A Fast Algorithm for Computing Zigzag Representatives

Tamal K. Dey* Tao Hou† Dmitriy Morozov‡

Abstract

Zigzag filtrations of simplicial complexes generalize the usual filtrations by allowing simplex deletions in addition to simplex insertions. The barcodes computed from zigzag filtrations encode the evolution of homological features. Although one can locate a particular feature at any index in the filtration using existing algorithms, the resulting *representatives* may not be compatible with the zigzag: a representative cycle at one index may not map into a representative cycle at its neighbor. For this, one needs to compute compatible representative cycles along each bar in the barcode. Even though it is known that the barcode for a zigzag filtration with m insertions and deletions can be computed in $O(m^\omega)$ time, it is not known how to compute the compatible representatives so efficiently. For a non-zigzag filtration, the classical matrix-based algorithm provides representatives in $O(m^3)$ time, which can be improved to $O(m^\omega)$. However, no known algorithm for zigzag filtrations computes the representatives with the $O(m^3)$ time bound. We present an $O(m^2n)$ time algorithm for this problem, where $n \leq m$ is the size of the largest complex in the filtration.

1 Introduction

Persistent homology and its computation have been a central theme in topological data analysis (TDA) [6, 7, 13]. Using persistent homology, one computes a signature called a *barcode* from data which is presented in the form of a growing sequence of simplicial complexes called a *filtration*. However, the barcode itself does not provide an avenue to go back to the data. For that, we need to compute a representative for each *bar* (interval) in the barcode, that is, a cycle whose homology class exists exactly over the duration of the bar. In other words, we aim to compute the *interval modules* themselves in the interval decomposition [9] instead of only the intervals.

In this paper, we consider computing representatives for the bars where the given filtration is no longer monotonically growing but may also shrink, resulting in what is known as a *zigzag* filtration. A number of algorithms have been proposed for computing the barcode from a zigzag filtration [2, 5, 6, 10, 11, 12]. All of them maintain *pointwise representatives*, i.e., a homology basis for every step in the filtration, but they do not compute the *barcode representatives*, i.e., a set of compatible pointwise bases, where elements of one basis are matched to the elements of its neighbors (see Definition 2.3). Solving this problem is the main topic of this paper.

The barcode representatives are not readily available during the zigzag computation because basis updates at any point may require changes both in the future and in the past to maintain the matching. To make this precise, let m be the number of additions and deletions and n be the maximum size of complexes in a zigzag filtration. The challenge is rooted in the fact that a barcode representative for a zigzag filtration (henceforth also called a *zigzag representative*) may consist of $O(m)$ different cycles [10] for each of the $O(m)$ indices in a bar (see Definition 2.3). Consequently, the space complexity for the straightforward way of maintaining a zigzag representative is $O(mn)$. This is in contrast to a non-zigzag representative which consists of the same cycle over the entire bar. One obvious way to obtain the zigzag representatives is to adapt the $O(mn^2)$ algorithm proposed by Maria and Oudot [10] which directly targets representatives. But then, the complexity increases to $O(m^2n^2)$, which stems from the need of summing two representatives each consisting of $O(m)$ cycles. In total these summations over the entire course of the algorithm incur an $O(m^2n^2)$ cost. To see this, notice that the algorithm in [10] is based on summations of bars (and their representatives) where each bar is associated with a single cycle from the $O(m)$ cycles in its representative. The algorithm performs $O(mn)$ summations of bars and the associated cycles resulting in an $O(mn^2)$ complexity. To adapt this algorithm for computing representatives, one instead maintains the full representative consisting of $O(m)$ cycles for each bar. Because a summation of two bars now costs $O(mn)$ time, the $O(mn)$ bar summations in the algorithm [10] then result in an $O(m^2n^2)$ complexity.

*Department of Computer Science, Purdue University. tamaldey@purdue.edu

†Department of Computer Science, University of Oregon. taohou@uoregon.edu

‡Lawrence Berkeley National Laboratory. dmitriy@mrzv.org

It has remained tantalizingly difficult to design an algorithm that brings down the theoretical complexity to $O(m^3)$, matching the complexity for non-zigzag filtrations [3, 12], while remaining practical. As mentioned already, the bottleneck of the computation lies in the summation of two representatives each consisting of $O(m)$ cycles. In this paper, we present an $O(m^2n)$ algorithm which overcomes the bottleneck by compressing the representatives into a more compact form each taking only $O(m)$ space instead of $O(mn)$ space. A preliminary implementation of our $O(m^2n)$ algorithm shows its practicality (see Section 5).

Figure 1: an illustrative example. The compression of representatives in our algorithm is made possible by adopting some novel constructs for computing zigzag persistence whose ideas are illustrated in Figure 1 (see also the beginning of Section 3 for more explanations; formal definitions of concepts mentioned below are provided in Section 2):

- First, we observe that the barcode of the regular (homology) zigzag module interconnects with the barcode of another module, namely, the *boundary* zigzag module, which arises out of the boundary groups for complexes in the input zigzag filtration. To see the interconnection, let $\boldsymbol{z}$ denote the bold cycle in K_2 (and its continuation in the complexes K_3–K_5) in Figure 1. In Figure 1 (top), the bar $[2, 2]$ for the homology module born at K_2 and dying entering K_3 (hence drawn as an orange dot at index 2) interconnects with the bar $[3, 5]$ for the boundary module born at K_3 and dying entering K_6 as the cycle $\boldsymbol{z}$ representing the bar $[2, 2]$ becomes a boundary in K_3. The bar $[3, 5]$, which is also represented by $\boldsymbol{z}$, in turn interconnects with the bar $[6, 6]$ for the homology module as $\boldsymbol{z}$ becomes a non-boundary at K_6.

- Second, we observe that the seamless transition between barcodes of the two modules allows us to define a construct called *wires* each of which is a single cycle with a fixed birth index, presumably extending indefinitely to infinity. A wire may be a boundary cycle (thus called a *boundary* wire) with its birth index coinciding with a birth in the boundary module, or a non-boundary cycle (thus called a *non-boundary* wire) with its birth index coinciding with a birth in the homology module. For the example in Figure 1, we have three non-boundary wires (orange) and two boundary wires (blue) subscripted by the birth indices with respective cycles also being illustrated.

  A collection of such wires forms what we call a *bundle* for a zigzag bar. In Figure 1, we show the bundle for the longest bar $\mathcal{b} = [1, 7]$. One surprising fact we find is that representative cycles of a bar can be recovered from index-wise summations of the wire cycles in its bundle even though a wire cycle involved in the summation may not be present in each complex over the bar (see Section 3). Figure 1 (bottom) shows the representative cycles of the bar $\mathcal{b}$ obtained by summing three wires $\{w_1, w_3, w_4\}$ even though the cycles for wires w_1, w_4 are not present in K_5–K_7.

  At each index in the filtration, there can be no more than one wire with birth at that index. Hence, each bundle is represented as a set of $O(m)$ wire indices in our algorithm. The summations among the bundles are then less costly and can be done in $O(m)$ time because each entails doing a symmetric sum among $O(m)$ wire indices rather than the actual $O(m)$ cycles. When a bar is completed, its actual representative is read from summing the cycles in its bundle. Wires and bundles allow our algorithm to have a space complexity of $O(mn)$ whereas the algorithm for computing representatives adapted from [10] has a space complexity of $O(mn^2)$.

Our compression using wires is also made possible by adopting a new way of computing zigzag barcodes, which processes the filtration from left to right similar to the algorithm in [2] but directly targets maintaining the zigzag representatives over the course of the computation. This is also in contrast to the other representative-based algorithm [10] which always maintains a reversed non-zigzag filtration at the end. Section 3 briefly describes the idea.

2 Core definitions

Throughout, we assume a *simplex-wise zigzag filtration* $\mathcal{F}$ as input to our algorithm:

$$\mathcal{F} : \varnothing = K_0 \xleftrightarrow{\sigma_0} K_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{m-1}} K_m, \tag{2.1}$$

in which each K_i is a simplicial complex and each arrow $K_i \xleftrightarrow{\sigma_i} K_{i+1}$ is either a forward inclusion $K_i \xhookrightarrow{\sigma_i} K_{i+1}$ (an addition of a simplex σ_i) or a backward one $K_i \xhookleftarrow{\sigma_i} K_{i+1}$ (a deletion of a simplex σ_i). Notice that assuming

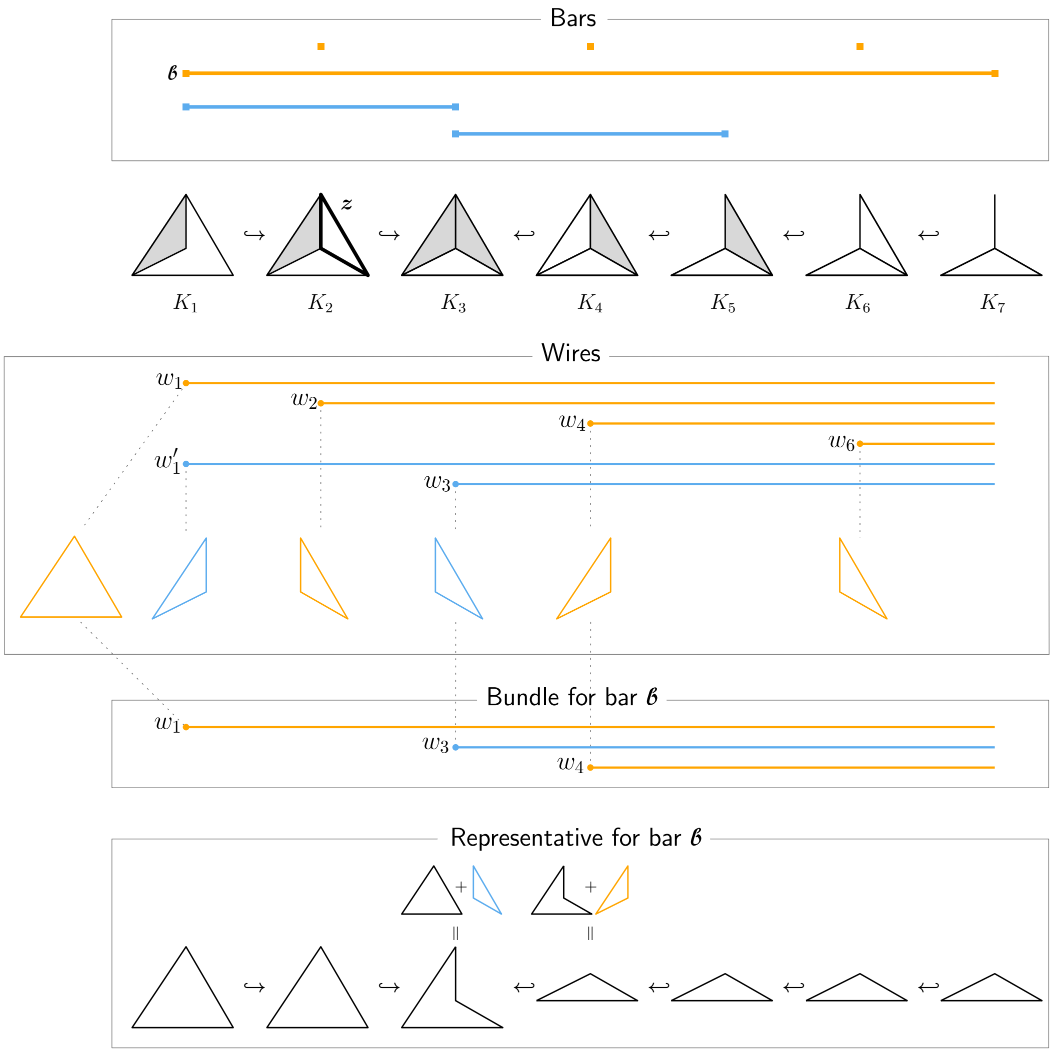

Figure 1: An example of wires, bundles, and the boundary zigzag module which are major constructs leading to the $O(m^2n)$ algorithm. Orange and blue colors are used for the constructs of homology and boundary zigzag modules respectively.

$\mathcal{F}$ to be simplex-wise and $K_0 = \varnothing$ is a standard practice in the computation of non-zigzag persistence [8] and its zigzag version [2, 10]. Also notice that any zigzag filtration in general can be converted into a simplex-wise version, and the representatives computed for this simplex-wise version can also be easily mapped to the ones for the original filtration. We let $\mathcal{F}_i$ denote the part of $\mathcal{F}$ up to index i, that is,

$$\mathcal{F}_i : \varnothing = K_0 \xleftrightarrow{\sigma_0} K_1 \xleftrightarrow{\sigma_1} \cdots \xleftrightarrow{\sigma_{i-1}} K_i. \tag{2.2}$$

Notice that $\mathcal{F} = \mathcal{F}_m$. The *total complex* $\overline{K}$ of $\mathcal{F}$ is the union of all complexes in $\mathcal{F}$. Let n be the maximum size of complexes in $\mathcal{F}$ (note that generally n is not equal to the size of $\overline{K}$).

For a complex K_i, we consider its homology group $\mathsf{H}(K_i)$ (with $\mathbb{Z}_2$ coefficients) *over all degrees*, which is the direct sum of $\mathsf{H}_p(K_i)$ for all p (so that the dimension of $\mathsf{H}(K_i)$ equals the sum of the dimensions of all $\mathsf{H}_p(K_i)$'s). Accordingly, $\mathsf{C}(K_i)$, $\mathsf{Z}(K_i)$, and $\mathsf{B}(K_i)$ denote the chain, cycle, and boundary groups of K_i over all degrees respectively. Since we take $\mathbb{Z}_2$ as coefficients, chains or cycles in this paper are also treated as sets of simplices. We also consider any chain $c \in \mathsf{C}(K_i)$ to be a chain in $\overline{K}$ in general and do not differentiate the same simplex appearing in different complexes in $\mathcal{F}$. For example, suppose that all simplices in $c \in \mathsf{C}(K_i)$ also belong to a K_j, we then have $c \in \mathsf{C}(K_j)$.

Taking the homology functor on $\mathcal{F}_i$ we obtain the following (homology) *zigzag module*:

$$\mathsf{H}(\mathcal{F}_i) : \mathsf{H}(K_0) \overset{\psi_0^*}{\longleftrightarrow} \mathsf{H}(K_1) \overset{\psi_1^*}{\longleftrightarrow} \cdots \overset{\psi_{m-1}^*}{\longleftrightarrow} \mathsf{H}(K_i).$$

Similarly, taking the boundary functor on $\mathcal{F}_i$ we obtain the (boundary) zigzag module:

$$\mathsf{B}(\mathcal{F}_i) : \mathsf{B}(K_0) \overset{\psi_0^\#}{\longleftrightarrow} \mathsf{B}(K_1) \overset{\psi_1^\#}{\longleftrightarrow} \cdots \overset{\psi_{m-1}^\#}{\longleftrightarrow} \mathsf{B}(K_i).$$

Each $\psi_j^* : \mathsf{H}(K_j) \leftrightarrow \mathsf{H}(K_{j+1})$ in $\mathsf{H}(\mathcal{F}_i)$ is a linear map induced by inclusion between homology groups whereas each $\psi_j^\# : \mathsf{B}(K_j) \leftrightarrow \mathsf{B}(K_{j+1})$ in $\mathsf{B}(\mathcal{F}_i)$ is an inclusion between chain groups. By [1, 9], for some index sets Λ_H and Λ_B, $\mathsf{H}(\mathcal{F}_i)$ and $\mathsf{B}(\mathcal{F}_i)$ have decompositions of the form

$$\mathsf{H}(\mathcal{F}_i) = \bigoplus_{k \in \Lambda_H} \mathcal{I}^{[b_k, d_k]} \quad \text{and} \quad \mathsf{B}(\mathcal{F}_i) = \bigoplus_{k \in \Lambda_B} \mathcal{I}^{[b_k, d_k]},$$

in which each $\mathcal{I}^{[b_k, d_k]}$ is an *interval module* over the interval $[b_k, d_k] \subseteq \{0, 1, \ldots, i\}$. The set of intervals $\mathsf{Pers}^H(\mathcal{F}_i) := \{[b_k, d_k] \mid k \in \Lambda_H\}$ for $\mathsf{H}(\mathcal{F}_i)$ and the set of intervals $\mathsf{Pers}^B(\mathcal{F}_i) := \{[b_k, d_k] \mid k \in \Lambda_B\}$ for $\mathsf{B}(\mathcal{F}_i)$ are called the *homology barcode* and *boundary barcode* of $\mathcal{F}_i$ respectively. In this paper, we introduce the computation of the intervals and representatives for $\mathsf{B}(\mathcal{F})$ as an integral part of the computation of those for $\mathsf{H}(\mathcal{F})$, which is critical to achieving the $O(m^2 n)$ complexity. We similarly define a barcode $\mathsf{Pers}_p^H(\mathcal{F}_i)$ for the module $\mathsf{H}_p(\mathcal{F}_i)$ over each degree p, so that $\mathsf{Pers}^H(\mathcal{F}_i) = \bigsqcup_p \mathsf{Pers}_p^H(\mathcal{F}_i)$. Notice that we can also define the barcode $\mathsf{Pers}_p^B(\mathcal{F}_i)$ where $\mathsf{Pers}^B(\mathcal{F}_i) = \bigsqcup_p \mathsf{Pers}_p^B(\mathcal{F}_i)$.

DEFINITION 2.1. (HOMOLOGY BIRTH/DEATH INDICES) *Since $\mathcal{F}_i$ is simplex-wise, each map ψ_j^* in $\mathsf{H}(\mathcal{F}_i)$ is either injective with a 1-dimensional cokernel or surjective with a 1-dimensional kernel but cannot be both. The set of* homology birth indices *of $\mathcal{F}_i$, denoted $\mathsf{P}^H(\mathcal{F}_i)$, and the set of* homology death indices *of $\mathcal{F}_i$, denoted $\mathsf{N}^H(\mathcal{F}_i)$, are constructively defined as follows: for each forward $\psi_j^* : \mathsf{H}(K_j) \to \mathsf{H}(K_{j+1})$, we add $j+1$ to $\mathsf{P}^H(\mathcal{F}_i)$ if ψ_j^* is injective and add j to $\mathsf{N}^H(\mathcal{F}_i)$ otherwise. Also, for each backward $\psi_j^* : \mathsf{H}(K_j) \leftarrow \mathsf{H}(K_{j+1})$, we add $j+1$ to $\mathsf{P}^H(\mathcal{F}_i)$ if ψ_j^* is surjective and add j to $\mathsf{N}^H(\mathcal{F}_i)$ otherwise. Finally, we add r copies of i to $\mathsf{N}^H(\mathcal{F}_i)$ where r is the dimension of $\mathsf{H}(K_i)$.*

REMARK 2.1. *Technically speaking, when we add r copies of i to $\mathsf{N}^H(\mathcal{F}_i)$, it becomes a multi-set.*

DEFINITION 2.2. (BOUNDARY BIRTH/DEATH INDICES) *Similarly as above, we define the* boundary birth indices *$\mathsf{P}^B(\mathcal{F}_i)$ and* boundary death indices *$\mathsf{N}^B(\mathcal{F}_i)$ of $\mathcal{F}_i$ by considering the module $\mathsf{B}(\mathcal{F}_i)$. Notice that $\psi_j^\#$ is always injective. So, for each forward $\psi_j^\# : \mathsf{B}(K_j) \to \mathsf{B}(K_{j+1})$ that is not surjective, we add $j+1$ to $\mathsf{P}^B(\mathcal{F}_i)$. Also, for each backward $\psi_j^\# : \mathsf{B}(K_j) \leftarrow \mathsf{B}(K_{j+1})$ that is not surjective, we add j to $\mathsf{N}^B(\mathcal{F}_i)$. Finally, we add q copies of i to $\mathsf{N}^B(\mathcal{F}_i)$ where q is the dimension of $\mathsf{B}(K_i)$.*

Whenever ψ_j^* is injective, $\psi_j^\#$ is an identity map; whenever ψ_j^* is surjective, $\psi_j^\#$ is not surjective. Hence, $\mathsf{P}^H(\mathcal{F}_i) \cap \mathsf{P}^B(\mathcal{F}_i) = \varnothing$ while (different copies of) i could belong to both $\mathsf{N}^H(\mathcal{F}_i)$ and $\mathsf{N}^B(\mathcal{F}_i)$. Also notice that $[b, d] \in \mathsf{Pers}^H(\mathcal{F}_i)$ implies that $b \in \mathsf{P}^H(\mathcal{F}_i)$ and $d \in \mathsf{N}^H(\mathcal{F}_i)$ (similar facts hold for $[b, d] \in \mathsf{Pers}^B(\mathcal{F}_i)$). We provide the definition of homology representatives (see Maria and Oudot [10]) as follows and then adapt it to define boundary representatives:

DEFINITION 2.3. (HOMOLOGY REPRESENTATIVES) *Consider a filtration $\mathcal{F}_i$ and let $[b, d] \subseteq [0, i]$ be an interval where $b \in \mathsf{P}^H(\mathcal{F}_i)$ (notice that $b > 0$ because $K_0 = \varnothing$ by assumption) and $d \in \mathsf{N}^H(\mathcal{F}_i)$. A sequence of cycles $\mathsf{rep} = \{z_\alpha \in \mathsf{Z}(K_\alpha) \mid \alpha \in [b, d]\}$ is called a* homology representative *(or simply* representative*) for $[b, d]$ if for every $b \le \alpha < d$, either $\psi_\alpha^*([z_\alpha]) = [z_{\alpha+1}]$ or $\psi_\alpha^*([z_{\alpha+1}]) = [z_\alpha]$ based on the direction of ψ_α^*. Furthermore, we have:*

Birth condition: *If $\psi_{b-1}^* : \mathsf{H}(K_{b-1}) \to \mathsf{H}(K_b)$ is forward (thus being injective), $z_b \in \mathsf{Z}(K_b) \setminus \mathsf{Z}(K_{b-1})$; if $\psi_{b-1}^* : \mathsf{H}(K_{b-1}) \leftarrow \mathsf{H}(K_b)$ is backward (thus being surjective), then $[z_b]$ is the non-zero element in $\ker(\psi_{b-1}^*)$.*

Death condition: *If* $d < i$ *and* $\psi_d^* : \mathsf{H}(K_d) \leftarrow \mathsf{H}(K_{d+1})$ *is backward (thus being injective),* $z_d \in \mathsf{Z}(K_d) \setminus \mathsf{Z}(K_{d+1})$*; if* $d < i$ *and* $\psi_d^* : \mathsf{H}(K_d) \to \mathsf{H}(K_{d+1})$ *is forward (thus being surjective), then* $[z_d]$ *is the non-zero element in* $\ker(\psi_d^*)$.

REMARK 2.2. *By definition, all* z_α*'s in a homology representative* rep *are* p*-cycles for the same* p*, so we can also call* rep *a* p*-th homology representative.*

DEFINITION 2.4. (BOUNDARY REPRESENTATIVES) *Let* $[b,d] \subseteq [0,i]$ *be an interval where* $b \in \mathsf{P}^B(\mathcal{F}_i)$ *and* $d \in \mathsf{N}^B(\mathcal{F}_i)$*. A sequence of cycles* $\mathsf{rep} = \{z_\alpha \in \mathsf{B}(K_\alpha) \mid \alpha \in [b,d]\}$ *is called a* boundary representative *(or simply* representative*) for the interval* $[b,d]$ *if for every* $b \le \alpha < d$*, either* $z_{\alpha+1} = \psi_\alpha^\#(z_\alpha) \stackrel{def}{=} z_\alpha$ *or* $z_\alpha = \psi_\alpha^\#(z_{\alpha+1}) \stackrel{def}{=} z_{\alpha+1}$ *based on the direction of* $\psi_\alpha^\#$*. Furthermore, we have:*

Birth condition: *The cycle* z_b *satisfies that* $z_b \in \mathsf{B}(K_b) \setminus \mathsf{B}(K_{b-1})$ *where* $\psi_{b-1}^\# : \mathsf{B}(K_{b-1}) \to \mathsf{B}(K_b)$ *is forward because* $b \in \mathsf{P}^B(\mathcal{F}_i)$.

Death condition: *If* $d < i$*, then* z_d *satisfies that* $z_d \in \mathsf{B}(K_d) \setminus \mathsf{B}(K_{d+1})$ *where the map* $\psi_d^\# : \mathsf{B}(K_d) \leftarrow \mathsf{B}(K_{d+1})$ *is backward because* $d \in \mathsf{N}^B(\mathcal{F}_i)$.

REMARK 2.3. *In the sequence* rep *in Definitions 2.3 and 2.4, we also call* z_α *a* cycle at index α.

The following Proposition (proof in Appendix A) is used later for proofs and algorithms.

PROPOSITION 2.1. *Let* $z_1^B, \dots, z_k^B$ *be the cycles at index* j *in representatives for all intervals of* $\mathsf{Pers}^B(\mathcal{F}_i)$ *containing* j*. Similarly, let* $z_1^H, \dots, z_{k'}^H$ *be the cycles at index* j *in representatives for all intervals of* $\mathsf{Pers}^H(\mathcal{F}_i)$ *containing* j*. Then,* $\{[z_1^H], \dots, [z_{k'}^H]\}$ *is a basis of* $\mathsf{H}(K_j)$*,* $\{z_1^B, \dots, z_k^B\}$ *is a basis of* $\mathsf{B}(K_j)$*, and* $\{z_1^H, \dots, z_{k'}^H, z_1^B, \dots, z_k^B\}$ *is a basis of* $\mathsf{Z}(K_j)$.

We then define summations of representatives for intervals ending at i. These summations respect a total order '$\prec$' on birth indices [10], that is, a representative for $[b,i]$ can be added to a representative for $[b',i]$ if and only if $b \prec b'$ (see Figure 2).

DEFINITION 2.5. (TOTAL ORDER ON BIRTH INDICES) *For two birth indices* $b, b' \in \mathsf{P}^H(\mathcal{F}_i) \cup \mathsf{P}^B(\mathcal{F}_i)$*, we have* $b \prec b'$ *if one of the following holds:*

(i) $b \in \mathsf{P}^B(\mathcal{F}_i)$ *and* $b' \in \mathsf{P}^H(\mathcal{F}_i)$*;*

(ii) $b, b' \in \mathsf{P}^B(\mathcal{F}_i)$ *and* $b < b'$*;*

(iii) $b, b' \in \mathsf{P}^H(\mathcal{F}_i)$*,* $b < b'$*, and* $K_{b'-1} \hookrightarrow K_{b'}$ *is a forward inclusion;*

(iv) $b, b' \in \mathsf{P}^H(\mathcal{F}_i)$*,* $b' < b$*, and* $K_{b-1} \hookleftarrow K_b$ *is a backward inclusion.*

DEFINITION 2.6. (REPRESENTATIVE SUMMATION) *For two intervals* $[b,i], [b',i] \in \mathsf{Pers}_p^H(\mathcal{F}_i) \cup \mathsf{Pers}_p^B(\mathcal{F}_i)$ *so that* $b \prec b'$*, let* $\mathsf{rep} = \{z_\alpha \mid \alpha \in [b,i]\}$ *and* $\mathsf{rep}' = \{z'_\alpha \mid \alpha \in [b',i]\}$ *be* p*-th representatives for* $[b,i]$ *and* $[b',i]$ *respectively. The* sum *of* rep *and* rep′*, denoted* $\mathsf{rep} \boxplus \mathsf{rep}'$*, is a sequence of cycles* $\{\overline{z}_\alpha \mid \alpha \in [b',i]\}$ *so that*

- *If* $b < b'$ *then* $\overline{z}_\alpha = z_\alpha + z'_\alpha$ *for each* α*; (Figure 2: (i) top, (ii), (iii))*
- *If* $b' < b$*, then* $\overline{z}_\alpha = z'_\alpha$ *for* $\alpha < b$ *and* $\overline{z}_\alpha = z_\alpha + z'_\alpha$ *for* $\alpha \ge b$*. (Figure 2: (i) bottom, (iv))*

PROPOSITION 2.2. *The sequence* $\mathsf{rep} \boxplus \mathsf{rep}'$ *in Definition 2.6 is a* p*-th representative for* $[b',i] \in \mathsf{Pers}_p^H(\mathcal{F}_i) \cup \mathsf{Pers}_p^B(\mathcal{F}_i)$.

Proof. See Appendix B. ∎

REMARK 2.4. *From Figure 2, it is not hard to see that the representative resulting from the summation in Definition 2.6 is still a valid representative for the interval. For example, in case (iii) of Figure 2, the resulting representative is valid because* $z_{b'} + z'_{b'}$ *still contains* $\sigma_{b'-1}$ *so that the birth condition in Definition 2.3 still holds.*

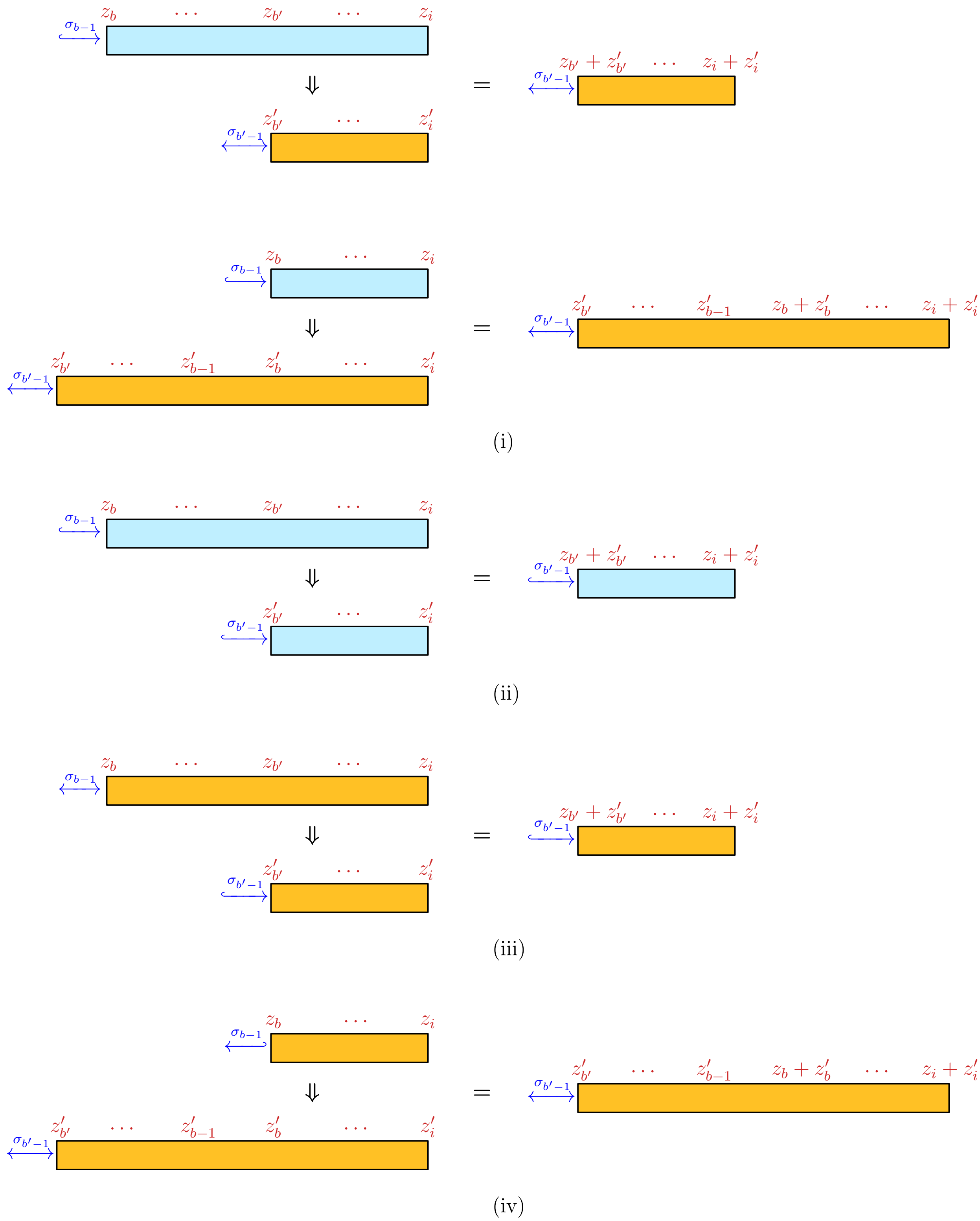

Figure 2: Illustration of how summations of representatives for intervals respect the order '≺' for the different cases in Definition 2.5, with the double arrows indicating the directions of the summations. Boundary module intervals are shaded blue while homology module intervals are shaded orange.

We then define wires and bundles as mentioned in Section 1 which compresses the zigzag representatives in a compact form.

DEFINITION 2.7. (WIRE) *A* wire *is a cycle* $\omega_i \in \mathsf{Z}(K_i)$ *with a* starting index $i \in \mathsf{P}^H(\mathscr{F}) \cup \mathsf{P}^B(\mathscr{F})$ *s.t.*

(i) $K_{i-1} \hookrightarrow K_i$ *is forward and* $\omega_i \in \mathsf{Z}(K_i) \setminus \mathsf{Z}(K_{i-1})$, *or*

(ii) $K_{i-1} \hookleftarrow K_i$ *is backward and* $\omega_i \in \mathsf{B}(K_{i-1}) \setminus \mathsf{B}(K_i)$, *or*

(iii) $K_{i-1} \hookrightarrow K_i$ *is forward and* $\omega_i \in \mathsf{B}(K_i) \setminus \mathsf{B}(K_{i-1})$.

We also say that ω_i *is a wire* at *index* i. *The wires satisfying (i) or (ii) are also called* non-boundary wires *whereas those satisfying (iii) are called* boundary wires.

REMARK 2.5. *In cases (i) and (ii) above,* $i \in \mathsf{P}^H(\mathscr{F})$, *whereas in case (iii),* $i \in \mathsf{P}^B(\mathscr{F})$.

DEFINITION 2.8. (WIRE BUNDLE) *A* wire bundle W *(or simply* bundle*) is a set of wires with distinct starting indices. The* sum *of* W *with another wire bundle* W', *denoted* $W \boxplus W'$, *is the symmetric difference of the two sets. We also call* W *a* boundary bundle *if* W *contains only boundary wires and call* W *a* non-boundary bundle *otherwise.*

As evident later, given an input filtration $\mathscr{F}$, a wire at an index i is fixed in our algorithm, and we always denote such a wire as ω_i. Hence, a wire bundle is simply stored as a list of wire indices in our algorithm. Since there are $O(m)$ indices in $\mathscr{F}$, a bundle summation takes $O(m)$ time.

DEFINITION 2.9. *Let* $[b,d] \in \mathsf{Pers}^H(\mathscr{F}_i) \cup \mathsf{Pers}^B(\mathscr{F}_i)$. *A wire bundle* W *is said to* generate a representative *for* $[b,d]$ *(or simply* represents $[b,d]$*) if the sequence of cycles* $\{z_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j \mid \alpha \in [b,d]\}$ *is a representative for* $[b,d]$.

REMARK 2.6. *In the sum* $z_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j$ *in the above definition, we consider each* ω_j *and the sum* z_α *as a cycle in the total complex* $\overline{K}$. *Notice that if* W *generates a representative for* $[b,d]$, *we may have that* $\omega_j \notin \mathsf{Z}(K_\alpha)$ *for a* ω_j *in the sum, but we still can have* $z_\alpha \in \mathsf{Z}(K_\alpha)$ *due to cancellation of simplices in the symmetric difference. See Figure 1.*

Figure 1 and 4 provide examples for representatives generated by bundles. Notice that since we always consider bundles that generate representatives in this paper, bundle summations also respect the order '$\prec$' in Definition 2.5. The main benefits of introducing wire bundles are that (i) they can be summed efficiently and (ii) explicit representatives can be generated from them also efficiently (see the Algorithm EXTREP below for the detailed process).

ALGORITHM 2.1. (EXTREP: EXTRACTING REPRESENTATIVE FROM BUNDLE) *Let* $W = \{\omega_{\iota_1}, \ldots, \omega_{\iota_\ell}\}$ *be a wire bundle where* $\iota_1 < \cdots < \iota_\ell$ *and let* rep *be the representative for an interval* $[b,d]$ *generated by* W. *We can assume* $\iota_\ell \leq d$ *because wires in* W *with indices greater than* d *do not contribute to a cycle in* rep. *Moreover, let* ι_k *be the last index in* $\iota_1, \ldots, \iota_\ell$ *no greater than* b. *We have that* $z = \sum_{j=\iota_1}^{\iota_k} \omega_j$ *is the cycle at indices* $[b, \iota_{k+1})$ *in* rep. *We then let* λ *iterate over* $k+1, \ldots, \ell-1$. *For each* λ, *we add* ω_{ι_λ} *to* z, *and the resulting* z *is the cycle at indices* $[\iota_\lambda, \iota_{\lambda+1})$ *in* rep. *Finally, we add* ω_{ι_ℓ} *to* z, *and* z *is the cycle at indices* $[\iota_\ell, d]$ *in* rep. *Since at every* $\lambda \in [k+1, \ell]$, *we add at most one cycle to another cycle, the whole process involves* $O(m)$ *chain summations.*

3 Representatives as wire bundles

We first give a brief overview of our algorithm to illustrate how representatives in zigzag modules can be compactly stored as wire bundles (see Section 4 for details of the algorithm). Consider computing only the homology barcode $\mathsf{Pers}^H(\mathscr{F})$. Our algorithm in Section 4 stems from an idea for computing $\mathsf{Pers}^H(\mathscr{F})$ that directly maintains representatives for the intervals: Before each iteration i, assume that we are given intervals in $\mathsf{Pers}^H(\mathscr{F}_i)$ and their representatives. The aim of iteration i is to compute those for $\mathsf{Pers}^H(\mathscr{F}_{i+1})$ by processing the inclusion $K_i \xleftrightarrow{\sigma_i} K_{i+1}$. For the computation, we only need to pay attention to those *active* intervals in $\mathsf{Pers}^H(\mathscr{F}_i)$ ending with i because the non-active intervals and their representatives have already been finalized. Consider an interval

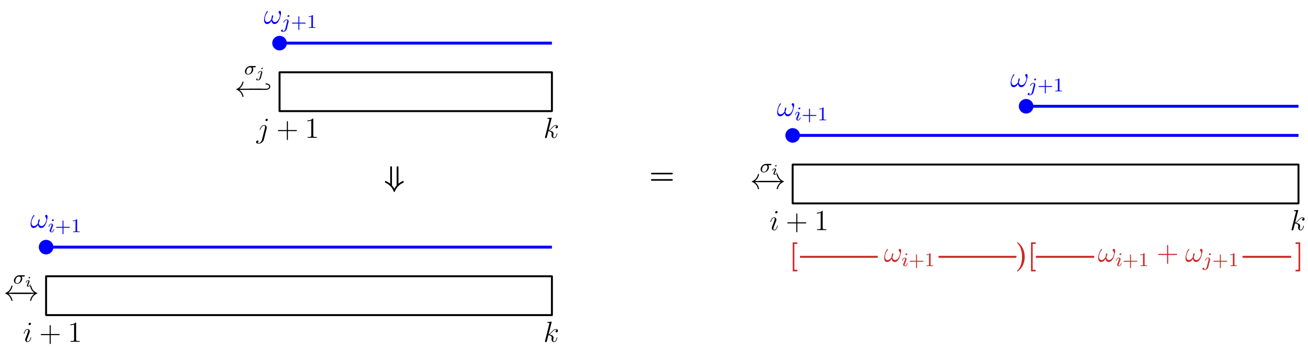

Figure 3: Summing the two representatives generated by a single wire results in a new representative generated by a bundle containing the two wires.

$[b,i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ with a representative rep. If the last cycle z_i (at index i) in rep resides in K_{i+1}, the interval $[b,i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ can be directly extended to $[b,i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ along with the representative where the cycle at $i+1$ equals z_i. Otherwise, if $z_i \nsubseteq K_{i+1}$, we perform summations on the representatives to modify z_i in rep so that z_i becomes contained in K_{i+1} and $[b,i]$ can be extended.

In iteration i, whenever the inclusion $K_i \leftrightarrow K_{i+1}$ generates a new birth index $i+1 \in \mathsf{P}^H(\mathcal{F}_{i+1})$, we have a new active interval $[i+1,i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. We assign a representative $\mathsf{rep}^{i+1} = \{z_{i+1}\}$ to $[i+1,i+1]$ where z_{i+1} only needs to satisfy the birth condition in Definition 2.3. Suppose that $[i+1,i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ is directly extended to $[i+1,k] \in \mathsf{Pers}^H(\mathcal{F}_k)$ in later iterations without its representative $\mathsf{rep}^{i+1} = \{z_\alpha \mid \alpha \in [i+1,k]\}$ being modified by representative summations. We then have that $z_\alpha = z_{i+1}$ for each α, which means that rep^{i+1} is generated by the wire $\omega_{i+1} := z_{i+1}$ (see Figure 3.). Suppose that we have a similar interval $[j+1,k] \in \mathsf{Pers}^H(\mathcal{F}_k)$ with a representative rep^{j+1} also generated by a single wire ω_{j+1}, where $j+1 > i+1$ and $K_j \hookleftarrow K_{j+1}$ is backward. Then, $j+1 \prec i+1$ according to Definition 2.5, and we can sum rep^{j+1} to rep^{i+1} to get a new representative for $[i+1,k]$. We have that the new representative is generated by the bundle $\{\omega_{i+1}, \omega_{j+1}\}$ as illustrated in Figure 3.

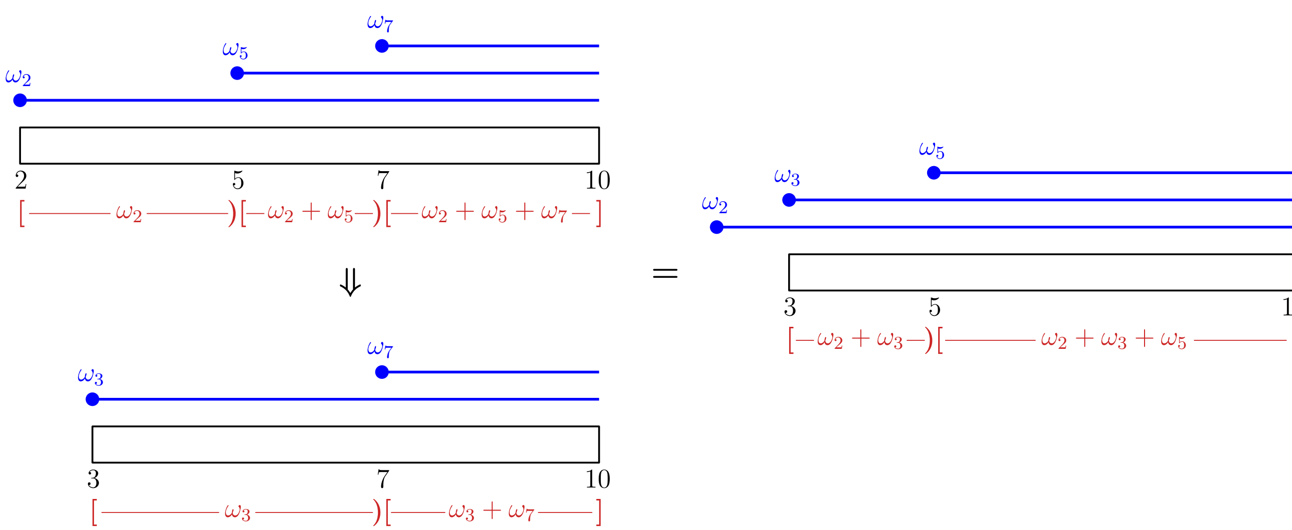

Figure 4: Summing two representatives generated by the bundles $\{\omega_2, \omega_5, \omega_7\}, \{\omega_3, \omega_7\}$ respectively results in a new representative generated by the bundle $\{\omega_2, \omega_3, \omega_5\}$.

In the computation of $\mathsf{Pers}^H(\mathcal{F})$, a representative can only be changed due to a direct extension or a representative summation after being created. It is easy to verify that a representative is generated by a bundle after being created and that a representative is still generated by a bundle after being extended given that it is generated by a bundle before the extension. We then only need to show that $\mathsf{rep} \boxplus \mathsf{rep}'$ is still generated by a bundle if two representatives rep and rep' are generated by bundles. Figure 4 provides an example involving two intervals $[2, 10], [3, 10]$ whose representatives are generated by the bundles $\{\omega_2, \omega_5, \omega_7\}, \{\omega_3, \omega_7\}$ respectively. The resulting representative of the summation is generated by the bundle $\{\omega_2, \omega_3, \omega_5\}$ which is the symmetric difference. In general, for two bundles W and W' generating representatives rep and rep' respectively, it could happen that the representative rep^* generated by $W \boxplus W'$ is not equal to $\mathsf{rep} \boxplus \mathsf{rep}'$. However, we have that each cycle in rep^* is always homologous to the corresponding cycle in $\mathsf{rep} \boxplus \mathsf{rep}'$. The rest of the section formally justifies the claim.

The reader may wonder why we need the boundary module and its representatives at all. While theoretically $\mathsf{Pers}^H(\mathscr{F})$ and the bundles generating the representatives can be computed independently without considering the boundary module $\mathsf{B}(\mathscr{F})$, introducing $\mathsf{B}(\mathscr{F})$ helps us achieve the $O(m^2 n)$ time complexity. See Remark 4.4 in Section 4 for a detailed explanation.

For any interval in $\mathsf{Pers}^H(\mathscr{F}_i) \cup \mathsf{Pers}^B(\mathscr{F}_i)$, our algorithm maintains a wire bundle generating its representative. Proposition 3.1 lets us replace representatives with wire bundles.

PROPOSITION 3.1. *Let $[b, i], [b', i] \in \mathsf{Pers}^H(\mathscr{F}_i) \cup \mathsf{Pers}^B(\mathscr{F}_i)$ and $b \prec b'$. Suppose that W and W' generate a representative for $[b, i]$ and $[b', i]$ respectively. Then, the sum $W \boxplus W'$ generates a representative for $[b', i] \in \mathsf{Pers}^H(\mathscr{F}_i) \cup \mathsf{Pers}^B(\mathscr{F}_i)$.*

Before proving Proposition 3.1, we prove a result (Proposition 3.2) which says that wires in a bundle for an interval, which gets added to other intervals, produce only boundaries outside the interval and those boundaries reside in the respective complexes. This, in turn, helps to prove Proposition 3.1.

Each time we extend an interval $[b, i-1]$ in $\mathsf{Pers}^H(\mathscr{F}_{i-1})$ (resp. $\mathsf{Pers}^B(\mathscr{F}_{i-1})$) to $[b, i]$ in $\mathsf{Pers}^H(\mathscr{F}_i)$ (resp. $\mathsf{Pers}^B(\mathscr{F}_i)$), the birth index b does not change. So we denote the bundle associated with $[b, i]$ as W^b in this section. After being created, W^b only changes when another bundle W^x is added to it because the representative generated by W^x needs to be added to the representative for $[b, i]$ generated by W^b.

DEFINITION 3.1. *A boundary bundle W is said to be* alive till index b *if the cycle $z_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j$ is in $\mathsf{B}(K_\alpha)$ for every $\alpha \leq b$. Notice that z_α is the empty chain if there is no $\omega_j \in W$ s.t. $j \leq \alpha$.*

PROPOSITION 3.2. *Let $[b, i] \in \mathsf{Pers}^H(\mathscr{F}_i)$ with $K_{b-1} \hookleftarrow K_b$ being backward, or $[b, i] \in \mathsf{Pers}^B(\mathscr{F}_i)$. Let $\overline{W}^b \subseteq W^b$ be defined as $\overline{W}^b = \{\omega_j \in W^b \mid j < b\}$. Then, $\overline{W}^b$ is a boundary bundle alive till b.*

Proof. Let X be the set containing each index $x \leq i$ so that either $x \in \mathsf{P}^H(\mathscr{F}_i)$ with backward $K_{x-1} \hookleftarrow K_x$, or simply $x \in \mathsf{P}^B(\mathscr{F}_i)$. Let $\overline{W}^x = \{\omega_j \in W^x \mid j < x\}$ for each $x \in X$.

Let $a_0, a_1, \ldots, a_k$ denote the series of all operations that change a bundle W^x for $x \in X$, i.e., each a_j either creates a bundle W^x at an index $x \in X$ or sums a bundle W^y to W^x for $x \in X$. Notice that y is necessarily in X because the bundle summation respects the order '$\prec$' in Definition 2.5. We show by induction on the number of operations k that the bundle W^x for any $x \in X$ maintains the property that the derived $\overline{W}^x$ is a boundary bundle alive till index x. The operation a_0 starts a representative with a single cycle $z \in \mathsf{Z}(K_x)$ at some index $x \in X$ with the wire $\omega_x = z$. The bundle W^x then equals $\{\omega_x\}$ and the claim is trivially true.

For the inductive step, assume that the claim is true after an operation a_ℓ for $\ell \geq 0$. If the the operation $a_{\ell+1}$ starts a representative, the claim holds trivially. Assume that $a_{\ell+1}$ adds a wire bundle W^y, $y \in X$, to a W^x. By the inductive hypothesis, $\overline{W}^x = \{\omega_j \mid \omega_j \in W^x, j < x\}$ and $\overline{W}^y = \{\omega_j \mid \omega_j \in W^y, j < y\}$ are boundary bundles alive till x and y respectively. There are two possibilities:

(i) $y > x$: Let $W = W^x \boxplus W^y$. Observe that, for $\alpha \leq x$, the cycle $z_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j = \sum_{\omega_j \in \overline{W}^x, j \leq \alpha} \omega_j + \sum_{\omega_j \in \overline{W}^y, j \leq \alpha} \omega_j$ is a boundary in K_α because the two cycles given by the two sums on RHS are boundaries in K_α. It follows that $\overline{W} = \{\omega_j \in W \mid j < x\}$ is a boundary bundle alive till x and the inductive hypothesis still holds for x.

(ii) $y < x$: Let $W = W^x \boxplus W^y$. In this case, $y \in \mathsf{Pers}^B(\mathscr{F})$. It can be verified that, since $y \in \mathsf{Pers}^B(\mathscr{F})$, W^y is necessarily a boundary bundle because the bundle summations respect the order in Definition 2.5. Then, the bundle $W' = \{\omega_j \mid \omega_j \in W^y, j < x\}$ is a boundary bundle alive till x. By the inductive hypothesis, the wire bundle $\overline{W}^x$ is a boundary bundle alive till x. Therefore, the sum $W' \boxplus \overline{W}^x$, which is the updated $\overline{W}^x$, is a boundary bundle alive till x; the inductive hypothesis follows. ∎

Proof. [Proof of Proposition 3.1] Let $\mathsf{rep} = \{z_\alpha \mid \alpha \in [b, i]\}$, $\mathsf{rep}' = \{z'_\alpha \mid \alpha \in [b', i]\}$ be the representatives generated by W and W' respectively. We have the following cases to consider:

Case 1, $b < b'$: In this case, every cycle $\overline{z}_\alpha$, $\alpha \in [b', i]$, in $\mathsf{rep} \boxplus \mathsf{rep}'$ satisfies that $\overline{z}_\alpha = z_\alpha + z'_\alpha$. Since $z_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j$ and $z'_\alpha = \sum_{\omega_j \in W', j \leq \alpha} \omega_j$, we have that

$$\overline{z}_\alpha = \sum_{\omega_j \in W, j \leq \alpha} \omega_j + \sum_{\omega_j \in W', j \leq \alpha} \omega_j = \sum_{\omega_j \in W \boxplus W', j \leq \alpha} \omega_j.$$

This means that $W \boxplus W'$ generates $\mathsf{rep} \boxplus \mathsf{rep}'$, a representative for $[b', i]$ by Proposition 2.2.

Case 2, $b' < b$: We have $\overline{z}_\alpha$ in $\mathsf{rep} \boxplus \mathsf{rep}'$ equals z'_α for $b' \leq \alpha < b$. However, the wire bundle W may have wires in $\overline{W} = \{\omega_j \in W \mid j < b\}$ whose addition to z'_α, $b' \leq \alpha < b$, may create a different cycle in the representative generated by $W \boxplus W'$. By Proposition 3.2, $\overline{W}$ is necessarily a boundary bundle alive till index b. Let $\overline{z}'_\alpha$ be the cycle at index α in the representative generated by $W \boxplus W'$, where $b' \leq \alpha < b$. Then, $\overline{z}'_\alpha = \sum_{\omega_j \in W \boxplus W', j \leq \alpha} \omega_j = z'_\alpha + \sum_{\omega_j \in \overline{W}, j \leq \alpha} \omega_j$, which means that $\overline{z}'_\alpha$ is homologous to z'_α. Hence, $\overline{z}'_\alpha$ can be taken as a cycle in a representative for $[b', i]$. This means that $W \boxplus W'$ generates a representative for $[b', i]$. ∎

THEOREM 3.1. *There is a wire bundle* $W = \{w_\iota \mid \iota \in \mathsf{P}^H(\mathcal{F}) \cup \mathsf{P}^B(\mathcal{F})\}$ *so that a representative for any* $[b, d] \in \mathsf{Pers}^H(\mathcal{F}) \cup \mathsf{Pers}^B(\mathcal{F})$ *is generated by a wire bundle that is a subset of* W.

Proof. We give a constructive proof. Assume inductively that we have constructed a wire bundle $W_i = \{w_\iota \mid \iota \in \mathsf{P}^H(\mathcal{F}_i) \cup \mathsf{P}^B(\mathcal{F}_i)\}$ so that for every $[b, d] \in \mathsf{Pers}^H(\mathcal{F}_i) \cup \mathsf{Pers}^B(\mathcal{F}_i)$, we have a wire bundle $W^{[b,d]} \subseteq W_i$ that generates a representative for $[b, d]$. The base case when $i = 0$ holds trivially. For the inductive step, consider extending the filtration $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$ while assuming the hypothesis for $\mathcal{F}_i$. Since any $[b, d] \in \mathsf{Pers}^H(\mathcal{F}_i) \cup \mathsf{Pers}^B(\mathcal{F}_i)$ where $d < i$ is not affected by the extension, we do not consider them in the arguments below.

Case 1, $K_i \xhookrightarrow{\sigma_i} K_{i+1}$ and $i + 1 \in \mathsf{P}^H(\mathcal{F}_{i+1})$: Any $[b, i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ extends to $[b, i + 1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ because the representative cycle z_i at index i for $[b, i]$ is also in K_{i+1} and thus we choose $z_{i+1} = z_i$ for $[b, i + 1]$. Then, the wire bundle $W^{[b,i]}$ also represents $[b, i + 1]$. The same holds for intervals in $\mathsf{Pers}^B(\mathcal{F}_i)$. We also have a new interval $[i + 1, i + 1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. Let a new wire ω_{i+1} be any cycle in $\mathsf{Z}(K_{i+1})$ containing σ_i. We have that the bundle $\{\omega_{i+1}\}$ generates a representative for $[i + 1, i + 1]$. Subsets of the wire bundle $W_{i+1} = W_i \cup \{\omega_{i+1}\}$ then represent intervals in both $\mathsf{Pers}^H(\mathcal{F}_{i+1})$ and $\mathsf{Pers}^B(\mathcal{F}_{i+1})$.

Case 2, $K_i \xhookrightarrow{\sigma_i} K_{i+1}$ and $i \in \mathsf{N}^H(\mathcal{F}_i)$: In this case, $\partial\sigma_i$ becomes a boundary in K_{i+1}, an interval in $\mathsf{Pers}^H(\mathcal{F}_i)$ does not extend to $i+1$, and a new interval $[i+1, i+1]$ in $\mathsf{Pers}^B(\mathcal{F}_{i+1})$ begins. To determine the interval $[b, i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ that does not extend to $i + 1$, consider the cycle $\partial\sigma_i$ which is in $\mathsf{Z}(K_i) \setminus \mathsf{B}(K_i)$. Let $[b_1, i], \ldots, [b_k, i]$ be all the intervals in $\mathsf{Pers}^H(\mathcal{F}_i)$ and $\mathsf{Pers}^B(\mathcal{F}_i)$ with representatives $\mathsf{rep}_1, \ldots, \mathsf{rep}_k$ respectively. Let $z_1, \ldots, z_k$ be their cycles at index i respectively. Since these cycles form a basis for $\mathsf{Z}(K_i)$ by Proposition 2.1, the cycle $\partial\sigma_i$ is a linear combination of them. Without loss of generality (WLOG), assume that after reindexing, $\partial\sigma_i = z_1 + \cdots + z_\ell$ for some $\ell \leq k$ where $b_1 \prec \cdots \prec b_\ell$. Add the representatives $\mathsf{rep}_1, \ldots, \mathsf{rep}_{\ell-1}$ to rep_ℓ to obtain a new representative rep'_ℓ for $[b_\ell, i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ (Proposition 2.2). The cycle of rep'_ℓ at index i is $\partial\sigma_i$ by construction which becomes a boundary in K_{i+1}. Therefore, rep'_ℓ is a representative for $[b_\ell, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ and $W^{[b_1,i]} \boxplus \cdots \boxplus W^{[b_\ell,i]}$ represents $[b_\ell, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ by Proposition 3.1. All other intervals in $\mathsf{Pers}^H(K_i)$ and $\mathsf{Pers}^B(K_i)$ extend to $\mathsf{Pers}^H(K_{i+1})$ and $\mathsf{Pers}^B(K_{i+1})$ with their wire bundles remaining the same. A new interval $[i + 1, i + 1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ begins whose representative is given by the cycle $\partial\sigma_i$. So, the wire $\omega_{i+1} = \partial\sigma_i$ represents this interval in $\mathsf{Pers}^B(\mathcal{F}_{i+1})$. Subsets of the wire bundle $W_{i+1} = W_i \cup \{\omega_{i+1}\}$ then generate representatives for all intervals in $\mathsf{Pers}^H(\mathcal{F}_{i+1}) \cup \mathsf{Pers}^B(\mathcal{F}_{i+1})$.

We have two remaining cases whose details are provided in Appendix C. To finish the proof, we also need to show that the zigzag barcodes we have are correct whenever we proceed from $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$. Since all intervals we have admit representatives, the correctness of the barcodes follows from Proposition 4.1 presented in Section 4. ∎

4 Algorithm

We present the $O(m^2 n)$ algorithm WIREDZIGZAG computing $\mathsf{Pers}^H(\mathcal{F})$, $\mathsf{Pers}^B(\mathcal{F})$, and their representatives based on exposition in the previous section. As mentioned, the general idea of the algorithm is to maintain a wire bundle for each interval in $\mathsf{Pers}^H(\mathcal{F})$ and $\mathsf{Pers}^B(\mathcal{F})$ so that the bundle generates a representative for the interval. In each iteration i, the algorithm processes the inclusion $K_i \xleftrightarrow{\sigma_i} K_{i+1}$ in $\mathcal{F}$ starting with $i = 0$. Before iteration i, we assume that we have computed all intervals in $\mathsf{Pers}^H(\mathcal{F}_i)$ and $\mathsf{Pers}^B(\mathcal{F}_i)$ along with the wire bundles. The aim of iteration i is to compute those for $\mathsf{Pers}^H(\mathcal{F}_{i+1})$ and $\mathsf{Pers}^B(\mathcal{F}_{i+1})$. In each iteration i, we have two sets of *active* intervals (ending with i) for $\mathsf{Pers}^H(\mathcal{F}_i)$ and $\mathsf{Pers}^B(\mathcal{F}_i)$ respectively,

$$\{[\hat{b}_j, i] \in \mathsf{Pers}^H(\mathcal{F}_i) \mid j = 1, 2, \ldots, r\}, \quad \{[b'_k, i] \in \mathsf{Pers}^B(\mathcal{F}_i) \mid k = 1, 2, \ldots, q\}$$

where r is the dimension of $\mathsf{H}(K_i)$ and q is the dimension of $\mathsf{B}(K_i)$. All non-active intervals in $\mathsf{Pers}^H(\mathcal{F}_i)$ (resp. $\mathsf{Pers}^B(\mathcal{F}_i)$) are automatically carried into $\mathsf{Pers}^H(\mathcal{F}_{i+1})$ (resp. $\mathsf{Pers}^B(\mathcal{F}_i)$) and their wire bundles do not change. For each homology interval $[\hat{b}_j, i]$, we let W^j denote the (non-boundary) bundle maintained for $[\hat{b}_j, i]$, and for each boundary interval $[b'_k, i]$, we let U^k denote the (boundary) bundle maintained for $[b'_k, i]$. At the end of the algorithm, we have all intervals and bundles in $\mathsf{Pers}^H(\mathcal{F}_m) = \mathsf{Pers}^H(\mathcal{F})$ and $\mathsf{Pers}^B(\mathcal{F}_m) = \mathsf{Pers}^B(\mathcal{F})$. We then generate a representative for each interval from its bundle.

4.1 Maintenance of pivoted matrices For the computation, we maintain three 0-1 matrices Z, B, and C where each column represents a chain s.t. the k-th entry of the column equals 1 iff the simplex with index k belongs to the chain. We also do not differentiate a matrix column and the chain it represents when describing the algorithm. In each iteration i, the following invariants hold:

1. Z has r columns each corresponding to an active interval in $\mathsf{Pers}^H(\mathcal{F}_i)$ s.t. a column $Z[j]$ equals the last cycle (at index i) in the representative for $[\hat{b}_j, i]$ generated by W^j.
2. B has q columns each corresponding to an active interval in $\mathsf{Pers}^B(\mathcal{F}_i)$ s.t. a column $B[k]$ equals the last cycle in the representative for $[b'_k, i]$ generated by U^k.
3. C also has q columns s.t. $B[k] = \partial(C[k])$ for each k.

By Proposition 2.1, columns in Z and B form a basis of $\mathsf{Z}(K_i)$. Throughout the algorithm, we also always ensure that columns in Z, B, and C form a basis for $\mathsf{C}(K_i)$ in each iteration. This can be inductively proved based on the details of the algorithm presented in this section and Appendix D. The detailed justification is omitted. Let the *pivot* of a matrix column be the index of its lowest entry equal to 1. Our algorithm maintains the invariant that *columns in Z and B altogether have distinct pivots* so that getting the coordinates of a cycle in $\mathsf{Z}(K_i)$ in terms of the basis represented by columns of Z and B takes $O(n^2)$ time. This is an essential part of our algorithm as demonstrated in its description below.

REMARK 4.1. *Since a bundle is just a set of wire indices and we have no more than one wire born at each index, we maintain all wires in a matrix representing the containments $\omega_i \mapsto \{\sigma \in K_i \mid \sigma \in \omega_i\}$. Similarly, bundles are maintained in a matrix representing the containments $W \mapsto \{\omega_i \in W\}$.*

4.2 Detailed processing in iteration i of WIREDZIGZAG Iteration i of the algorithm has the following processes in different cases (more details such as how we ensure the distinct pivots in Z, B and how to determine the injective/surjective cases are provided in Appendix D):

$K_i \xhookrightarrow{\sigma_i} K_{i+1}$ **is forward, ψ_i^* is injective:** We have:

Birth in homology module ($i+1 \in \mathsf{P}^H(\mathcal{F})$): An interval $[i+1, i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ active in the next iteration is created. We find a new non-boundary wire ω_{i+1} which is a cycle in K_{i+1} containing σ_i so that condition (i) in Definition 2.7 is satisfied. We also have a new non-boundary bundle $\{\omega_{i+1}\}$ for $[i+1, i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. (The validity of the new bundle $\{\omega_{i+1}\}$ can be seen by examining Definitions 2.3 and 2.7.) Each $[\hat{b}_j, i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ extends to be an active interval $[\hat{b}_j, i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. Since $Z[j] \subseteq K_{i+1}$ because $K_i \xhookrightarrow{\sigma_i} K_{i+1}$ is forward, W^j stays the same for the next iteration. Finally, since ω_{i+1} is the cycle at index $i+1$ in the representative generated by the bundle $\{\omega_{i+1}\}$, we add ω_{i+1} as a new column to Z corresponding to the new active interval.

Since $\mathsf{B}(K_i) = \mathsf{B}(K_{i+1})$, each $[b'_k, i] \in \mathsf{Pers}^B(\mathcal{F}_i)$ extends to be an active interval $[b'_k, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ and the bundle U^k stays the same.

$K_i \xhookrightarrow{\sigma_i} K_{i+1}$ **is forward, ψ_i^* is surjective:** Both of the following happen:

Death in homology module ($i \in \mathsf{N}^H(\mathcal{F})$): By performing reductions on $\partial\sigma_i$ and the columns in Z and B, we find a subset of columns ($J \subseteq \{1, \ldots, r\}$) in Z s.t.

$$\sum_{j \in J} [Z[j]] = [\partial\sigma_i] \tag{4.3}$$

in $\mathsf{H}(K_i)$. Let $\hat{b}_\lambda$ be the maximum birth index in $\{\hat{b}_j \mid j \in J\}$ w.r.t the order '$\prec$'. We have that $[\hat{b}_\lambda, i]$ ceases to be active, i.e., $[\hat{b}_\lambda, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. Let W^* be the sum of all the bundles in $\{W^j \mid j \in J\}$. We have that W^* generates a representative rep^* for $[\hat{b}_\lambda, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$. (To see this, notice that the last cycle in rep^* is $\sum_{j \in J} Z[j]$ which is homologous to $\partial\sigma_i$ in K_i, and so the death condition in Definition 2.3 is satisfied. The validity of W^* then follows from Proposition 3.1.) For each $j \in \{1, \ldots, r\} \setminus \{\lambda\}$, $[\hat{b}_j, i] \in \mathsf{Pers}^H(\mathcal{F}_i)$ extends to be $[\hat{b}_j, i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ for which W^j stays the same because $K_i \xhookrightarrow{\sigma_i} K_{i+1}$ is forward. Finally, we delete $Z[\lambda]$ from Z.

Birth in boundary module ($i+1 \in \mathsf{P}^B(\mathcal{F})$): A new active interval $[i+1, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ is created. We have a new boundary wire $\omega_{i+1} = \partial\sigma_i$ satisfying condition (iii) in Definition 2.7. We also have a new boundary bundle $\{\omega_{i+1}\}$ for $[i+1, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$. Each $[b'_k, i] \in \mathsf{Pers}^B(\mathcal{F}_i)$ extends to be $[b'_k, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ for which U^k stays the same. Finally, we add $\partial\sigma_i$ as a new column to B and add a new column containing only σ_i to C.

$K_i \xleftarrow{\sigma_i} K_{i+1}$ **is backward, ψ_i^* is surjective:** Both of the following happen:

Birth in homology module ($i+1 \in \mathsf{P}^H(\mathcal{F})$): A new active interval $[i+1, i+1] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ is created. We find a new non-boundary wire ω_{i+1} which is a cycle homologous to $\partial\sigma_i$ in K_{i+1} so that condition (ii) in Definition 2.7 is satisfied. The rest of the processing is the same as in the previous birth event for the homology module. Notice that each W^j stays the same because $\mathsf{Z}(K_i) = \mathsf{Z}(K_{i+1})$.

Death in boundary module ($i \in \mathsf{N}^B(\mathcal{F})$): Since σ_i is not in a cycle in K_i and columns in Z, B, and C form a basis of $\mathsf{C}(K_i)$, at least one column in C contains σ_i. Whenever there are two columns $C[j]$, $C[k]$ in C containing σ_i with $b'_k \prec b'_j$, set $C[j] = C[j] + C[k]$, $B[j] = B[j] + B[k]$, and $U^j = U^j \boxplus U^k$ to remove σ_i from $C[j]$. After this, only one column $C[\lambda]$ in C contains σ_i and we have that $[b'_\lambda, i] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ ceases to be active. Notice that U^λ still generates a representative for $[b'_\lambda, i] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$. For each $k \in \{1, \ldots, q\} \setminus \{\lambda\}$, $[b'_k, i] \in \mathsf{Pers}^B(\mathcal{F}_i)$ extends to be $[b'_k, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ for which U^k now stays the same because $\sigma_i \notin C[k]$ so that $B[k] \in \mathsf{B}(K_{i+1})$. Finally, we delete $B[\lambda]$ from B and delete $C[\lambda]$ from C.

$K_i \xleftarrow{\sigma_i} K_{i+1}$ **is backward, ψ_i^* is injective:** We have:

Death in homology module ($i \in \mathsf{N}^H(\mathcal{F})$): We have that at least one column in Z contains σ_i. (To see this, notice that σ_i cannot be in a column in B because σ_i has no cofaces in K_i. So σ_i has to be in a column in Z because Z and B provide a basis for $\mathsf{Z}(K_i)$ and there is a cycle in K_i containing σ_i.) Whenever there are two columns $Z[j]$, $Z[k]$ in Z with $\hat{b}_k \prec \hat{b}_j$ containing σ_i, set $Z[j] = Z[j] + Z[k]$ and $W^j = W^j \boxplus W^k$ to remove σ_i from $Z[j]$. After this, only one column $Z[\lambda]$ in Z contains σ_i and we have that $[\hat{b}_\lambda, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ ceases to be active. The remaining processing resembles what is done in the death event for the boundary module and is omitted. Notice that we also need to remove σ_i from C and the details are provided in Appendix D.

Since $\mathsf{B}(K_i) = \mathsf{B}(K_{i+1})$, each $[b'_k, i] \in \mathsf{Pers}^B(\mathcal{F}_i)$ extends to be $[b'_k, i+1] \in \mathsf{Pers}^B(\mathcal{F}_{i+1})$ and the bundle U^k stays the same.

REMARK 4.2. *We can also consider our algorithm to have a 'pairing of birth/death points' structure as adopted by the algorithm for computing standard persistence [8], where, e.g., $\hat{b}_1, \ldots, \hat{b}_r$ are carried as 'unpaired' birth indices to be paired for the homology module.*

The following proposition from [4] (Proposition 9) helps draw our conclusion:

PROPOSITION 4.1. *Let $\pi : \mathsf{P}^H(\mathcal{F}_i) \to \mathsf{N}^H(\mathcal{F}_i)$ be a bijection. If every $b \in \mathsf{P}^H(\mathcal{F}_i)$ satisfies that $b \le \pi(b)$ and the interval $[b, \pi(b)]$ has a representative, then $\mathsf{Pers}^H(\mathcal{F}_i) = \{[b, \pi(b)] \mid b \in \mathsf{P}^H(\mathcal{F}_i)\}$.*

REMARK 4.3. *Similar facts hold for $\mathsf{P}^B(\mathcal{F}_i)$, $\mathsf{N}^B(\mathcal{F}_i)$, and $\mathsf{Pers}^B(\mathcal{F}_i)$.*

THEOREM 4.1. *The barcodes $\mathsf{Pers}^H(\mathcal{F})$ and $\mathsf{Pers}^B(\mathcal{F})$ along with the representatives for the intervals can be computed in $O(m^2 n)$ time and $O(mn)$ space.*

Proof. First, to see that the algorithm presented above runs in $O(m^2 n)$ time, we notice that there are no more than $O(n)$ summations of matrix columns and wire bundles in each iteration, which can be verified from the details presented in this section and Appendix D. Hence, each iteration runs in $O(mn)$ time where the costliest steps are the bundle summations. At the end of the algorithm, we also need to generate a representative for each interval from the maintained bundle. Generating representatives for all the $O(m)$ intervals can be done in $O(m^2 n)$ time (see the Algorithm EXTREP). The $O(m^2 n)$ complexity then follows. The space complexity follows from maintaining $O(m)$ wires each being a cycle of size $O(n)$, $O(n)$ bundles for the active intervals each of size $O(m)$, and the three matrices of size at most $O(n^2)$.

Based on Proposition 4.1, the correctness of the algorithm follows from the fact that wire bundles always correctly generate representatives for the intervals in our algorithm. The validity of the wire bundles follows from Proposition 3.1 (the only way a bundle changes after being created is by summations) and how we assign a bundle to an interval in the algorithm when an interval is created or ceases to be active (finalized). ∎

REMARK 4.4. *The key to achieving the $O(m^2 n)$ time complexity are the following two invariants maintained in our algorithm as described in Section 4.1: (i) pivots for the matrices Z and B are always distinct and (ii) $Z[j]$ always equals the last cycle in the representative for $[\hat{b}_j, i]$ generated by W^j. By invariant (i), we can obtain the sum in Equation (4.3) in $O(n^2)$ time by reductions. By invariant (ii), we can take the sum W^* of the bundles $\{W^j \mid j \in J\}$ based on Equation (4.3) for the finalized interval $[\hat{b}_\lambda, i]$ when a death happens in the homology module. It ensures that the last cycle in the representative for $[\hat{b}_\lambda, i] \in \mathsf{Pers}^H(\mathcal{F}_{i+1})$ generated by W^* satisfies the death condition in Definition 2.3. As evident in Appendix D, in order to maintain the distinctness of pivots, one cannot avoid summations of columns in B to columns in Z. Without incorporating the module $\mathsf{B}(\mathcal{F})$ and its bundles, invariant (ii) would not hold when columns in B are summed to columns in Z.*

5 Experiments

We generate zigzag filtrations using the oscillating Rips [14] which are produced from point clouds of size 2000 – 4000 sampled from triangular meshes (Space Shuttle from an online repository*; Bunny and Dragon from the Stanford Computer Graphics Laboratory). Table 1 lists the running time for the these filtrations with different maximum dimensions for the simplices taken.

Table 1: Running time for WIREDZIGZAG on several filtrations. All tests were run on a Ubuntu 20.04 server with two AMD EPYC 7513 2.6 GHz CPUs having 32 cores and 1TB memory (program is single-threaded).

Filtration	**Max. Dim.**	**Length**	**Runtime**
`bunny`	1	1,012,198	37S
`space_shuttle`	3	5,135,721	5M57S
`dragon`	4	5,811,311	5M24S

*Ryan Holmes: `http://www.holmes3d.net/graphics/offfiles/`

Acknowledgment

T. K. Dey was supported by NSF funds CCF 2049010 and DMS 2301360. T. Hou was supported by NSF fund CCF 2439255. D. Morozov was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program, under Contract Number DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory. The authors would also like to thank Aman Timalsina for implementing a preliminary version of the software.

References

[1] Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, 2010.

[2] Gunnar Carlsson, Vin de Silva, and Dmitriy Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, pages 247–256, 2009.

[3] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry*, pages 119–126, 2006.

[4] Tamal K. Dey and Tao Hou. Computing zigzag persistence on graphs in near-linear time. In *37th International Symposium on Computational Geometry*, 2021.

[5] Tamal K. Dey and Tao Hou. Fast computation of zigzag persistence. In *30th Annual European Symposium on Algorithms, ESA 2022, September 5-9, 2022, Berlin/Potsdam, Germany*, volume 244 of *LIPIcs*, pages 43:1–43:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[6] Tamal K. Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[7] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Soc., 2010.

[8] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463. IEEE, 2000.

[9] Peter Gabriel. Unzerlegbare Darstellungen I. *Manuscripta Mathematica*, 6(1):71–103, 1972.

[10] Clément Maria and Steve Y. Oudot. Zigzag persistence via reflections and transpositions. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 181–199. SIAM, 2015.

[11] Clément Maria and Hannah Schreiber. Discrete Morse theory for computing zigzag persistence. In *Algorithms and Data Structures - 16th International Symposium, WADS Proceedings*, volume 11646 of *Lecture Notes in Computer Science*, pages 538–552. Springer, 2019.

[12] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry*, pages 216–225, 2011.

[13] Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209. AMS Mathematical Surveys and Monographs, 2015.

[14] Steve Y. Oudot and Donald R. Sheehy. Zigzag zoology: Rips zigzags for homology inference. *Foundations of Computational Mathematics*, 15(5):1151–1186, 2015.

A Proof of Proposition 2.1

First, the fact that $\{[z_1^H], \dots, [z_{k'}^H]\}$ is a basis of $\mathsf{H}(K_j)$ and $\{z_1^B, \dots, z_k^B\}$ is a basis of $\mathsf{B}(K_j)$ follows from the definition of interval decomposition and representatives. Consider any cycle z in $\mathsf{Z}(K_j)$. Then, there exists a unique $\alpha_t \in \{0,1\}$ for each $1 \le j \le k'$ so that $[z] = \sum_t \alpha_t [z_t^H]$. Then, $[z] + \sum_t \alpha_t [z_t^H] = [z + \sum_t \alpha_t z_t^H] = 0$. It follows that $(z + \sum_t \alpha_t z_t^H) \in \mathsf{B}(K_j)$, which implies that there exists a unique β_ℓ for each $1 \le \ell \le k$ so that $z + \sum_t \alpha_t z_t^H = \sum_\ell \beta_\ell z_\ell^B$. So, $z = \sum_t \alpha_t z_t^H + \sum_\ell \beta_\ell z_\ell^B$ for unique α_j's, $1 \le t \le k'$ and β_ℓ's, $1 \le \ell \le k$. It follows that the union of the cycles $\{z_\ell^B\}$ and $\{z_t^H\}$ generate $\mathsf{Z}(K_j)$. Since $k + k' = \dim(\mathsf{B}(K_j)) + \dim(\mathsf{H}(K_j)) = \dim(\mathsf{Z}(K_j))$, they form a basis.

B Proof of Proposition 2.2

Case 1, $b < b'$: In this case, every cycle $\overline{z}_\alpha$, $\alpha \in [b', i]$, in $\mathsf{rep} \boxplus \mathsf{rep}'$ satisfies that $\overline{z}_\alpha = z_\alpha + z'_\alpha$. It can be verified that we only have three different cases: (i) $b \in \mathsf{P}^B(\mathscr{F}_i)$, $b' \in \mathsf{P}^H(\mathscr{F}_i)$, (ii) $b \in \mathsf{P}^B(\mathscr{F}_i)$, $b' \in \mathsf{P}^B(\mathscr{F}_i)$, and (iii) $b \in \mathsf{P}^H(\mathscr{F}_i)$, $b' \in \mathsf{P}^H(\mathscr{F}_i)$. We take up the case (i) and the proof for the other cases is similar. Assuming $K_\alpha \hookrightarrow K_{\alpha+1}$ is forward for $b' \le \alpha < i$, we have $\psi_\alpha^\#(z_\alpha) = z_\alpha = z_{\alpha+1}$ and $\psi_\alpha^*([z'_\alpha]) = [z'_{\alpha+1}]$. Therefore, $\psi_\alpha^*([\overline{z}_\alpha]) = \psi_\alpha^*([z_\alpha]) + \psi_\alpha^*([z'_\alpha]) = [z_{\alpha+1}] + [z'_{\alpha+1}] = [\overline{z}_{\alpha+1}]$ as required. The same applies when $K_\alpha \hookleftarrow K_{\alpha+1}$ is backward. Finally, we verify that the birth and death conditions hold for $\overline{z}_{b'}$. First assume that $K_{b'-1} \hookrightarrow K_{b'}$ is forward. For the birth condition, we have that $\overline{z}_{b'} = z_{b'} + z'_{b'} \in \mathsf{Z}(K_{b'}) \setminus \mathsf{Z}(K_{b'-1})$ because $z'_{b'} \in \mathsf{Z}(K_{b'}) \setminus \mathsf{Z}(K_{b'-1})$ and $z_{b'} = z_{b'-1} \in \mathsf{Z}(K_{b'}) \cap \mathsf{Z}(K_{b'-1})$. One can also verify the death condition for the cycle $\overline{z}_i$. For a backward $K_{b'-1} \hookleftarrow K_{b'}$, we omit the verification for the birth and death conditions. It follows that in case (i), $\mathsf{rep} \boxplus \mathsf{rep}'$ is a homology representative for $[b', i]$, $b' \in \mathsf{P}^H(\mathscr{F}_i)$, as required. We also have that the justification for case (ii) and (iii) can be similarly done.

Case 2, $b' < b$: In this case, we have $\overline{z}_\alpha = z'_\alpha$ for $\alpha \in [b', b-1]$ and $\overline{z}_\alpha = z_\alpha + z'_\alpha$ for $\alpha \in [b, i]$. We have only two possible cases: (i) $b \in \mathsf{P}^B(\mathscr{F}_i)$ and $b' \in \mathsf{P}^H(\mathscr{F}_i)$; (ii) $b, b' \in \mathsf{P}^H(\mathscr{F}_i)$ and $K_{b-1} \hookleftarrow K_b$ is backward. Again, using the case analysis, one can show that $\psi^*_\alpha([\overline{z}_\alpha]) = [\overline{z}_{\alpha+1}]$ if ψ^*_α is forward and $\psi^*_\alpha([\overline{z}_{\alpha+1}]) = [\overline{z}_\alpha]$ otherwise. Moreover, the birth and death conditions can also be verified easily implying that $\mathsf{rep} \boxplus \mathsf{rep}'$ in both cases is a homology representative for $[b', i]$, $b' \in \mathsf{P}^H(\mathscr{F}_i)$.

C Missing cases in the proof of Theorem 3.1

Case 3, $K_i \xleftarrow{\sigma_i} K_{i+1}$ and $i+1 \in \mathsf{P}^H(\mathscr{F}_{i+1})$: In this case, an interval $[b, i] \in \mathsf{Pers}^B(\mathscr{F}_i)$ does not extend to $i+1$ and a new interval $[i+1, i+1] \in \mathsf{Pers}^H(\mathscr{F}_{i+1})$ begins. Let $[b_1, i], \ldots, [b_k, i]$ be all the intervals in $\mathsf{Pers}^B(\mathscr{F}_i)$ with representatives $\mathsf{rep}_1, \ldots, \mathsf{rep}_k$ respectively, and let z^j_i be the cycle at index i in rep_j for each $1 \leq j \leq k$. Since $z^j_i \in \mathsf{B}(K_i)$, z^j_i has a 'bounding chain' $c^j \in \mathsf{C}(K_i)$ s.t. $z^j_i = \partial(c^j)$. Assuming after reindexing $z^j_1, \ldots, z^j_\ell$ are all the cycles whose bounding chains contain σ_i where $b_1 \prec \cdots \prec b_\ell$. We add rep_1 to $\mathsf{rep}_2, \ldots, \mathsf{rep}_\ell$ to remove σ_i from their bounding chains. Then, the new representatives $\mathsf{rep}'_2 := \mathsf{rep}_1 \boxplus \mathsf{rep}_2, \ldots, \mathsf{rep}'_\ell := \mathsf{rep}_1 \boxplus \mathsf{rep}_\ell$ for the intervals $[b_2, i], \ldots, [b_\ell, i]$ can extend to $i+1$ because their bounding chains now do not contain σ_i. By Proposition 3.1, $W^{[b_j,i]} \boxplus W^{[b_1,i]}$ represents $[b_j, i+1] \in \mathsf{Pers}^B(\mathscr{F}_{i+1})$ for $2 \leq j \leq \ell$. So, we update $W^{[b_j,i]}$ as $W^{[b_j,i]} \boxplus W^{[b_1,i]}$ for $2 \leq j \leq k$. The interval $[b_1, i]$ does not extend to $i+1$ with the wire bundle $W^{[b_1,i]}$ still representing $[b_1, i] \in \mathsf{Pers}^B(\mathscr{F}_{i+1})$. A new interval $[i+1, i+1] \in \mathsf{Pers}^H(\mathscr{F}_{i+1})$ begins with a representative $\{\partial\sigma_i\}$ which is generated by a new wire $\omega_{i+1} = \partial\sigma_i$. Subsets of the wire bundle $W_{i+1} = W_i \cup \{\omega_{i+1}\}$ then generate representatives for all intervals in $\mathsf{Pers}^H(\mathscr{F}_{i+1}) \cup \mathsf{Pers}^B(\mathscr{F}_{i+1})$.

Case 4, $K_i \xleftarrow{\sigma_i} K_{i+1}$ and $i \in \mathsf{N}^H(\mathscr{F}_i)$: In this case, an interval $[b, i] \in \mathsf{Pers}^H(\mathscr{F}_i)$ does not extend to $i+1$. Let $\mathsf{rep}_1, \ldots, \mathsf{rep}_k$ be all the representatives for $[b_1, i], \ldots, [b_k, i] \in \mathsf{Pers}^H(\mathscr{F}_i)$ respectively whose cycles at index i contain σ_i. WLOG, assume that $b_1 \prec \cdots \prec b_k$. We cannot extend these representatives to $i+1$ because $\sigma_i \notin K_{i+1}$. We add rep_1 to $\mathsf{rep}_2, \ldots, \mathsf{rep}_k$ to obtain new representatives $\mathsf{rep}'_2, \ldots, \mathsf{rep}'_k$ for the intervals whose cycles at index i now do not contain σ_i. Similar to previous cases, the bundle $W^{[b_j,i]} \boxplus W^{[b_1,i]}$ represents $[b_j, i+1] \in \mathsf{Pers}^H(\mathscr{F}_{i+1})$ for $2 \leq j \leq k$. So, we update $W^{[b_j,i]}$ as $W^{[b_j,i]} \boxplus W^{[b_1,i]}$ for $2 \leq j \leq k$. The interval $[b_1, i]$ does not extend to $i+1$ and rep_1 remains a representative for $[b_1, i] \in \mathsf{Pers}^H(\mathscr{F}_{i+1})$.

D Implementation details

We provide implementation details for the algorithm presented in Section 4. For a column c of the matrices maintained, we denote the pivot of c as pivot(c). Also, in our algorithm, each simplex σ_i added in $\mathscr{F}$ is assigned an `id` i. This means that a simplex has a new `id` when it is added again after being deleted. We then present the details for the different cases.

D.1 Forward $K_i \xhookrightarrow{\sigma_i} K_{i+1}$

We need to determine whether $\partial\sigma_i$ is already a boundary in K_i. If this is true, a new cycle containing σ_i is created in K_{i+1} and ψ^*_i is injective; otherwise, the homology class $[\partial\sigma_i]$ becomes trivial in $\mathsf{H}(K_{i+1})$ and ψ^*_i is surjective. To determine this, we perform reductions on $\partial\sigma_i$ and the columns in Z and B to get a sum $\partial\sigma_i = \sum_{j\in J} Z[j] + \sum_{k\in I} B[k]$. We then have that $\partial\sigma_i$ is a boundary in K_i iff $J = \varnothing$.

D.1.1 ψ^*_i is injective

Since $\partial\sigma_i = \sum_{k\in I} B[k]$, we let the new wire ω_{i+1} containing σ_i be $\omega_{i+1} = \sigma_i + \sum_{k\in I} C[k]$, where $\partial(\sigma_i + \sum_{k\in I} C[k]) = \partial\sigma_i + \sum_{k\in I} B[k] = 0$. Notice that as mentioned, we need to add ω_{i+1} as a new column to the matrix Z. Since pivot$(\omega_{i+1}) = i$, columns in Z and B still have distinct pivots.

D.1.2 ψ^*_i is surjective

The subset J derived from the reductions is the same as the subset J in Equation (4.3) in the corresponding case of Section 4. So the processing for the corresponding case described in Section 4 can be directly performed. Notice that we add a new column to B in this case. Since the pivot of the new column of B may conflict with the pivot of another column in Z or B, we use a loop to repeatedly sum two columns whose pivots are the same until the pivots become distinct again. In each iteration of the loop, three cases can happen:

1. Two columns $B[j]$ and $B[k]$ have the same pivot: WLOG, assume that $b'_k \prec b'_j$. Let $B[j] = B[j] + B[k]$, $C[j] = C[j] + C[k]$, and $U^j = U^j \boxplus U^k$.
2. Two columns $Z[j]$ and $B[k]$ have the same pivot: We have $b'_k \prec \hat{b}_j$. Let $Z[j] = Z[j] + B[k]$ and $W^j = W^j \boxplus U^k$.
3. Two columns $Z[j]$ and $Z[k]$ have the same pivot: WLOG, assume that $\hat{b}_k \prec \hat{b}_j$. Let $Z[j] = Z[j] + Z[k]$ and $W^j = W^j \boxplus W^k$.

Since in each iteration of the above loop we change only one column of Z and B, there are at most two columns of Z and B with the same pivot at any time. Hence, the above loop ends in no more than n iterations because the pivot of the two clashed columns is always decreasing.

D.2 Backward $K_i \xleftarrow{\sigma_i} K_{i+1}$ We need to determine whether σ_i is in a cycle z in K_i. If this is true, z is a cycle in K_i but not in K_{i+1} indicating that ψ_i^* is injective; otherwise, ψ_i^* is surjective. Since columns in Z and B form a basis for $\mathsf{Z}(K_i)$, we only need to check whether σ_i is in a column in Z or B. Moreover, since σ_i has no cofaces in K_i, we have that σ_i cannot be in a boundary in K_i. Therefore, we only need to check whether σ_i is in a column in Z.

D.2.1 ψ_i^* is surjective Since columns in Z, B, and C form a basis for $\mathsf{C}(K_i)$ and σ_i is not in a column in Z or B, we have that σ_i must be in at least one column of C. Since $\sigma_i \notin K_{i+1}$, we need to remove σ_i from C when proceeding from K_i to K_{i+1}. To do this, we use a loop to repeatedly sum two columns in C containing σ_i until only one column in C contains σ_i. Notice that whenever we sum two columns in C, we also need to sum the corresponding columns in B and their wire bundles. Hence, the summations have to respect the order '$\prec$'. We use the following loop to perform the summations:

```
α_1, ..., α_ℓ ← indices of all columns of C containing σ_i
sort and rename α_1, ..., α_ℓ s.t. b'_{α_1} ≺ ··· ≺ b'_{α_ℓ}.
c_1 ← C[α_1]
c_2 ← B[α_1]
U ← U^{α_1}
for α ← α_2, ..., α_ℓ do:
    if pivot(B[α]) > pivot(c_2) then:
        C[α] ← C[α] + c_1
        B[α] ← B[α] + c_2
        U^α ← U^α ⊞ U
    else:
        temp_c1 ← C[α]
        C[α] ← C[α] + c_1
        c_1 ← temp_c1
        temp_c2 ← B[α]
        B[α] ← B[α] + c_2
        c_2 ← temp_c2
        temp_U ← U^α
        U^α ← U^α ⊞ U
        U ← temp_U
```

We always maintain the following invariants for the loop: (i) $c_2 = \partial(c_1)$; (ii) c_2 is the last cycle (at index i) in the representative generated by U; (ii) the birth index corresponding to c_2 (and U) is always less than b'_α in the total order '$\prec$'; (iv) c_2 along with $B[\alpha_2], \ldots, B[\alpha_\ell]$ have distinct pivots. When the loop terminates, we are left with a single column $C[\lambda] := C[\alpha_1]$ in C containing σ_i. Notice that $B[\lambda] = \partial(C[\lambda]) = \partial(C[\lambda] \setminus \{\sigma_i\}) + \partial\sigma_i$, where $C[\lambda] \setminus \{\sigma_i\} \subseteq K_{i+1}$. This indicates that $B[\lambda]$ is homologous to $\partial\sigma_i$ in K_{i+1}. So we let the new wire ω_{i+1} be $B[\lambda]$ and need to add ω_{i+1} as a new column to Z. Notice that we also delete $B[\lambda]$ and $C[\lambda]$ from B and C respectively. Since the pivot of the newly added column in Z may clash with that of another column in B or Z, we need to perform summations as in Section D.1.2 to make the pivots distinct again. Notice that assumptions on the matrices Z, B, and C still hold. For example, columns in B still form a basis for $\mathsf{B}(K_{i+1})$ because columns in B are still linearly independent and the dimension of $\mathsf{B}(K_{i+1})$ is one less than that of $\mathsf{B}(K_i)$.

D.2.2 ψ_i^* **is injective** We first update C so that no columns of C contain σ_i. Let $Z[k]$ be a column of Z containing σ_i. For each column $C[j]$ of C containing σ_i, set $C[j] = C[j] + Z[k]$. Notice that $\partial(C[j])$ stays the same but the updated $C[j]$ does not contain σ_i.

As indicated in Section 4, whenever there are two columns in Z which contain σ_i, we sum the two columns and their corresponding bundles to remove σ_i from one column. We implement the summations as follows, which is similar to the loop in Section D.2.1:

$\alpha_1, \ldots, \alpha_\ell \leftarrow$ indices of all columns of Z containing σ_i
sort and rename $\alpha_1, \ldots, \alpha_\ell$ s.t. $\hat{b}_{\alpha_1} \prec \cdots \prec \hat{b}_{\alpha_\ell}$.
$z \leftarrow Z[\alpha_1]$
$W \leftarrow W^{\alpha_1}$
for $\alpha \leftarrow \alpha_2, \ldots, \alpha_\ell$ **do**:
    if pivot$(Z[\alpha]) >$ pivot(z) **then**:
        $Z[\alpha] \leftarrow Z[\alpha] + z$
        $W^\alpha \leftarrow W^\alpha \boxplus W$
    else:
        `temp_z` $\leftarrow Z[\alpha]$
        $Z[\alpha] \leftarrow Z[\alpha] + z$
        $z \leftarrow$ `temp_z`
        `temp_W` $\leftarrow W^\alpha$
        $W^\alpha \leftarrow W^\alpha \boxplus W$
        $W \leftarrow$ `temp_W`
delete the column $Z[\alpha_1]$ from Z

In the above pseudocodes, α_1 is the index 'λ' as in the corresponding case in Section 4.