcal{M})$ as the multiset of pairs $\{(b_m, d_m)\}_{m=1}^N$, i.e., the collection of pairs in the formula counted with multiplicity. The pairs of the kind (b, ∞) are called *essential pairs*.

An alternative persistence module decomposition is the one that obtains a so-called *coherent basis*.

Definition 2. Given a persistence module $\{M_i, \varphi_i\}_{i=0}^n$, a *coherent basis* is given by a set of bases $\{\mathcal{B}_i = \{v_1^i, \dots, v_{d_i}^i\}\}_{i=0}^n$, one for each space M_i , such that for any $i = 0, \dots, n$ it holds,

$$\left(\varphi_i(v_k^i) = v_{i_k}^{i+1} \in \mathcal{B}_{i+1} \vee \varphi_i(v_k^i) = 0 \right) \quad \forall v_k^i \in \mathcal{B}_i.$$

In words, a coherent basis is a choice of a basis for each step of the module such that the structure maps are block identity matrices.

Example 1. Consider a persistence module $\{M_i, \varphi_i\}_{i=0}^n$ where all φ_i 's are injective. Take any basis of M_0 as \mathcal{B}_0 . Suppose inductively that we have already obtained \mathcal{B}_i For some $i = 0, \dots, n - 1$, since φ_i is injective, $\varphi_i(\mathcal{B}_i)$ defines a basis for a subspace B_{i+1} in M_{i+1} . Take any basis completion of B_{i+1} in M_{i+1} as \mathcal{B}_{i+1} . Hence, $\{\mathcal{B}_i\}_{i=0}^n$ is a coherent basis for $\{M_i, \varphi_i\}_{i=0}^n$.

Remark 2. Given an interval basis, we get an induced coherent basis.

Indeed, an interval basis generates each step basis \mathcal{B}_i by simply considering the images of the vectors of the interval basis across the maps φ_j with $j \leq i$.

Persistence module decomposition via interval basis

Decomposing a persistence module via interval basis consists in retrieving, given a persistence module \mathcal{M} , an interval basis v_1, \dots, v_N with N counting the number of interval modules in the interval decomposition of Definition 1.

In the following section, we present Algorithm 3 for addressing the decomposition as stated above. In order to do so from now on, we restrict to persistence modules $\mathcal{M} = \{M_i, \varphi_i\}$ admitting no essential pairs in the decomposition. Notice that this is not restrictive since we deal with persistence modules of finite type. This means that there exists an integer n such that φ_m is an isomorphism for all $m \geq n$. By setting $\varphi'_m = 0$ for all $m \geq n + 1$, we slightly modify our persistence module to get $\mathcal{M}' = \{M_i, \varphi'_i\}$ in such a way that the interval decomposition admits no essential pairs. We notice that the interval modules $I_{[b,d]}$ with $d \leq n$ are preserved. Whereas, an interval module $I_{[b,\infty]}$ decomposing \mathcal{M} corresponds to an interval module $I_{[b,n+1]}$ decomposing \mathcal{M}' . Hence, our working assumption is not restrictive in practice. The role of our assumption is clarified in the following section.

3 Decomposition of persistence modules

Consider a persistence module $\{M_i, \varphi_i\}_{i=1}^n$, as described in the previous section. In particular, we suppose without loss of generality that the module is finite and that the last map $\varphi_n : M_n \rightarrow M_{n+1}$ is the null one. Denote with m_i the dimension of the space M_i and with r_i the dimension of $\text{Im}(\varphi_{i-1})$. For each i there is a flag of vector sub-spaces of M_i given by the kernel of the maps $\varphi_{i,j}$:

$$0 \subseteq \ker(\varphi_{i,i+1}) \subseteq \ker(\varphi_{i,i+2}) \subseteq \cdots \subseteq \ker(\varphi_{i,n+1}) = M_i. \quad (4)$$

where the last equality holds by the assumption above.

Denote for simplicity each space $\ker(\varphi_{i,j})$ as V_j^i .

An *adapted basis* for the flag in M_i is given by a set of linearly independent vectors $\mathcal{V}^i = \{v_1, \dots, v_{m_i}\}$, and an index function $J : \mathcal{V}^i \rightarrow \{1, \dots, n - i + 1\}$, such that

$$V_{i+s}^i = \langle \{v \in \mathcal{V}^i \mid J(v) \leq i + s\} \rangle \quad \forall s, 1 \leq s \leq n - i + 1. \quad (5)$$

In words, an adapted basis is an ordered list of vectors in M_i such that for every j , the first $\dim V_j^i$ vectors are a basis of V_j^i (where we assume an empty list is a basis of the trivial space).

Remark 3. Notice that, without loss of generality, it is possible to choose an adapted basis of M_i in such a way that it contains as a subset a basis of $\text{Im}(\varphi_{i-1})$.

Proof. Let us consider an adapted basis $\mathcal{V}^i = (t_1, \dots, t_{m_i})$ for the flag of kernels in M_i , with the vectors t_1, \dots, t_{m_i} ordered by index function J . We construct the desired basis explicitly: set $\underline{\mathcal{V}}^i = \{t_1\}$. For every $s = 2, \dots, m_i$, if $t_s \notin \langle \underline{\mathcal{V}}^i \rangle + \text{Im}(\varphi_{i-1})$ add the vector t_s to $\underline{\mathcal{V}}^i$. Otherwise, it must hold that $t_s = \sum_{a < s} \lambda_a t_a + x$ with $x \in \text{Im}(\varphi_{i-1})$. Then we add to $\underline{\mathcal{V}}^i$ the vector $x = t_s - \sum_{a < s} \lambda_a t_a$. In this way $\underline{\mathcal{V}}^i$ is another adapted basis, and the elements added by the second route form a basis of $\text{Im}(\varphi_{i-1})$. \square

From now on, therefore, we shall assume that each basis \mathcal{V}^i is in the form of Remark 3.

Let us introduce two subspaces of $\langle \mathcal{V}^i \rangle$: it holds that $\langle \mathcal{V}^i \rangle = \langle \mathcal{V}_{\text{Birth}}^i \rangle \oplus \langle \mathcal{V}_{\text{Im}}^i \rangle$, where $\mathcal{V}_{\text{Im}}^i$ is the subset of \mathcal{V}^i made of a basis of $\text{Im} \varphi_{i-1}$, and $\mathcal{V}_{\text{Birth}}^i$ is its complement.

Our objective is to construct a basis of the whole persistence module by leveraging the adapted bases at each step i .

Definition 3. Let us define $\mathcal{V} := \bigcup_i \mathcal{V}_{\text{Birth}}^i$

So, set \mathcal{V} is made up of the elements of the adapted basis in each degree i that are not elements of $\text{Im}(\varphi_{i-1})$. In the following, we prove that \mathcal{V} is in fact an interval basis for $\{M_i, \varphi_i\}_{i=1}^n$.

Lemma 1. *For any $i < j$, the map $\varphi_{i,j}|_T$ restriction of the map $\varphi_{i,j}$ to the subspace $T = \langle \{v \in \mathcal{V}^i \mid J(v) > j\} \rangle$ is an injection.*

Proof. By definition of T , it holds $M_i = \ker(\varphi_{i,j}) \oplus T$. If the restriction of $\varphi_{i,j}$ onto T were not injective, then T and $\ker(\varphi_{i,j})$ would have nontrivial intersection. This is a contradiction. \square

Lemma 2. For any $i < j \in \mathbb{N}$, it holds

$$\varphi_{i,j} (\langle \mathcal{V}_{\text{Birth}}^i \rangle) \cap \varphi_{i,j} (\langle \mathcal{V}_{\text{Im}}^i \rangle) = \{0\}. \quad (6)$$

Proof. Suppose that the intersection of the two considered spaces contains a nonzero vector u :

$$0 \neq u = \varphi_{i,j} \left(\sum_{v_k \in \mathcal{V}_{\text{Birth}}^i} \lambda_k v_k \right) = \varphi_{i,j} \left(\sum_{w_l \in \mathcal{V}_{\text{Im}}^i} \mu_l w_l \right).$$

Denote by u_{B} and u_{I} the vectors

$$u_{\text{B}} = \sum_{\substack{J(v_k) > j \\ v_k \in \mathcal{V}_{\text{Birth}}^i}} \lambda_k v_k, \quad u_{\text{I}} = \sum_{\substack{J(w_l) > j \\ w_l \in \mathcal{V}_{\text{Im}}^i}} \mu_l w_l.$$

It holds $u = \varphi_{i,j}(u_{\text{B}}) = \varphi_{i,j}(u_{\text{I}})$, since all the elements v such that $J(v) \leq j$ belong to $\ker(\varphi_{i,j})$. Then, u is the image through $\varphi_{i,j}$ of an element of $T = \langle \{v \in \mathcal{V}^i \mid J(v) > j\} \rangle$. On the other hand also the difference $u_{\text{B}} - u_{\text{I}}$ belongs to the same space and is mapped to zero by $\varphi_{i,j}$. The restriction of $\varphi_{i,j}$ to T is injective because of Lemma 1, therefore it must be $u_{\text{B}} - u_{\text{I}} = 0$. Since, $\langle \mathcal{V}^i \rangle = \langle \mathcal{V}_{\text{Birth}}^i \rangle \oplus \langle \mathcal{V}_{\text{Im}}^i \rangle$, it must be $u_{\text{B}} = u_{\text{I}} = 0$, hence $u = 0$. \square

Theorem 1. The set \mathcal{V} is an interval basis for the persistence module M .

Proof. Say that $\mathcal{V} = \{v_1, \dots, v_N\}$. Each vector v_j in the set \mathcal{V} induces an interval module $I(v_j)$. We want to show that $M = \bigoplus_{j=1}^N I(v_j)$. To do so, let us see that for each $0 \leq i \leq n$, the space M_i is exactly $\bigoplus_{j=1}^N I_i(v_j)$. By construction we know that

$$M_i = \langle \{v \in \mathcal{V}^i \mid v \notin \text{Im}(\varphi_{i-1})\} \rangle \oplus \text{Im}(\varphi_{i-1}) = \bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v = i}} I_i(v) \oplus \text{Im}(\varphi_{i-1}). \quad (7)$$

All we have to show is that $\text{Im}(\varphi_{i-1})$ can be written as $\bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v < i}} I_i(v)$. At first we will see that it holds $\text{Im}(\varphi_{i-1}) = \bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v < i}} I_i(v)$. By definition, it holds $\bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v < i}} I_i(v) \subseteq \text{Im}(\varphi_{i-1})$. On the other hand, $\text{Im}(\varphi_{i-1}) \subseteq \bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v < i}} I_i(v)$. We can see it by induction. Let us consider $M_0 = \langle \mathcal{V}^0 \rangle$, where none of the elements of \mathcal{V}^0 belongs to $\text{Im}(\varphi_{-1}) = \{0\}$. Then, $\text{Im}(\varphi_0) \subseteq \bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v = 0}} I_1(v)$. Suppose by induction that for any $k-1$ it holds $\text{Im}(\varphi_{k-2}) \subseteq \bigoplus_{\substack{v \in \mathcal{V} \\ \text{deg } v < k-1}} I_{k-1}(v)$. Then, since $M_{k-1} = \langle \{v \in \mathcal{V}^{k-1} \mid v \notin \text{Im}(\varphi_{k-2})\} \rangle \oplus \text{Im}(\varphi_{k-2})$, it holds

$$\text{Im}(\varphi_{k-1}) = \bigoplus_{\substack{v \in \mathcal{V}^{k-1} \\ v \notin \text{Im}(\varphi_{k-2})}} I_k(v) + \varphi_{k-1}(\text{Im}(\varphi_{k-2})) \quad (8)$$

Therefore, by the induction hypothesis

$$\text{Im} (\varphi_{k-1}) \subseteq +_{\substack{v \in \mathcal{V} \\ \text{deg } v < k}} I_k(v).$$

Now that we have shown that $\text{Im} (\varphi_{i-1}) = +_{\substack{v \in \mathcal{V} \\ \text{deg } v < i}} (I(v))_i$, it remains to see that this sum is direct. Suppose to have a non trivial combination $w_1 + \dots + w_k = 0$, where each w_q belongs to $I_i(v_{t_q})$. Suppose that the w_1, \dots, w_k are ordered according to the degree of the element v_{t_q} that generates the interval module they belong to. Let us say that these elements have a maximum degree $l < i$. Then, it holds

$$w_1 + \dots + w_k = \varphi_{l,i}(x) + \sum_{v_{t_r} | \text{deg } v_{t_r} = l} \lambda_r \varphi_{l,i}(v_{t_r}),$$

where $x \in \text{Im} (\varphi_{l-1})$. Because of Lemma 2, it must be

$$\varphi_{l,i}(x) = \sum_{v_{t_r} | \text{deg } v_{t_r} = l} \lambda_r \varphi_{l,i}(v_{t_r}) = 0.$$

On the other hand we also assumed that the w_q are different from zero, therefore the index $J(v_{t_q})$ of the vectors in the adapted basis has to be greater than $i - l$. Hence, because of Lemma 1, it holds that $\sum_{v_{t_r} | \text{deg } v_{t_r} = l} \lambda_r \varphi_{l,i}(v_{t_r}) = 0$ implies $\sum_{v_{t_r} | \text{deg } v_{t_r} = l} \lambda_r v_{t_r} = 0$. Since the $\{v_{t_r} | \text{deg } v_{t_r} = l\}$ are linearly independent it must be $\lambda_r = 0$ for any r , and therefore $w_r = 0$. The same idea can be repeated for all the previous elements w_1, \dots, w_s , coming from interval modules generated by vectors with degree less than l . Since there are finitely many vectors this process has an end and it shows that all the vectors w_1, \dots, w_k are 0 and the sum is direct. \square

We now provide an explicit construction for set \mathcal{V} . To do so, we must first obtain sets \mathcal{V}^i .

Remark 4. Notice that the construction of each \mathcal{V}^i is independent from the others. Therefore they can be computed simultaneously.

Construction of $\mathcal{V}_{\text{Birth}}^i$

We first recall that a simple basis extension algorithm is given by the the procedure described in the following Algorithm 1. The set \mathcal{W} is ordered, and its elements are added to \mathcal{U} in their ascending order in \mathcal{W} , so that \mathcal{U} is extended to a basis of $\langle \mathcal{U} \rangle + \langle \mathcal{W} \rangle$. In the following, we refer to the extension of basis \mathcal{U} by the vectors in set \mathcal{W} through Algorithm 1 as $\text{bca}(\mathcal{U}, \mathcal{W})$.

Algorithm 1: Basis completion algorithm

Input: linearly independent vectors $\mathcal{U} = \{u_1, \dots, u_r\}$, linearly independent vectors $\mathcal{W} = \{w_1, \dots, w_n\}$;

Result: minimal set of vectors $w_{i_1}, \dots, w_{i_p} \notin \langle \mathcal{U} \rangle$ such that $\langle \mathcal{U} \cup \{w_{i_1}, \dots, w_{i_p}\} \rangle = \langle \mathcal{U} \cup \mathcal{W} \rangle$

$\mathcal{R} = \{\}$;

for $i=1, \dots, n$ **do**

if $\text{rank} \mathcal{U} < \text{rank}(\mathcal{U} \cup \{w_i\})$ **then**

$\mathcal{U} = \mathcal{U} \cup \{w_i\}$;

$\mathcal{R} = \mathcal{R} \cup \{w_i\}$;

end

end

return \mathcal{R}

We now give a general algorithm to construct the set $\mathcal{V}_{\text{Birth}}^i$ for a given M_i of the persistence module. To find an adapted basis \mathcal{V}^i we need only to iteratively complete a basis of $\ker(\varphi_{i,j})$ to a basis of $\ker(\varphi_{i,j+1})$, using for example the algorithm described above. In general, the basis obtained through Algorithm 1 will not contain a basis of the space $\text{Im}(\varphi_{i-1})$, so the procedure does not match the construction in Remark 3. We show below that this is not strictly necessary, so we can save computations without hindering the correctness of the algorithm. In fact, we obtain the set $\mathcal{V}_{\text{Birth}}^i$ applying the reduction algorithm, using as inputs any basis of $\text{Im}(\varphi_{i-1})$ and the adapted basis \mathcal{V}^i . This is equivalent to using the construction of Remark 3 and then discarding the elements that belong to $\text{Im}(\varphi_{i-1})$. Consider the ordered set $\mathcal{V}^i = \{v_1, \dots, v_{m_i}\}$ and a set $\mathcal{U} = \{u_1, \dots, u_{r_i}\}$, basis of $\text{Im}(\varphi_{i-1})$. Vector v_j is modified by the procedure in Remark 3 if it can be written as a linear combination of the vectors $\{u_1, \dots, u_{r_i}, v_1, \dots, v_{i-1}\}$. If that is the case, it is modified so that it belongs to $\mathcal{V}_{\text{Im}}^i$, and then it is not included in $\mathcal{V}_{\text{Birth}}^i$. In the same way, Algorithm 1 discards such vectors without performing further calculations. The vectors belonging to $\mathcal{V}_{\text{Birth}}^i$ are the same in both procedures, hence the result is valid.

The full procedure to construct $\mathcal{V}_{\text{Birth}}^i$ from M_i and the structure maps is described in Algorithm 2.

Algorithm 2: Single step decomposition

Input: map $\varphi_{i-1} : M_{i-1} \rightarrow M_i$, maps $\{\varphi_j : M_j \rightarrow M_{j+1}\}, i \leq j \leq N$;

Result: $\mathcal{V}_{\text{Birth}}^i$ and associated index function J

Take a basis $\mathcal{U} = \{u_1, \dots, u_k\}$ of $\text{Im}(\varphi_{i-1})$;

$k := \dim \text{Im}(\varphi_{i-1})$;

$R := \text{Id} : M_i \rightarrow M_i$;

$r := \text{rank}(R)$;

$\mathcal{V}_{\text{Birth}}^i = \{\}$;

for $s = 0, \dots, N - i$ **do**

$R = \varphi_{i+s} \cdot R$;

$r' = \text{rank } R$;

if $r' < r$ **then**

$B = \text{basis of } \ker(R)$;

$v_1^s, \dots, v_{\rho_s}^s = \text{bca}(\mathcal{U}, B)$;

$\mathcal{U} = \mathcal{U} \cup \{v_1^s, \dots, v_{\rho_s}^s\}$;

$\mathcal{V}_{\text{Birth}}^i = \mathcal{V}_{\text{Birth}}^i \cup \{v_1^s, \dots, v_{\rho_s}^s\}, J(v_1^s) = \dots = J(v_{\rho_s}^s) = s + 1$;

$r = r'$;

end

if $r = 0$ **or** $|\mathcal{V}_{\text{Birth}}^i| + k = \dim M_i$ **then**

break ;

end

end

return $\mathcal{V}_{\text{Birth}}^i, J : \mathcal{V}_{\text{Birth}}^i \rightarrow \{1, \dots, N - i + 1\}$.

In the following, we refer to the construction of $\mathcal{V}_{\text{Birth}}^i$ through Algorithm 2 as $\text{ssd}(M_i)$.

Once the decomposition of each space is performed, it is immediate to assemble the interval basis \mathcal{V} . Further, we can read the persistence diagram of module $\{M_i, \varphi_i\}_i$ off of the interval basis by storing the indices of appearance and death of its elements, without increasing the computational cost. This is the content of Algorithm 3, which summarizes the procedures introduced so far into a single routine that takes a persistence module and returns its interval basis and persistence diagram.

Algorithm 3: Persistence module decomposition

Input: persistence module $\{M_i, \varphi_i\}_{i=1}^n$;

Result: persistence basis $\{v_i^s\}$ and persistence diagram

$\varphi_0 :=$ empty matrix with $\dim M_0$ rows and 0 columns;

$\varphi_{n+1} :=$ empty matrix with 0 rows and $\dim M_n$ columns;

$PB = \{\}$;

$PD = \{\}$;

for $b = 1, \dots, n + 1$ **do**

$\mathcal{V}_{\text{Birth}}^i, J = \text{ssd}(\varphi_{b-1}, \{\varphi_j\}_{j \geq b})$;

if $\mathcal{V}_{\text{Birth}}^i$ *is not empty* **then**

 Add to the interval basis PB the vectors in $\mathcal{V}_{\text{Birth}}^i$;

 Update the persistence diagram with the points $(b, J(v) + b)$ for

$v \in \mathcal{V}_{\text{Birth}}^i$;

end

end

return *Interval basis, persistence diagram*

We refer to the decomposition of Algorithm 3 as $\text{pmd}(M)$.

Example. Consider the following \mathbb{R} -persistence module

$$0 \xrightarrow[\begin{pmatrix} 0 \end{pmatrix}]{\varphi_0} \mathbb{R} \xrightarrow[\begin{pmatrix} 1 \\ 0 \end{pmatrix}]{\varphi_1} \mathbb{R}^2 \xrightarrow[\begin{pmatrix} 1 & 1 \end{pmatrix}]{\varphi_2} \mathbb{R} \xrightarrow[\begin{pmatrix} 0 \end{pmatrix}]{\varphi_3} 0$$

Where the matrices below each arrow represent the map above it in the canonical bases. We showcase the procedure of Algorithm 3 and compute its interval basis. Notice that this example matches the persistence module generated by persistent homology in Fig. 2. For $i = 0, 1, 2, 3$ we need to compute $\mathcal{V}_{\text{Birth}}^i$.

Clearly φ_0 is the null map, so the flag for the first step is trivial and $\mathcal{V}_{\text{Birth}}^0$ is empty.

For $i = 1$ we have $\text{Im}(\varphi_{i-1}) = 0$ and $\ker(\varphi_{1,2}) = \ker(\varphi_{1,3}) = 0$, so $\mathbb{R} = \ker(\varphi_{1,4})$. By ssd we extend a basis of $\text{Im}(\varphi_0)$ (which is empty) to a basis of \mathbb{R} , which yields

vector 1. Then $\mathcal{V}_{\text{Birth}}^1 = \{1\}$ with persistence pair $(1, 4)$.

For $i = 2$ we have $\text{Im}(\varphi_{i-1}) = \langle \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rangle$. Furthermore $\ker(\varphi_{2,3}) = \langle \begin{pmatrix} 1 \\ -1 \end{pmatrix} \rangle$, so we extend the basis of $\text{Im}(\varphi_1)$ against the basis of $\ker(\varphi_{2,3})$ obtaining set $\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \}$, which spans \mathbb{R}^2 , so ssd terminates setting $\mathcal{V}_{\text{Birth}}^2 = \{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \}$ with persistence pair $(2, 4)$.

For $i = 3$ we have $\text{Im}(\varphi_{i-1}) = \mathbb{R}$, so $\mathcal{V}_{\text{Birth}}^3$ is empty.

Finally, the interval basis is $\mathcal{V} = \{1, \begin{pmatrix} 1 \\ -1 \end{pmatrix}\}$, with persistence diagram $PD = \{(1, 4), (2, 4)\}$.

The case of real coefficients

In case we use the field \mathbb{R} in the persistence module, we can specialise the decomposition of the space described in the previous paragraph. We will use the following notation: given a matrix A with m rows and n columns, $A[:, i]$ denotes the i -th column of the matrix, whereas $A[:, : i]$ denotes the submatrix given by the first i columns of A . The same notation is used on the first arguments in the parenthesis to denote operations on rows. We will make use of this simple result in linear algebra.

Lemma 3. *Given three vector spaces V_1, V_2 , and V_3 over \mathbb{R} and two linear maps $\psi_1 : V_1 \rightarrow V_2$ and $\psi_2 : V_2 \rightarrow V_3$ it holds*

$$\ker(\psi_2 \circ \psi_1) = \ker(\psi_1) \oplus \ker\left(\psi_2 \circ \psi_1|_{(\ker(\psi_1))^\perp}\right).$$

Proof. Let x be an element of $\ker(\psi_2 \circ \psi_1)$. It can be written uniquely as $x = v + w$, with $v \in \ker(\psi_1)$ and $w \in (\ker(\psi_1))^\perp$. Since $(\psi_2 \circ \psi_1)(v + w) = 0$ and $v \in \ker(\psi_1)$, it must be $\psi_2(\psi_1(w)) = 0$, therefore $w \in \ker(\psi_2 \circ \psi_1)$. Then, w belongs to $\ker\left(\psi_2 \circ \psi_1|_{(\ker(\psi_1))^\perp}\right)$ and the statement follows. \square

Fix M_0 , and suppose that $\varphi_n = 0$. For each M_i , denote with d_i the number $\dim M_i$. Consider φ_0 and decompose it via the SVD decomposition in $\varphi_0 = U_0 S_0 V_0^T$. If $r_0 = \text{rank } \varphi_0$, then $k_0 = d_0 - r_0$ is the dimension of $\ker \varphi_0$. Notice that S_0 is a matrix $d_1 \times d_0$ with non-zero elements only on the first r_0 positions on the main diagonal. Therefore, if e_i is the i -th element of the canonical basis of \mathbb{R}^{d_0} , with $r_0 < i \leq d_0$, then $\varphi_0 V_0 e_i = U_0 S_0 e_i = 0$. Then, a basis of $\ker \varphi_0$ is given by the vectors $\{V_0 e_{r_0+1}, \dots, V_0 e_{d_0}\}$. The index function J attains the value 1 on all of them. All such vectors will be also in the kernel of the maps $\varphi_{0,j}$ for all $j > 0$. In order to avoid repetitions, it will be considered only the restriction of each $\varphi_{0,j}$ on the orthogonal complement of $\ker \varphi_0$. This operation will not change

the result because of Lemma 3. To do so, consider the map $\tilde{\varphi}_0 = U_0 \tilde{S}_0$, where $\tilde{S}_0 = S_0[:, : r_0]$, given by the first r_0 columns of S_0 . Repeating the same process, it will be considered $m_1 = \varphi_1 \tilde{\varphi}_0$ instead of $\varphi_{0,2}$. Call $d_1 = d_0 - k_0$. Decompose again $m_1 = U_1 S_1 V_1^T$ and call $r_1 = \text{rank } m_1$ and $k_1 = d_1 - r_1 = \dim \ker m_1$. Again, a basis of $\ker m_1$ is given by the vectors $V_1 e_{r_1+1}, \dots, V_1 e_{d_1}$. Recall that this vectors are expressed in the basis $\{V_0[:, 1], \dots, V_0[:, r_0]\}$ of $\ker \varphi_0^\perp$. To return them in the canonical basis of M_0 it is sufficient to consider the matrix η_0 with d_0 rows and r_0 columns such that $\eta_0[i, j]$ is equal to 1 if $1 \leq i = j \leq r_0$ and 0 otherwise. Then, the vectors in the canonical basis of M_0 are $\{V_0 \eta_0 V_1 e_{r_1+1}, \dots, V_0 \eta_0 V_1 e_{d_1}\}$. In this case the value of the index function for these vectors will be 2. For the general step j , consider $m_j = \varphi_j \tilde{m}_{j-1} = U_j S_j V_j^T$. The adapted basis of M_0 will be updated with the vectors

$$V_0 \eta_0 \dots V_{j-1} \eta_{j-1} V_j e_x, \quad r_j + 1 \leq x \leq d_j, \quad (9)$$

and it will be $J(V_0 \eta_0 \dots V_{j-1} \eta_{j-1} V_j e_x) = j + 1$ for every $r_j + 1 \leq x \leq d_j$. Once all the vectors are obtained, as in the general case, it is necessary to complete a basis of $\text{Im}(\varphi_{i-1})$ to a basis of M_0 , introducing the vectors in \mathcal{V} in ascending order given by the function J . The resulting vectors will be part of the interval basis. The procedure is encoded in Algorithm 5, which makes use of the matrix decomposition routine Algorithm 4 and specializes Algorithm 2 to the case of real coefficients. We denote it by $\text{ssdR}(M_i)$. Then, the full decomposition of Algorithm 3 can be specialized to the reals by replacing $\text{ssd}(M_i)$ with $\text{ssdR}(M_i)$.

Algorithm 4: Matrix decomposition

Input: matrix A ;

Result: Restriction of A on the space orthogonal to its kernel with respect to a basis V of the domain, V matrix whose columns are a basis of the domain of A , $\dim(\ker A)^\perp, \dim \ker A$

$U, S, V = \text{SVD}(A)$;

$nz = \text{rank } S, d = \text{number of columns of } A, dk = d - nk$;

$R = US[:, : nz]$;

return R, V, nz, dk

Algorithm 5: single step decomposition on \mathbb{R}

Input: map $\varphi_{i-1} : M_{i-1} \rightarrow M_i$, maps $\{\phi_j : M_j \rightarrow M_{j+1}\}, i \leq j \leq N$;

Result: Vectors $\mathcal{V}_{\text{Birth}}^i$

$U, S, V = \text{SVD}(\varphi_{i-1})$;

$r := \text{rank}(\varphi_{i-1})$;

$\mathcal{U} = U[:, : r]$ basis of the image of φ_{i-1} ;

$\mathcal{V}_{\text{Birth}}^i = \{\}$; $lk = 0$;

$R = \text{Id} : M_i \rightarrow M_i$;

$d := \dim M_i$;

$V_{\text{tot}} = I_d$;

for $s = 0, \dots, N - i$ **do**

$R = \varphi_{s+1} \cdot R$;

if *number of rows of* $R = 0$ **then**

$k :=$ number of columns of R ;

$V = I_k$;

$nz = 0$;

$dk = k$;

else

$R, V, nz, dk = \text{dec}(R)$;

end

$V_{\text{temp}} = I_d, l = \text{ord}(V), V_{\text{temp}}[:, l] = V^t$;

$V_{\text{tot}} = V_{\text{tot}} \cdot V_{\text{temp}}$;

if $dk > 0$ **then**

$\mathcal{T} = \text{bca}(\mathcal{U}, V_{\text{tot}}[:, d - lk - dk : d - lk])$;

$\mathcal{U} = \mathcal{U} \cup V_{\text{tot}}[:, d - lk - dk : d - lk]$;

$\mathcal{V}_{\text{Birth}}^i = \mathcal{V}_{\text{Birth}}^i \cup \mathcal{T}$;

$J(t) = s + 1$ for all $t \in \mathcal{T}$;

$lk = lk + dk$;

end

if $nz = 0$ **or** $|\mathcal{V}| + r = d$ **then**

break;

end

end

return $\mathcal{V}_{\text{Birth}}^i, J$

4 The k -persistent homology module

A *chain complex* with coefficients in \mathbb{F} is a sequence $C = (C_\bullet, \partial_\bullet)$ of \mathbb{F} -vector spaces connected by linear maps with $k \in \mathbb{N}$

$$\dots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

such that $\partial_{k+1}\partial_k = 0$ for all $k \in \mathbb{N}$. Each \mathbb{F} -vector space C_k is called the space of k -chains.

A chain complex can be constructed from a *simplicial complex*. We refer the reader to the Example 6 in Appendix B for the details of the construction.

The subspace $Z_k = \ker(\partial_k)$ is called the space of k -cycles. The subspace $B_k = \text{Im}(\partial_{k+1})$ is called the space of k -boundaries. The condition $\partial_{k+1}\partial_k = 0$ ensures that

$$B_k \subseteq Z_k, \quad \text{for all } k \in \mathbb{N}.$$

The quotient space $H_k = Z_k/B_k$ is called the k -homology space.

A *chain map* $f : (C_\bullet, \partial_\bullet^C) \rightarrow (D_\bullet, \partial_\bullet^D)$ is a collection of linear maps $f_k : C_k \rightarrow D_k$ such that

$$f_k \partial_{k+1}^C = \partial_{k+1}^D f_{k+1}, \quad \text{for all } k \in \mathbb{N}. \quad (10)$$

A chain map induces a linear map of homology spaces. Indeed, for each k -cycle z in C_k or D_k , we write $[z]_C$ or $[z]_D$ for its projection onto the homology space H_k^C or H_k^D , respectively. By Equation (10), we get that $f_k(B_k^C) \subseteq B_k^D$. Hence, we get induced a map $\tilde{f}_k : H_k^C \rightarrow H_k^D$ defined by

$$\tilde{f}_k([z]_C) = [f_k(z)]_D. \quad (11)$$

This property is called *functoriality* of the homology construction.

The k -persistent homology module with coefficients in \mathbb{F} is the persistence module obtained from a sequence of chain maps

$$0 \xrightarrow{f_{(0)}} C_{(1)} \xrightarrow{f_{(1)}} \dots \longrightarrow \dots \xrightarrow{f_{(i-1)}} C_{(i)} \xrightarrow{f_{(i)}} \dots \xrightarrow{f_{(n-1)}} C_{(n)}, \quad (12)$$

by applying the homology construction to get the following diagram of vector spaces

$$0 \xrightarrow{\tilde{f}_{k,(0)}} H_{k,(1)} \xrightarrow{\tilde{f}_{k,(1)}} \dots \longrightarrow \dots \xrightarrow{\tilde{f}_{k,(i-1)}} H_{k,(i)} \xrightarrow{\tilde{f}_{k,(i)}} \dots \xrightarrow{\tilde{f}_{k,(n-1)}} H_{k,(n)}. \quad (13)$$

The persistent homology is a persistence module $\mathcal{M} = \{M_i, \varphi_i\}_{i=0}^n$ by setting $M_i = H_{k,(i)}$ and $\varphi_i = \tilde{f}_{k,(i)}$.

In the following section, we introduce a way of computing the persistent homology of a given sequence of chain complex connected by linear maps.

4.1 Construction of the persistent homology module

In this section, we present a construction of the k -persistent homology module $\{H_k^i, \varphi_i\}_{i \in \mathbb{N}}$ for general coefficients from a general chain complex $C = (C_\bullet, \partial_\bullet)$. For each index $i \in \mathbb{N}$, we first introduce Algorithm 6 for constructing the step H_k^i . Afterwards, we introduce Algorithm 7 for retrieving the structure map φ_i as the map at homology level induced by a general chain map $f : (C_\bullet, \partial_\bullet^C) \rightarrow (D_\bullet, \partial_\bullet^D)$ in degree k . In the persistent homology module construction, f is one of chain maps $f_{(i)}$ connecting subsequent chain complexes (See Section 4). Notice that these two constructions can be distributed and performed in parallel for each $i \in \mathbb{N}$.

Computing the homology steps in parallel

For computing the step H_k^i , we want to find a split sequence of kind c) (see the Splitting Lemma 5 in the Appendix A) for the short exact sequence given by $B_k \subseteq Z_k$ and the definition of homology as $H_k^i = Z_k/B_k$ (see Example 2 in the Appendix A)

$$\begin{array}{ccccccc}
 0 & \longrightarrow & K & \xrightarrow{s} & A & \xrightarrow{q} & Q \longrightarrow 0 \\
 & & \text{id}_K \uparrow & & h \uparrow & & \text{id}_Q \uparrow \\
 0 & \longrightarrow & K & \xrightarrow{\text{incl}_1} & K \oplus Q & \xrightarrow{\text{proj}_2} & Q \longrightarrow 0,
 \end{array} \tag{14}$$

where the map h is the isomorphism assigning the standard basis of the biproduct to the basis $\{h_1, \dots, h_{\beta_k}\}$ retrieved by the Algorithm 6, and the maps incl_i and proj_i are the standard inclusions and projection of the component i in the biproduct space.

Algorithm 6 accepts as inputs the boundary matrices ∂_{k+1} and ∂_k expressed in terms of the standard basis of C . The algorithm reflects the standard idea introduced by Edelsbrunner et al ([7]) and called *left-to-right reduction*, and it boils down to computing the column reduction of the boundary matrices ∂_k and ∂_{k+1} . Let us call V_k and V_{k+1} the matrices such that $R_k = \partial_k V_k$ and $R_{k+1} = \partial_{k+1} V_{k+1}$ are matrices reduced by columns. The columns of V_k corresponding to zero columns of R_k provide the the basis $\{v_1, \dots, v_s\}$ for Z_k , whereas the non-zero columns of R_{k+1} provide the basis $\{b_1, \dots, b_r\}$ for B_k . Let us call $\beta_k = \dim H_k$. In the for-cycle, the left-to-right reduction is exploited again to obtain the result by completing the basis of B_k to a basis of Z_k using the vectors yielded by V_k . It is straightforward

to check that the Diagram 14 commutes.

Algorithm 6: Computing homology

Input: Boundary matrices $\partial_k, \partial_{k+1}$ of the chain complex C ;

Result: Betti number β_k and basis $\{h_1, \dots, h_{\beta_k}, b_1, \dots, b_r\}$ of Z_k , where
 $\text{span}\{[h_1], \dots, [h_{\beta_k}]\} = H_k$ and $\text{span}\{b_1, \dots, b_r\} = B_k$.

Compute the reduction $R_k = \partial_k V_k$;

Compute the reduction $R_{k+1} = \partial_{k+1} V_{k+1}$;

$b_1, \dots, b_r :=$ non-zero columns of R_{k+1} ;

$v_1, \dots, v_s :=$ columns of V_k corresponding to zero columns of R_k ;

$J :=$ matrix with columns $\{b_1, \dots, b_r, v_1, \dots, v_s\}$;

$\beta_k = 1$;

for $i = r + 1, \dots, r + s$ **do**

while $\exists j < i$ s.t. $\text{low}(J[i]) = \text{low}(J[j])$ **do**

$l := \text{low}(J[i]);$

$\gamma := J[l, i] / J[l, j];$

$J[i] = J[i] - \gamma J[j];$

end

if $J[i]$ is non-zero **then**

$h_{\beta_k} := J[i];$

$\beta_k = \beta_k + 1;$

end

end

return β_k , basis $\{h_1, \dots, h_{\beta_k}, b_1, \dots, b_r\}$

Computing the homology structure maps in parallel

For any index i , Algorithm 7 computes the structure map φ_i as the map \tilde{f}_k at homology level induced by a general linear map as the degree k of chain map $f_k : C_k \rightarrow D_k$. Our algorithm assumes that the homology splittings $Z_k(C) \cong H_k(C) \oplus B_k(C)$ and $Z_k(D) \cong H_k(D) \oplus B_k(D)$ are already obtained by applying Algorithm 6.

In particular, this means that we have the splitting bases: $\{[h_1^C], \dots, [h_{\beta_k^C}^C]\}$ for $H_k(C)$, $\{[h_1^D], \dots, [h_{\beta_k^D}^D]\}$ for $H_k(D)$, and $\{b_1^D, \dots, b_r^D\}$ for $B_k(D)$.

Theorem 2. *The map \tilde{f}_k defined in Algorithm 7 is well-defined and it is the map induced by f_k through the homology functor.*

Proof. The map \tilde{f}_k is obtained as $\tilde{f}_k = \text{proj}_1 h^{-1} f_k \text{hincl}_1$ as derived in Remark 10 in the Appendix A. Precomposing by $f_k \text{hincl}_1$ implies applying f_k to $[h_i^C] \in H_k(C)$. The following composition of $\text{proj}_1 h^{-1}$ implies that the image of $[h_i^C] \in H_k(C)$

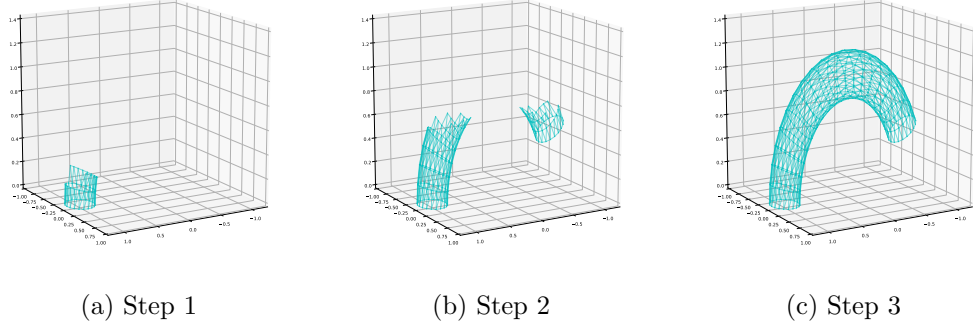


Figure 5: Example of filtration.

under \tilde{f} is independently retrieved by finding $\lambda_1, \dots, \lambda_{\beta_k^D}$ solving the linear system in the for-cycle of Algorithm 7.

This yields the desired map in the form $\tilde{f}_k([h_i^C]) = \sum_{j=1}^{\beta_k^D} \lambda_j(i)[h_j^D]$. \square

Algorithm 7: Induced map between homology spaces

Input: Chain map $f_k : C_k \rightarrow D_k$, representatives cycles $h_1^C, \dots, h_{\beta_k^C}^C$ of a basis of $H_k(C)$, β_k^D and $\{h_1^D, \dots, h_{\beta_k^D}^D, b_1^D, \dots, b_r^D\}$ output of Algorithm 6 for D ;

Result: map $\tilde{f}_k : H_k(C) \rightarrow H_k(D)$ induced by f_k .

$\tilde{f}_k :=$ zero matrix $\beta_k^D \times \beta_k^C$;

for $i = 1, \dots, \beta_k^C$ **do**

Solve $f_k(h_i^C) = \sum_{j=1}^{\beta_k^D} \lambda_j h_j^D + \sum_{l=1}^r \mu_l b_l$;
 $\tilde{f}_k[i] = (\lambda_1, \dots, \lambda_{\beta_k^D})^T$

end

return \tilde{f}_k

4.2 Constructing the persistent homology through harmonics

In this section, we present a construction of the k -persistent homology module $\{\mathcal{H}_k^i, \hat{f}_i\}_{i \in \mathbb{N}}$ for coefficients in \mathbb{R} through the space of k -harmonics. We call the persistence module $\{\mathcal{H}_k^i, \hat{f}_i\}_{i \in \mathbb{N}}$ the *harmonic persistence module*.

After some preliminaries on the Hodge Laplacian operator, by means of the Hodge decomposition (Formula 3), for each index $i \in \mathbb{N}$, we retrieve the persistence mod-

ule step of the harmonic persistence module. Afterwards, we introduce Algorithm 8 for retrieving the structure map \hat{f}_i as the map, at k -harmonics level, ensuring the isomorphism of persistence modules between the k -persistent homology module $\{H_k^i, \varphi_i\}$ and the harmonic persistence module $\{\mathcal{H}_k^i, \hat{f}_i\}$ proven in Theorem 5. Notice that this construction, as it was for the case of general coefficients, can be distributed and performed in parallel for each $i \in \mathbb{N}$.

The Hodge Laplacian operator

Since \mathbb{R} has characteristic equal to 0, given a chain complex

$$\dots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

we fix an inner product $\langle \cdot, \cdot \rangle_k$ on each space of k -chains C_k so that each operator ∂_k has a well-defined adjoint operator ∂_k^* , i.e., $\langle \partial_k(c), d \rangle_k = \langle c, \partial_k^*(d) \rangle_k$.

We consider the sequence C^*

$$\dots \xleftarrow{\partial_{k+2}^*} C_{k+1} \xleftarrow{\partial_{k+1}^*} C_k \xleftarrow{\partial_k^*} C_{k-1} \xleftarrow{\partial_{k-1}^*} \dots \xleftarrow{\partial_2^*} C_1 \xleftarrow{\partial_1^*} C_0 \xleftarrow{\partial_0^*} 0.$$

For $k \in \mathbb{N}$, the *Hodge Laplacian* in degree k (Laplacian, for short) is the linear operator on k -chains $L_k : C_k \rightarrow C_k$ given by

$$L_k := \partial_{k+1} \partial_{k+1}^* + \partial_k^* \partial_k. \quad (15)$$

Constructing the harmonic step in parallel

For each index $i \in \mathbb{N}$, the step in the harmonic persistence module is given by the space of k -harmonics. The space of k -harmonics of a chain complex is the subspace of C_k

$$\mathcal{H}_k := \ker(L_k). \quad (16)$$

Remark 5. If $h \in \mathcal{H}_k$, then $h \in \ker(\partial_k)$ and $h \in \ker(\partial_{k+1}^*)$.

Indeed, both $\partial_k^* \partial_k$ and $\partial_{k+1} \partial_{k+1}^*$ are positive semidefinite, as $\langle c, \partial_k^* \partial_k c \rangle_k = \langle \partial_k c, \partial_k c \rangle_k \geq 0$ and $\langle c, \partial_{k+1} \partial_{k+1}^* c \rangle_k = \langle \partial_{k+1}^* c, \partial_{k+1}^* c \rangle_k \geq 0$. Hence, L_k is positive semidefinite, and $L_k c = 0$ implies $\partial_k^* \partial_k c = 0$ and $\partial_{k+1} \partial_{k+1}^* c = 0$. Thus, $\langle \partial_k c, \partial_k c \rangle_k = 0$ and $\langle \partial_{k+1}^* c, \partial_{k+1}^* c \rangle_k = 0$.

Elements of $\ker(\partial_{k+1}^*)$ are called k -cocycles. We refer to [18] for more details.

Before moving to the computation of the structure maps in the harmonic persistence module, we discuss some useful results relative to the single step \mathcal{H}_k^i .

First of all, the Hodge decomposition theorem states that:

Theorem 3. For a chain complex C and for every natural k ,

$$C_k = \mathcal{H}_k \oplus \text{Im}(\partial_{k+1}) \oplus \text{Im}(\partial_k^*)$$

Moreover, this decomposition is orthogonal and $Z_k = \mathcal{H}_k \oplus \text{Im}(\partial_{k+1})$.

By Remark 5, we can consider the homology class $[h]$ of an element $h \in \mathcal{H}_k$. As per Hodge theory ([11, 18]), it holds

Theorem 4. The linear map $\psi_k : \mathcal{H}_k \rightarrow H_k$ defined by $\psi_k(h) = [h]$ is an isomorphism.

Given the Hodge decomposition there is the orthogonal projection $\pi_k : Z_k \rightarrow \mathcal{H}_k$ that is equal to the identity on \mathcal{H}_k and sends the elements of B_k to zero.

Lemma 4. For any k -cycle $z \in Z_k$, it holds $[z] = [\pi_k(z)]$ in H_k .

Proof. Any k -cycle z can be written uniquely as $h + b$, where $h \in \mathcal{H}_k$ and $b \in B_k$. Since π_k sends boundaries to zero and it is the identity on \mathcal{H}_k , it holds $z - \pi_k(z) = h + b - h = b$, and therefore z and $\pi_k(z)$ are in the same homology class. \square

Remark 6. If one chooses an orthonormal basis for the k -harmonics \mathcal{H}_k and represent it via the matrix V_k whose columns are the basis vectors, then the projection π_k is represented by the matrix V_k^T .

Computing the harmonic structure map in parallel

In order to define the persistent homology with respect to harmonic representatives, we focus on the behavior of the harmonic subspace under the action of chain maps. On the one hand, we need to remark the following.

Remark 7. A chain map $f : C \rightarrow D$ does not restrict to a map between the harmonic subspaces \mathcal{H}_k^C and \mathcal{H}_k^D .

Indeed, given an element $h \in \mathcal{H}_k^C$, the k -cycle $f(h)$ is not necessarily in \mathcal{H}_k^D . More precisely, $f(h)$ is necessarily a k -cycle but not necessarily a k -cocycle.

On the other hand, we can define a map $\hat{f}_k : \mathcal{H}_k^C \rightarrow \mathcal{H}_k^D$ in the following way. Consider the natural inclusion $i_k^C : \mathcal{H}_k^C \rightarrow C_k$ and the projection $\pi_k^D : D_k \rightarrow \mathcal{H}_k^D$. We define

$$\hat{f}_k = \pi_k^D f_k i_k^C. \quad (17)$$

Notice that the above definition corresponds to the general quotient map construction in Equation (27) of the Appendix A.

Algorithm 8 computes Equation (17) by applying Remark 6 the domain C and the target D to represent maps i_k^C and π_k^D .

Consider now a sequence of chain complexes connected by linear maps as in the sequence (12). We get induced the following sequence of linear maps between k -harmonic spaces

$$0 \xrightarrow{\hat{f}_{k,(0)}} \mathcal{H}_{k,(1)} \xrightarrow{\hat{f}_{k,(1)}} \dots \longrightarrow \dots \xrightarrow{\hat{f}_{k,(i-1)}} \mathcal{H}_{k,(i)} \xrightarrow{\hat{f}_{k,(i)}} \dots \xrightarrow{\hat{f}_{k,(n-1)}} \mathcal{H}_{k,(n)}.$$

Theorem 5. *The persistent homologies $\{H_{k,(i)}, \tilde{f}_{k,(i)}\}_{i=0}^n$ and $\{\mathcal{H}_{k,(i)}, \hat{f}_{k,(i)}\}_{i=0}^n$ are isomorphic as persistence modules with coefficients in \mathbb{R} ,*

Proof. It is enough to show that, for any chain map $f : C \rightarrow D$, the following diagram commutes

$$\begin{array}{ccc} \mathcal{H}_k^C & \xrightarrow{\hat{f}_k} & \mathcal{H}_k^D \\ \downarrow \psi_k^C & & \downarrow \psi_k^D \\ H_p^C & \xrightarrow{\tilde{f}_k} & H_k^D. \end{array}$$

That is, for any $h \in \mathcal{H}_k^C$, it must be $\tilde{f}_k(\psi_k^C(h)) = \psi_k^D(\hat{f}_k(h))$. By the definition of \tilde{f}_k and \hat{f}_k the equality becomes $[f_k(h)] = [\pi_k^D(f_k(h))]$. Since $f_k(h)$ belongs to Z_k^D , the statements follows from Lemma 4. □

Algorithm 8: Induced map between Laplacian kernels

Input: Chain map $f_k : C_k(C) \rightarrow C_k(D)$, V_C basis of $\ker(L_k(C_k(C)))$, V_D basis of $\ker(L_k(C_k(D)))$;

Result: Matrix Φ representing $\hat{f}_k : \mathcal{H}_k(C) \rightarrow \mathcal{H}_k(D)$

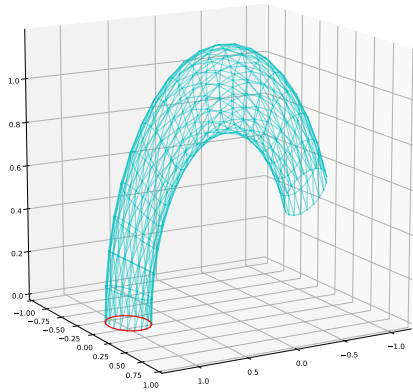
$\Phi = V_D^T \cdot f_k \cdot V_C$;

return (Φ_i)

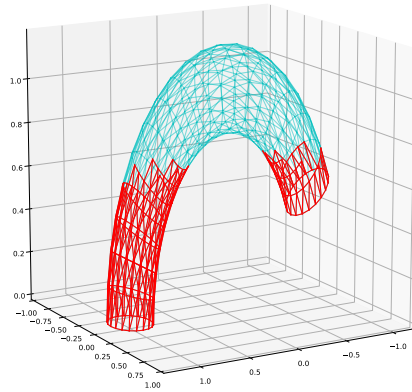
5 Conclusions and Future Works

In this work, we have introduced an algorithm for decomposing any persistence module, not necessarily a persistent homology module, into interval modules. We have discussed our method as supporting parallel and distributed implementations. We have presented it into two versions: a general one and a specific one for the case of real coefficients where one can take advantage of the SVD matrix decomposition.

We have introduced the notion of interval basis as the persistence module analogue for the notion of minimal homogeneous generators of a graded module with



(a) Generator 1



(b) Generator 2

Figure 6: Generators, classical algorithm.

coefficients in $\mathbb{F}[x]$. We have provided a proof of correctness of our approach which can be seen as an alternative and constructive proof of the Structure Theorem.

Afterwards, we have made explicit how to integrate our approach into the persistent homology pipeline, the main computational framework in TDA. Indeed, we have practically indicated how to obtain a persistent homology module out of a monotone sequence of chain complexes. Above all, we have remarked that each step and structure map in the module can be obtained independently, thus being suitable for parallel and distributed approaches.

Such an integration has offered interesting insights to be investigated further. For instance, it has made possible to geometrically locate the interval basis vectors onto a filtration of simplicial complexes. We have discussed simple examples to make comparisons with the classical generators obtained through the reduction algorithm [5]. We believe that the descriptive power of the interval basis deserves further study since it encodes implicitly the relations among evolving homology classes along a filtration.

As a last point to be addressed in the future, we have seen how working at persistence module level might be favorable for dealing with the persistence of harmonics. In particular, we have discussed how to overcome the non-functoriality of the harmonic forms by working at persistence module level. From the computational complexity point of view, we have shown how, for a persistence module,

both our decomposition and our construction procedures can take advantage of having real coefficients. From the geometrical point of view, we have shown how the interval basis choice for generators applies to the harmonic case. Hence, our work contributes in combining harmonic generators into the persistent homology framework.

Acknowledgements

The authors acknowledge the support from the Italian MIUR Award “Dipartimento di Eccellenza 2018-2022” - CUP: E11G18000350001 and the SmartData@PoliTO center for Big Data and Machine Learning.

References

- [1] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. “Clear and Compress: Computing Persistent Homology in Chunks”. In: *Topological Methods in Data Analysis and Visualization III*. Mathematics and Visualization. Springer, Cham, 2014, pp. 103–117.
- [2] Gunnar Carlsson. “Topology and data”. In: *Bulletin of the American Mathematical Society* 46.2 (2009), pp. 255–308.
- [3] Gunnar Carlsson, Anjan Dwaraknath, and Bradley J. Nelson. “Persistent and Zigzag Homology: A Matrix Factorization Viewpoint”. In: *arXiv preprint* (Mar. 2021). arXiv: 1911.10693.
- [4] Gunnar Carlsson and Vin de Silva. “Zigzag Persistence”. In: *Foundations of Computational Mathematics* 10.4 (2010), pp. 367–405.
- [5] Gunnar Carlsson and Afra Zomorodian. “Computing Persistent Homology”. In: *Discrete & Computational Geometry* 33.2 (2005), pp. 249–274.
- [6] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. Springer, 2016.
- [7] Cecil Jose A. Delfinado and Herbert Edelsbrunner. “An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere”. In: *Computer Aided Geometric Design* 12.7 (1995), pp. 771–784.
- [8] Tamal K. Dey, Fengtao Fan, and Yusu Wang. “Computing topological persistence for simplicial maps”. In: *Proceedings of the Annual Symposium on Computational Geometry*. Association for Computing Machinery, 2014, pp. 345–354.
- [9] Tamal K Dey, Tianqi Li, and Yusu Wang. “Efficient algorithms for computing a minimal homology basis”. In: *Latin American Symposium on Theoretical Informatics*. Springer. 2018, pp. 376–398.
- [10] Stefania Ebli and Gard Spreemann. “A notion of harmonic clustering in simplicial complexes”. In: *Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019*. Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 1083–1090.
- [11] Beno Eckmann. “Harmonische Funktionen und Randwertaufgaben in einem Komplex”. In: *Commentarii Mathematici Helvetici* 17.1 (1944), pp. 240–255.

- [12] Herbert Edelsbrunner and John Harer. “Persistent homology—a survey”. In: *Surveys on Discrete and Computational Geometry*. Vol. 453. Contemporary Mathematics. providence, RI: American Mathematical Society, 2008, pp. 257–282.
- [13] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. “Topological persistence and simplification”. In: *Discrete and Computational Geometry* 28.4 (2002), pp. 511–533.
- [14] Ulderico Fugacci, Sara Scaramuccia, Federico Iuricich, and Leila De Florian. “Persistent homology: a step-by-step introduction for newcomers”. In: *STAG: Smart Tools and Apps in Computer Graphics*. Eurographics Proceedings, 2016, pp. 1–10.
- [15] Robert Ghrist. “Barcodes: The persistent topology of data”. In: *Bulletin of the American Mathematical Society* 45.1 (2008), pp. 61–75.
- [16] Marco Guerra, Alessandro De Gregorio, Ulderico Fugacci, Giovanni Petri, and Francesco Vaccarino. “Homological scaffold via minimal homology bases”. In: *Scientific reports* 11.1 (Mar. 2021), p. 5355.
- [17] Allen Hatcher. *Algebraic Topology*. Vol. 60. Cornell University, New York, 2002, pp. 147–288.
- [18] Danijela Horak and Jürgen Jost. “Spectra of combinatorial Laplace operators on simplicial complexes”. In: *Advances in Mathematics* 244 (2013), pp. 303–336.
- [19] Sara Kališnik, Vitaliy Kurlin, and Davorin Lešnik. “A higher-dimensional homologically persistent skeleton”. In: *Advances in Applied Mathematics* 102 (2019), pp. 113–142.
- [20] Michael Kerber and Hannah Schreiber. “Barcodes of Towers and a Streaming Algorithm for Persistent Homology”. In: *Discrete and Computational Geometry* 61.4 (2019), pp. 852–879.
- [21] Dimitry Kozlov. *Combinatorial algebraic topology*. Vol. 21. Springer Science & Business Media, 2007.
- [22] Saunders Mac Lane. *Categories for the Working Mathematician*. Vol. 5. Graduate Texts in Mathematics 2. Springer-Verlag New York, 1998.
- [23] James R. Munkres. *Elements of algebraic topology*. CRC Press, 2018.
- [24] Alice Patania, Francesco Vaccarino, and Giovanni Petri. “Topological analysis of data”. In: *EPJ Data Science* 6.1 (2017).

- [25] Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. “Persistent spectral graph”. In: *International Journal for Numerical Methods in Biomedical Engineering* 36.9 (2020), e3376.
- [26] Rui Wang, Rundong Zhao, Emily Ribando-Gros, Jiahui Chen, Yiyong Tong, and Guo-Wei Wei. “HERMES: Persistent spectral graph software”. In: *Foundations of Data Science* 3.1 (Mar. 2021), p. 67.

A Appendix - Categorical Aspects

A.1 Notes on linear maps of short exact sequences in the category of finite vector spaces

In this section, all spaces are finite vector spaces and all maps are linear maps.

Short exact sequences

A *short exact sequence* (s.e.s., for short) is a sequence of maps

$$0 \longrightarrow K \xrightarrow{s} A \xrightarrow{q} Q \longrightarrow 0 \quad (18)$$

such that, for each space in the sequence, the image of the map targeting the space and the kernel of the map which follows are equal.

In the category of finite vector spaces, a quotient space $Q = A/\sim$ of a space A induces an s.e.s. where K is the kernel of the quotient relation \sim , the map s is the natural injection of K into A , q is the natural projection from A to A/\sim . Since the category of finite vector spaces with linear maps is known to be an Abelian category (see Chapter VIII in [22]), the category admits kernel representations for surjective maps and quotient representations for injective maps. Hence, up to isomorphisms, a short exact sequence represents a quotient space. This motivates us in calling, given an s.e.s. such as the one in (18), the space Q the quotient space, and the space K the kernel space of the s.e.s.

Example 2. An example of a short exact sequence is given by the k -homology space H_k as a quotient of the k -cycle space Z_k with kernel of the quotient relation given by the k -boundary space B_k :

$$0 \longrightarrow B_k \hookrightarrow Z_k \xrightarrow{[\cdot]} H_k \longrightarrow 0,$$

where the arrow \hookrightarrow denotes the natural injection and the map $[\cdot]$ is the natural quotient projection defining the k -homology space (see Section 4).

Example 3. Another relevant short exact sequence is the one involving the space of k -harmonics \mathcal{H}_k introduced in Section 4.2. Indeed, the Theorem 4, as a corollary of the Hodge Decomposition Theorem 3, and the definition of the projection map π_k satisfying Lemma 4 ensure the exactness of the following sequence

$$0 \longrightarrow B_k \hookrightarrow Z_k \xrightarrow{\pi_k} \mathcal{H}_k \longrightarrow 0. \quad (19)$$

Consider the two short exact sequences as in the rows of the following diagram,

$$\begin{array}{ccccccccc}
0 & \longrightarrow & K & \xrightarrow{s} & A & \xrightarrow{q} & Q & \longrightarrow & 0 \\
& & \downarrow \varepsilon_0 & & \downarrow \varepsilon_1 & & \downarrow \varepsilon_2 & & \\
0 & \longrightarrow & K' & \xrightarrow{s'} & A' & \xrightarrow{q'} & Q' & \longrightarrow & 0,
\end{array} \tag{20}$$

a *map of short exact sequences* is a tuple $(\varepsilon_0, \varepsilon_1, \varepsilon_2)$ of maps such that the diagram in (20) commutes. An *isomorphism* of short exact sequences is a map of short sequences where each map is an isomorphism.

Lemma 5 (Splitting [17]). *For a short exact sequence*

$$0 \longrightarrow K \xrightarrow{s} A \xrightarrow{q} Q \longrightarrow 0,$$

the following statements are equivalent:

- a) *there exists a surjective map $p : A \longrightarrow K$ such that $ps = id_K : K \longrightarrow K$*
- b) *there exists an injective map $r : Q \longrightarrow A$ such that $qr = id_Q : Q \longrightarrow Q$*
- c) *there exists an isomorphism $K \oplus Q \xrightarrow{h} A$ such that (id_Q, h, id_Q) is an isomorphism of short exact sequences*

$$\begin{array}{ccccccccc}
0 & \longrightarrow & K & \xrightarrow{s} & A & \xrightarrow{q} & Q & \longrightarrow & 0 \\
& & \uparrow id_K & & \uparrow h & & \uparrow id_Q & & \\
0 & \longrightarrow & K & \xrightarrow{incl_1} & K \oplus Q & \xrightarrow{proj_2} & Q & \longrightarrow & 0,
\end{array} \tag{21}$$

where morphisms $incl_1$, $proj_2$ are the canonical inclusion and projection of the direct sum \oplus .

Remark 8. In the category of finite vector spaces with linear maps, each short exact sequence admits a splitting, i.e. the sequence in the bottom row in (21) is a *split sequence*.

Indeed, this follows by noticing that any set of linearly independent vectors in a vector space can be extended to a basis.

Maps induced on quotient spaces by representatives

With respect to the notation in Diagram (20), we say that a map $\varepsilon : A \longrightarrow A'$ is *compatible with the short exact sequences* if it satisfies $\varepsilon(s(A)) \subseteq s'(A')$. In other words, a compatible map preserves kernels between the domain and the target sequences so that it is possible to induce maps on kernels and quotient spaces.

Remark 9. If a map $\varepsilon : A \longrightarrow A'$ is compatible with the short exact sequences in Diagram (20), then there exists only one map of short exact sequences $(\varepsilon_{\downarrow}, \varepsilon, \tilde{\varepsilon})$ where

$$\varepsilon_{\downarrow} = p'f_s, \quad \text{with } p' \text{ as in the splitting of case a)} \quad (22)$$

$$\tilde{\varepsilon} = q'fr, \quad \text{with } r \text{ as in the splitting of case b)}. \quad (23)$$

Indeed, the maps ε_{\downarrow} and $\tilde{\varepsilon}$ are well defined and unique by the compatibility assumption. As for the existence, it is ensured by Remark 8.

Example 4. An example of a map compatible with two short exact sequences is given by the map f_k in the degree k of a chain map (see Section 4). By homology functoriality, we have the following commutative diagram

$$\begin{array}{ccccccc} 0 & \longrightarrow & B_k & \hookrightarrow & Z_k & \xrightarrow{[\cdot]} & H_k & \longrightarrow & 0 \\ & & \downarrow f_{\downarrow k} & & \downarrow f_k & & \downarrow \tilde{f}_k & & \\ 0 & \longrightarrow & B'_k & \hookrightarrow & Z'_k & \xrightarrow{[\cdot]'} & H'_k & \longrightarrow & 0, \end{array} \quad (24)$$

where maps f_{\downarrow} and \tilde{f} satisfy Equation (22) and (23), respectively.

Indeed, homology functoriality depends on the map f_k preserving k -cycles and k -boundaries. In particular for the map \tilde{f} between quotient spaces, a choice of homology representative corresponds to a splitting map $r_k : H_k \longrightarrow Z_k$ satisfying case b) in Lemma 5.

This gives us a way of retrieving the map \tilde{f}_k as

$$\tilde{f}_k = [\cdot]' f_k r_k \quad (25)$$

for some choice r_k of homology representatives, i.e., such that $[\cdot]r_k = \text{id}_{H_k}$.

Example 5. Moreover, by homology functoriality, we get that the map f_k is compatible with two short sequences like in Diagram (19) where we substitute the k -homology space with the k -harmonic space \mathcal{H}_k and where, this time, we call \hat{f}_k the map induced between quotient spaces

$$\begin{array}{ccccccc} 0 & \longrightarrow & B_k & \hookrightarrow & Z_k & \xrightarrow{\pi_k} & \mathcal{H}_k & \longrightarrow & 0 \\ & & \downarrow f_{\downarrow k} & & \downarrow f_k & & \downarrow \hat{f}_k & & \\ 0 & \longrightarrow & B'_k & \hookrightarrow & Z'_k & \xrightarrow{\pi'_k} & \mathcal{H}'_k & \longrightarrow & 0. \end{array} \quad (26)$$

Indeed, the compatibility condition on the map f_k depends on the natural inclusions $B_k \subseteq Z_k$ whereas the exactness of rows has already been discussed.

In particular, given a basis for the space of k -harmonics as a map $r_k : \mathcal{H}_k \rightarrow Z_k$ as in the splitting of case b) in Lemma 5, we can express the map induced at harmonics level as follows

$$\hat{f}_k = \pi'_k f_k r_k. \quad (27)$$

Maps induced on quotient spaces by split sequences

As a last remark, we express the Equations (23) and (22) of the maps induced on quotient and kernel terms in the case of split sequences satisfying case c) in Lemma 5 with respect to two given short exact sequences with the notation of the rows in Diagram (18). Let ε be a map compatible with the two original short exact sequences with maps ε_{\downarrow} and $\tilde{\varepsilon}$ induced on kernel and quotient spaces, respectively. We claim that the map ε induces a map ε^* , that we call *direct sum map*, compatible with the two split sequences of case c) associated to the domain and target sequences

$$\begin{array}{ccccccc} 0 & \longrightarrow & K & \xrightarrow{\text{incl}_1} & K \oplus Q & \xrightarrow{\text{proj}_1} & Q \longrightarrow 0 \\ & & \downarrow \varepsilon_{\downarrow} & & \downarrow \varepsilon^* & & \downarrow \tilde{\varepsilon} \\ 0 & \longrightarrow & K' & \xrightarrow{\text{incl}_1} & K' \oplus Q' & \xrightarrow{\text{proj}_1} & Q' \longrightarrow 0. \end{array} \quad (28)$$

Indeed by assumption, we have two isomorphisms $h : K \oplus Q \rightarrow A$ such that $qh = \text{proj}_2$ and $h' : K' \oplus Q' \rightarrow A'$ such that $q'h' = \text{proj}_2$. We define $\varepsilon^* = h'^{-1}\varepsilon h$. It is easy to check that conditions $qh = \text{proj}_2$ and $q'h' = \text{proj}_2$ expressing the commutativity of Diagram (21) ensure that the maps ε_{\downarrow} and $\tilde{\varepsilon}$ are the same as in Diagram (20). Hence the following remark holds.

Remark 10. Given the direct sum map ε^* of a compatible map ε inducing maps ε_{\downarrow} and $\tilde{\varepsilon}$ between kernel and quotient spaces respectively, we get that

$$\tilde{\varepsilon} = \text{proj}_2 \varepsilon^* \text{incl}_2 \quad (29)$$

$$\varepsilon_{\downarrow} = \text{proj}_1 \varepsilon^* \text{incl}_1, \quad (30)$$

where the tuple $(\varepsilon_{\downarrow}, \varepsilon^*, \tilde{\varepsilon})$ defines a map of short exact split sequences.

B Appendix - Preliminaries on simplicial complexes and persistent homology

In this appendix, we recall some basics in Simplicial Homology [17, 21, 23].

Our purpose is that of moving from the purely algebraic setting presented so far along the paper to the main input representation within the persistent homology pipeline, namely the notion of *simplicial complex*. (Geometric) simplicial complexes allow for representing geometric shapes into combinatorial and discrete terms. This way, a simplicial complex captures the topology of a shape into computable terms.

Here, we start by introducing the (abstract) definition of a simplicial complex. The relevance of the abstract point of view derives from its descriptive power. Indeed, not all datasets come associated with an embedding into a coordinate space. Nonetheless, those datasets might be endowed with the purely combinatorial structure of an abstract simplicial complex where the topology is given by pairwise and higher-order relations among data. The reader interested in the persistent homology setting is referred to the surveys [2, 12, 14, 15, 24].

Abstract simplicial complexes

An *abstract simplicial complex* Σ on a finite set V is a subset of the power set of V , with the property of being closed under restriction. An element of Σ is called a *simplex* and if $\sigma \in \Sigma$, $\tau \subseteq \sigma$ then $\tau \in \Sigma$. Elements of V are usually called *vertices*. Simplices of cardinality $p + 1$ are called p -simplices. We also say a p -simplex has *dimension* p . We call the p -skeleton of Σ the set of simplices of Σ of dimension $\leq p$. If $\tau \subseteq \sigma$ we say that τ is a face of σ and σ is a coface of τ . The dimension of a simplicial complex is defined as $\dim \Sigma := \max\{\dim \sigma / \sigma \in \Sigma\}$.

Geometric simplicial complexes

Simplicial complexes have a natural geometric interpretation: a (geometric) p -simplex σ is the convex hull of $p + 1$ affinely independent points in the Euclidean space. As such, a 0-simplex is a point, a 1-simplex an edge, a 2-simplex a triangle, a 3-simplex a tetrahedron, and so on. Every abstract simplicial complex has a geometric realization (and actually many of them), i.e. a geometric simplicial complex to which it is homeomorphic. A realization of an abstract complex is denoted by $|\Sigma|$.

From simplicial complexes to chain complexes

For Σ a simplicial complex and \mathbb{F} any field, the k^{th} -chain space $C_k(\Sigma; \mathbb{F})$ of Σ is defined as the \mathbb{F} -vector space generated by the oriented k -simplices of Σ and such that $-\sigma$ coincides with the opposite orientation on σ . When clear from the context, we drop the indication of the base field to ease the notation. An element of $C_k(\Sigma)$ is called k -chain: it is a finite linear combination $\sum_i \eta_i [\sigma_i]$ with $\eta_i \in \mathbb{F}$.

Up to fixing an orientation for each k -simplex when $k > 0$, and instead naturally when $k = 0$, a canonical basis of $C_k(\Sigma)$ is given by the k -chains corresponding to the individual oriented k -simplices of Σ .

Chain spaces are instrumental in introducing a linear operator that expresses the above-mentioned adjacency relations across dimensions. The *boundary operator* is a linear map $\partial_k : C_k(\Sigma) \rightarrow C_{k-1}(\Sigma)$ that on an element of canonical basis corresponding to the oriented k -simplex $\sigma = [v_0, \dots, v_k]$, is defined as:

$$\partial_k(\sigma) := \sum_{i=0}^k (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_k], \quad (31)$$

where \hat{v}_i means that vertex v_i is omitted. It extends to the whole chain space by linearity.

Example 6. The collection $C_\bullet = \{C_k\}_{k=0}^\infty$ along with the collection of boundary operators ∂_\bullet defines a chain complex like the one introduced in Section 4. Indeed, it is easy to check that $\partial_k \partial_{k+1} = 0$, for all integers k .

Simplicial complexes and persistence modules

A simplicial complex Σ can be made into a *filtration of simplicial complexes* (or a *filtered simplicial complex*) by taking a finite sequence of subcomplexes $\Sigma_0 \subseteq \Sigma_1 \subseteq \dots \subseteq \Sigma_p = \Sigma$, where a *subcomplex* is a subset and also a simplicial complex.

For any index $i = 0, \dots, p$, define $C_{(i)} = (C_{k,(i)}, \partial_{k,(i)})$ to be the chain complex associated with the simplicial complex Σ_i in the filtered complex. Notice that each inclusion induces a linear chain map $\iota_{k,(i)} : C_{k,(i)} \rightarrow C_{k,(i+1)}$. Indeed, $\iota_{k,(i)} \partial_{k+1,k,(i)} = \partial_{k,(i)} \iota_{k+1,(i)}$. Hence, any filtered complex induces a (free) persistence module

$$0 \xrightarrow{\iota_{k,(0)}} C_{k,(1)} \xrightarrow{\iota_{k,(1)}} \dots \longrightarrow \dots \xrightarrow{\iota_{k,(i-1)}} C_{k,(i)} \xrightarrow{\iota_{k,(i)}} \dots \xrightarrow{\iota_{k,(n-1)}} C_{k,(n)}, \quad (32)$$

and by applying the homology construction we get the associated persistent homology module

$$0 \xrightarrow{\tilde{\iota}_{k,(0)}} H_{k,(1)} \xrightarrow{\tilde{\iota}_{k,(1)}} \dots \longrightarrow \dots \xrightarrow{\tilde{\iota}_{k,(i-1)}} H_{k,(i)} \xrightarrow{\tilde{\iota}_{k,(i)}} \dots \xrightarrow{\tilde{\iota}_{k,(n-1)}} H_{k,(n)}. \quad (33)$$

The sequences (32) and (33) correspond to the general sequences constructed in Formula (12) and (13) in Section 4 where the discussion was purely algebraic.