



**HAL**  
open science

# Computing Persistent Homology with Various Coefficient Fields in a Single Pass

Jean-Daniel Boissonnat, Clément Maria

► **To cite this version:**

Jean-Daniel Boissonnat, Clément Maria. Computing Persistent Homology with Various Coefficient Fields in a Single Pass. [Research Report] RR-8436, 2013, pp.16. hal-00922572v2

**HAL Id: hal-00922572**

**<https://inria.hal.science/hal-00922572v2>**

Submitted on 7 Mar 2014 (v2), last revised 9 Jan 2020 (v5)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Computing Persistent Homology with Various Field Coefficients in a Single Pass

Jean-Daniel Boissonnat, Clément Maria

**RESEARCH  
REPORT**

**N° 8436**

December 2013

Project-Team Geometrica

ISSN INRIA/RR--8436--FR+ENG

ISSN 0249-6399





## Computing Persistent Homology with Various Field Coefficients in a Single Pass

Jean-Daniel Boissonnat<sup>\*†</sup>, Clément Maria<sup>‡†</sup>

Project-Team Geometrica

Research Report n° 8436 — December 2013 — ?? pages

---

This research has been partially supported by the 7th Framework Programme for Research of the European Commission, under FET-Open grant number 255827 (CGL Computational Geometry Learning).

\* Jean-Daniel.Boissonnat@inria.fr

† INRIA Sophia Antipolis Méditerranée

‡ Clement.Maria@inria.fr

**RESEARCH CENTRE  
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93  
06902 Sophia Antipolis Cedex

**Abstract:** In this article, we introduce the *multi-field persistence diagram* for the persistence homology of a filtered complex. It encodes compactly the *superimposition* of the persistence diagrams of the complex with several field coefficients, and provides a substantially more precise description of the topology of the filtered complex. Specifically, the multi-field persistence diagram encodes the Betti numbers of integral homology and the prime divisors of the torsion coefficients of the underlying shape. Moreover, it enjoys similar stability properties as the ones of standard persistence diagrams, with the appropriate notion of distance. These properties make the multi-field persistence diagram a useful tool in computational topology.

We develop algorithms to compute the multi-field persistence diagram of a filtered complex as well as distances between multi-field persistence diagrams. On the one hand, we introduce *modular reconstruction for Gaussian eliminations*, an algorithmic method of independent interest for matrix algorithms. We apply it to the persistent cohomology algorithm and obtain an output-sensitive algorithm in the size  $m'$  of the multi-field persistence diagram, where the size  $m'$  is the number of distinct points in the superimposition of diagrams with the several field coefficients. For a filtered complex of size  $m$ , the standard persistence cohomology algorithm computes the persistence diagram in any field in time  $O(f(m))$ ; the modular reconstruction algorithm computes the multi-field persistence diagram of size  $m'$  in time  $O(f(m')A)$ , where  $A$  is a factor measuring the complexity of arithmetic operations. In practice,  $m'$  is close to  $m$ , and in particular  $m' \ll r \times m$  when multi-field persistent homology is computed in  $r$  distinct fields. On the other hand, we generalize the computation of graph matching to the multi-field persistence setting in order to compute distances between multi-field persistence diagrams. For standard persistence diagrams of size  $O(m)$ , the distance between diagrams can be computed in time  $O(g(m))$ ; our algorithm computes the distance between multi-field persistence diagrams of size  $O(m')$  in time  $O(g(m')A\sqrt{t})$ , where  $t$  counts the number of distinct prime divisors of the torsion coefficients of the filtered complex.

We provide theoretical and experimental analyses of the algorithms. In particular, we show that  $t$  and  $A$  are very small in real-world examples and the multi-field algorithms are, in practice, as fast as algorithms that compute persistent homology in a single field.

**Key-words:** Computational topology, persistent homology, modular reconstruction

## Calcul simultané de l'homologie persistante sur plusieurs corps

**Résumé :** Dans cet article, on introduit les diagrammes d'homologie persistante multi-corps d'un complexe filtré. Ces diagrammes codent de manière compacte la superposition des diagrammes de persistance d'un complexe filtré calculés dans plusieurs corps, et fournissent une information plus précise de la topologie du complexe. Plus précisément, le diagramme de persistance multi-corps code les nombres de Betti de l'homologie sur les entiers et les diviseurs premiers des coefficients de torsion. Avec une notion de distance appropriée, ce diagramme jouit de propriétés de stabilité analogues à celles des diagrammes standard.

**Mots-clés :** Topologie algorithmique, homologie persistante, reconstruction modulaire

# 1 Introduction

Persistent homology [?, ?] is an algebraic method for measuring the topological features of the sublevel sets of a function defined on a topological space. Its generality and stability [?] with regard to noise have made it a widely used tool for the study of data. Very efficient methods have been developed for computing persistent homology [?, ?] and its dual, persistent cohomology [?, ?, ?]. All the algorithms are based on the reduction of a matrix with coefficients in a field [?]. At the algebraic level [?], persistent homology with field coefficients admits a canonical decomposition, and is represented by means of a persistence diagram. The major restriction of homology with coefficients in a field is that it is blind to torsion. Torsion can be pictured geometrically as a “twisting” of the shape and happens frequently in topological data analysis where, for example, Klein bottles appear naturally [?, ?]. Algebraically, torsion is characterized by cyclic subgroups of the integral homology groups. When computed with field coefficients, these subgroups may either vanish or appear as “infinite”, and consequently obfuscate the study of the topology of data. A simple solution is to compute persistent homology with different field coefficients and track the differences in the persistence diagram, in order to detect the presence of torsion. We build on this idea and introduce the *multi-field persistence diagram*, a compact representation of the superimposition of persistence diagrams for a family of field coefficients  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ . We assume that  $r$  is relatively small ( $\leq 100$ ) and the size  $m'$  of the multi-field persistence diagram is close to the size  $m$  of the filtered complex; in particular  $m' \ll r \times m$ . As will be shown, this assumption holds in most practical situations. We study, in Section ??, the properties of multi-field persistence diagrams, we show how to deduce torsion coefficients and generalize the stability theorem of persistent homology [?]. In Section ??, we generalize the persistent cohomology algorithm to compute multi-field persistence diagrams in an output-sensitive way. To do so, we introduce *modular reconstruction for Gaussian elimination*, which is an algorithmic method of independent interest for matrix reduction, and which can, in particular, be used for both persistent homology and cohomology.

When the cohomology algorithm computes a diagram in time  $O(f(m))$ , our modular reconstruction algorithm computes the multi-field persistence diagram in time  $O(f(m')A)$ , where  $A$  is a factor representing the arithmetic complexity for elementary operations. Finally, we describe in Section ?? an efficient algorithm to compute distances between multi-field persistence diagrams. Specifically, if an algorithm computes the distance between two persistence diagrams in a single field in time  $O(g(m))$ , our algorithm returns the distance between two multi-field persistence diagrams in time  $O(g(m')A\sqrt{t})$ , where  $t$  counts the number of distinct prime divisors of the torsion coefficients in the filtered complex.

For computational topologists, the multi-field persistence diagram offers several advantages over the persistence diagram in a single field. It provides a substantially more precise description of topology, by detecting characterizing torsion, in addition to providing the Betti numbers of integral homology. Moreover, it admits fast algorithms for construction and manipulation. The algorithms described in this article are, in practice, as fast as the algorithms for persistence homology in a single field. We introduce some background and notations:

*Simplicial Homology:* A *simplicial complex*  $\mathbf{K}$  is a collection of simplices  $\{\sigma\}$  such that  $\tau \subseteq \sigma \in \mathbf{K} \Rightarrow \tau \in \mathbf{K}$ . The dimension  $p = |\sigma| - 1$  of  $\sigma$  is its number of elements minus 1. The group of  $p$ -chains, denoted  $\mathbf{C}_p(\mathbf{K}, \mathcal{R})$ , of  $\mathbf{K}$  is the group of formal sums of  $p$ -simplices with  $\mathcal{R}$  coefficients. The *boundary operator* is a linear operator  $\partial_p : \mathbf{C}_p(\mathbf{K}, \mathcal{R}) \rightarrow \mathbf{C}_{p-1}(\mathbf{K}, \mathcal{R})$  such that  $\partial_p \sigma = \partial_p[v_0, \dots, v_p] = \sum_{i=0}^p (-1)^i [v_0, \dots, \widehat{v}_i, \dots, v_p]$ , where  $\widehat{v}_i$  means  $v_i$  is deleted from the list. Denote by  $\mathbf{Z}_p(\mathbf{K}, \mathcal{R})$  and  $\mathbf{B}_{p-1}(\mathbf{K}, \mathcal{R})$  the kernel and the image of  $\partial_p$  respectively. Since  $\partial_p \circ \partial_{p+1} = 0$ , we define the  $p^{\text{th}}$  homology group  $\mathbf{H}_p(\mathbf{K}, \mathcal{R})$  of  $\mathbf{K}$  by the quotient  $\mathbf{H}_p(\mathbf{K}, \mathcal{R}) = \mathbf{Z}_p(\mathbf{K}, \mathcal{R}) / \mathbf{B}_p(\mathbf{K}, \mathcal{R})$ . The dual of the homology groups are the *cohomology groups*  $H^p(\mathbf{K}, \mathcal{R})$ , which contain *cocycles*. A  $p$ -cocycle is a linear form  $\mathbf{C}_p(\mathbf{K}, \mathcal{R}) \rightarrow \mathcal{R}$ . It will be convenient for the description of algorithms to extend the linear forms to the group  $\mathbf{C}_*(\mathbf{K}, \mathcal{R}) = \mathbf{C}_0(\mathbf{K}, \mathcal{R}) \oplus \dots \oplus \mathbf{C}_d(\mathbf{K}, \mathcal{R})$ , setting  $\phi : \mathbf{C}_p(\mathbf{K}, \mathcal{R}) \rightarrow \mathcal{R}$  to 0 on every simplex of dimension different from  $p$ .

For any integer  $n > 1$ , let  $\mathbb{Z}_n$  be the ring  $(\mathbb{Z}/n\mathbb{Z}, +, \times)$  and  $\mathbb{F}$  be a field. The *fundamental theorem of finitely generated abelian groups* [?] states that homology groups admits a *primary decomposition*:  $\mathbf{H}_p(\mathbf{K}, \mathbb{Z}) \cong \mathbb{Z}^{\beta_p(\mathbb{Z})} \bigoplus_{q \text{ prime}} \left( \mathbb{Z}_{q^{k_1}} \oplus \dots \oplus \mathbb{Z}_{q^{k_{t(p,q)}}} \right)$  for uniquely defined integers  $\beta_p(\mathbb{Z})$ , called the *Betti numbers*, and  $k_i > 0$  and  $t(p, q) \geq 0$  over all prime numbers  $q$ . If  $t(p, q) > 0$ , the integers  $q^{k_1}, \dots, q^{k_{t(p,q)}}$  of the primary decomposition are called *torsion coefficients*, each of them admitting  $q$  as unique *prime divisor*. With field coefficients, torsion is hidden and we have  $\mathbf{H}_p(\mathbf{K}, \mathbb{F}) \cong \mathbb{F}^{\beta_p(\mathbb{F})}$ . We detail in Section ?? the relation

between the Betti numbers  $\beta_p(\mathbb{Z})$  and  $\beta_p(\mathbb{F})$ .

*Persistence:* A *filtration* of a simplicial complex is a function  $f : \mathbf{K} \rightarrow \mathbb{R}$  verifying  $f(\tau) \leq f(\sigma)$  whenever  $\tau \subseteq \sigma$ . The sequence  $[\sigma_i]_{i=1, \dots, m}$  sorted according to increasing  $f$  values induces:

$\emptyset = \mathbf{K}_0 \subsetneq \mathbf{K}_1 \subsetneq \dots \subsetneq \mathbf{K}_{m-1} \subsetneq \mathbf{K}_m = \mathbf{K}, \mathbf{K}_i = \mathbf{K}_{i-1} \cup \{\sigma_i\}$  and  $0 = \mathbf{H}_p(\mathbf{K}_0, \mathbb{F}) \rightarrow \mathbf{H}_p(\mathbf{K}_1, \mathbb{F}) \rightarrow \dots \rightarrow \mathbf{H}_p(\mathbf{K}_{m-1}, \mathbb{F}) \rightarrow \mathbf{H}_p(\mathbf{K}_m, \mathbb{F}) = \mathbf{H}_p(\mathbf{K}, \mathbb{F})$ . Persistent homology studies the later sequence.

We refer to [?, ?] for an introduction to homology and cohomology, and to [?] for an introduction to persistent homology.

## 2 Multi-Field Persistence Diagram

When persistent homology is considered with field coefficients  $\mathbb{F}$ , the sequence of homology groups admits a canonical decomposition into a pairing of simplices [?] of  $\mathbf{K}$ . The  $p^{\text{th}}$  *persistence diagram*, noted  $D(\mathbf{K}, \mathbb{F})$  (we omit the dimension  $p$ ), is a plot of the points  $(f(\sigma_i), f(\sigma_j))$  for each persistent pair  $(\sigma_i, \sigma_j)$ , and  $(f(\sigma_\ell), +\infty)$  for the unpaired simplices  $\sigma_\ell$ . The first simplex of a pair is a *creator*, of dimension  $p$ , and the second a *destructor*, of dimension  $p + 1$ .

We consider a family of finite fields  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ , for distinct prime numbers  $q_1, \dots, q_r$  (usually the first  $r$  prime numbers), and a filtered complex  $\mathbf{K}$ . The *multi-field (MF) persistence diagram*  $D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  of  $\mathbf{K}$  represents the *superimposition* of the persistence diagrams  $D(\mathbf{K}, \mathbb{Z}_{q_1}) \cdots D(\mathbf{K}, \mathbb{Z}_{q_r})$  in each field. If a point appears in  $D(\mathbf{K}, \mathbb{Z}_{q_s})$ , we attach to the point its index  $s$ .

The size of a persistence diagram  $|D(\mathbf{K}, \mathbb{Z}_q)|$  is the number of persistent pairs  $(\sigma_i, \sigma_j)$  plus the number of unpaired simplices  $\sigma_\ell$ . Similarly, we define the *size* of the MF-persistence diagram  $|D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})|$  to be the number of distinct persistent pairs of simplices  $(\sigma_i, \sigma_j)$  plus the number of distinct unpaired simplices  $\sigma_\ell$  in  $D(\mathbf{K}, \mathbb{Z}_{q_s})$ , for all  $1 \leq s \leq r$ . We give a linear size encoding of MF-persistence diagrams in Section ??.

**Topology Inference and Torsion:** For a topological space  $\mathbb{X}$ , the *universal coefficient theorem for homology* [?] establishes the relationship between the homology with  $\mathbb{Z}$  coefficients and the homology with coefficients in a field  $\mathbb{F}$ . We use the following corollary of the theorem:

**Corollary 1 (Universal Coefficient Theorem)** *For a topological space  $\mathbb{X}$ , with integral homology groups  $\mathbf{H}_p(\mathbb{X}, \mathbb{Z})$ , and a field  $\mathbb{Z}_q$  for  $q$  prime,  $\mathbf{H}_p(\mathbb{X}, \mathbb{Z}_q)$  verifies:*

$\beta_p(\mathbb{Z}_q) = \beta_p(\mathbb{Z}) + t(p, q) + t(p - 1, q)$ , where  $\beta_p(\mathbb{Z})$  and  $\beta_p(\mathbb{Z}_q)$  are the Betti numbers of  $\mathbf{H}_p(\mathbb{X}, \mathbb{Z})$  and  $\mathbf{H}_p(\mathbb{X}, \mathbb{Z}_q)$  respectively, and  $t(j, q)$  is the number of  $\mathbb{Z}_{q^k}$  summands in the primary decomposition of  $\mathbf{H}_j(\mathbb{X}, \mathbb{Z})$  for any integer  $k \geq 1$ .

Suppose  $\{q_1, \dots, q_r\}$  are the first  $r$  prime numbers and  $q_r$  is a strict upper bound on the prime divisors of the torsion coefficients of  $\mathbb{X}$ . Given the Betti numbers of  $\mathbb{X}$  in all fields  $\mathbb{Z}_{q_s}, 1 \leq s \leq r$ , and knowing that  $H_0$  has no torsion and  $\beta_p(\mathbb{Z}_{q_r}) = \beta_p(\mathbb{Z})$  for all dimensions  $p$ , we deduce from Corollary ?? the recurrence formula  $t(p, q_s) = \beta_p(\mathbb{Z}_{q_s}) - \beta_p(\mathbb{Z}_{q_r}) - t(p - 1, q_s)$ , from which we compute the value of  $t(p, q)$  for every dimension  $p$  and prime  $q$  dividing a torsion coefficient of  $\mathbb{X}$ . We note that the powers  $k_i$  remain unknown; to simplify notations, we write  $\mathbb{Z}_{q^{\alpha_1}} \oplus \dots \oplus \mathbb{Z}_{q^{\alpha_t}}$  as  $\mathbb{Z}_{q^*}^{(t)}$ , for any family of integers  $\alpha_i \geq 1$ . Consequently, computing persistent homology in the fields  $(\mathbb{Z}_{q_s})_{s=1, \dots, r}$ , for a filtration approximating  $\mathbb{X}$ , allows us to infer the Betti numbers of  $\mathbb{X}$  and to compute an expression of the form  $\mathbf{H}_p(\mathbb{X}, \mathbb{Z}) \cong \mathbb{Z}^{\beta_p(\mathbb{Z})} \bigoplus_{q \text{ prime}} \mathbb{Z}_{q^*}^{(t(p, q))}$  for the homology groups of  $\mathbb{X}$ .

We known [?] that, when the sequence of complexes of the filtered complex  $\mathbf{K}$  approximates an underlying topological space  $\mathbb{X}$  at various scales, the persistent Betti numbers of  $\mathbf{K}$  coincide with the Betti numbers  $\beta_p(\mathbb{F})$  of  $\mathbb{X}$ . We generalize the representation of the persistence diagram so as to unveil torsion, knowing  $D(\mathbf{K}, \mathbb{Z}_{q_s}), 1 \leq s \leq r$  (see Figure ??). To each point  $(f(\sigma_i), f(\sigma_j))$  in some  $D(\mathbf{K}, \mathbb{F})$ , we associate the triangle  $\mathcal{T}_{i,j}(\mathbb{F})$  “under” this point: i.e. the set  $\{(x, y) : x \geq f(\sigma_i), y < f(\sigma_j), x \leq y\}$  which contains all points  $(x, y)$  such that the homology feature created by  $\sigma_i$  and destroyed by  $\sigma_j$  persists in the range  $[x; y]$ . The persistent Betti number  $\beta_p^{x,y}(\mathbb{F})$  is the number of such triangles covering the point  $(x, y)$  in  $D(\mathbf{K}, \mathbb{F})$ . Knowing  $D(\mathbf{K}, \mathbb{Z}_{q_s}), 1 \leq s \leq r$ , we get a sequence  $\beta_p^{x,y}(\mathbb{Z}_{q_s}), 1 \leq s \leq r$ , for every point  $(x, y)$ : we deduce an expression of the form  $\mathbb{Z}^{\beta_p(\mathbb{Z})} \bigoplus_{s=1, \dots, r} \mathbb{Z}_{q_s^*}^{(t(p, q_s))}$  for every cell in the subdivision induced by the triangles of all diagrams. The persistent homology groups in this representation are the ones whose associated cells have big areas. This leads to the representation in Figure ?? of the MF-persistence diagram of  $\mathbf{H}_1$  for a point cloud sampling a Klein bottle, where the homology group  $H_1 \cong \mathbb{Z} \oplus \mathbb{Z}_2$  appears as persistent.



**Distance between Multi-Field Persistence Diagrams and Stability:** For two filtered complexes  $\mathbf{K}$  and  $\widehat{\mathbf{K}}$ , and a field  $\mathbb{F}$ , we define the bottleneck distance [?] between  $D(\mathbf{K}, \mathbb{F})$  and  $D(\widehat{\mathbf{K}}, \mathbb{F})$ . Let  $P$  (resp.  $\widehat{P}$ ) be the set of points of  $D(\mathbf{K}, \mathbb{F})$  (resp.  $D(\widehat{\mathbf{K}}, \mathbb{F})$ ) plus all points on the diagonal  $x = y$ . The bottleneck distance between the diagrams is:  $d_B(D(\mathbf{K}, \mathbb{F}), D(\widehat{\mathbf{K}}, \mathbb{F})) = \inf_{\eta: P \rightarrow \widehat{P}} \sup_{X \in P} \|X - \eta(X)\|_\infty$  over all bijections  $\eta$  from  $P$  to  $\widehat{P}$ . We define the *MF-bottleneck distance* between two MF-persistence diagrams  $D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  and  $D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  to be  $d_{MF}(D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}), D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})) = \max_{1 \leq s \leq r} d_B(D(\mathbf{K}, \mathbb{Z}_{q_s}), D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_s}))$ . A direct consequence is that stability of MF-persistent diagrams is a corollary of the key *stability theorem* [?] of persistence homology.

**Corollary 2 (Stability for MF-Persistence Diagrams)** *Let  $\mathbf{K}$  be a simplicial complex and  $f, g : \mathbf{K} \rightarrow \mathbb{R}$  two monotonic functions inducing the filtrations  $\mathbf{K}_f$  and  $\mathbf{K}_g$  on  $\mathbf{K}$ : the MF-bottleneck distance between the MF-diagrams of  $\mathbf{K}_f$  and  $\mathbf{K}_g$  is bounded from above by  $\|f - g\|_\infty$ .*

In the next sections, we present a compact encoding of MF-persistence diagrams, and we design algorithms to compute MF-persistence diagrams and MF-bottleneck distances. Our framework is especially useful when the size  $m'$  of the MF-persistence diagram is close to  $m$ , the size of a persistence diagram in a single field. In particular,  $m' \ll r \times m$ . Indeed, torsion is due to a geometric “twisting” of shapes and rarely happens at small scales on point cloud data. Consequently, most points match on the superimposition of diagrams. Moreover, it is usually true that  $r$  is small ( $\leq 100$ ) because the torsion coefficients are usually small, which implies that  $r$  does not need to be big to ensure that the family  $\{q_1, \dots, q_r\}$  of first  $r$  prime numbers divide all the torsion coefficients.

### 3 Persistent Cohomology Algorithm for Multi-Field Persistence Diagram

In this section, we present the algorithm to compute persistent cohomology [?, ?, ?] of a filtered complex  $K$ , which is a particular case of a reduction to row echelon form of an integer matrix, modulo a prime number. The reduction of the integer matrix does not exist in general, but the reduction of the corresponding matrix in  $\mathbb{Z}/q\mathbb{Z}$  for  $q$  prime exists [?], and gives the persistence diagram  $D(\mathbf{K}, \mathbb{Z}_q)$ . The persistence diagram  $D(\mathbf{K}, \mathbb{Z}_q)$  is encoded as a list of pairs  $(f(\sigma_i), f(\sigma_j))$  and  $(f(\sigma_\ell), +\infty)$ , corresponding to the persistent pairs of simplices. This representation has size  $\Theta(|D(\mathbf{K}, \mathbb{Z}_q)|)$ . We extend this algorithm to compute MF-persistence diagrams. For each pair of simplices  $(\sigma_i, \sigma_j)$  (resp. unpaired simplex  $\sigma_{ell}$ ) which is persistent in homology with  $\mathbb{Z}_{q_s}$  coefficients, for  $s \in S \subseteq [r]$ , we store a triplet  $(f(\sigma_i), f(\sigma_j), S)$  (resp.  $(f(\sigma_i), +\infty, S)$ ).  $S$  will later be represented by the product  $Q_S = \prod_{s \in S} q_s$ . The representation contains  $|D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})|$  triplets.

Recall that the ring  $\mathbb{Z}_n$  is a field iff  $n$  is prime; the cartesian product  $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$  refers to the ring  $(\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}, +, \times)$  with componentwise addition and multiplication. We denote by 0 and 1 the additive and multiplicative identity respectively, for any ring.

**Reduction Algorithm:** We are given a filtered complex  $\mathbf{K}$  with  $m$  faces. Let  $[\sigma_1, \dots, \sigma_m]$  be the set of simplices of  $\mathbf{K}$  ordered according to the filtration. For a ring  $\mathcal{R}$ , let  $H(\mathcal{R})$  be the cohomology matrix of  $\mathbf{K}$  defined as follows.  $H(\mathcal{R})$  is an  $m \times m$ -matrix with  $\mathcal{R}$  coefficients. Column  $j$  is associated with the simplex  $\sigma_j$  and row  $i$  represents a linear form  $\phi_i : \mathbf{C}_* \rightarrow \mathcal{R}$ . The element  $i, j$  of the matrix is  $H(\mathcal{R})[i, j] = \phi_i(\sigma_j)$ . We call the  $j^{\text{th}}$  column of  $H(\mathcal{R})$  the *annotation of  $\sigma_j$*  and denote it by  $\mathbf{a}_{\sigma_j}$ . We extend linearly the notion of annotation to every chain  $c = \sum a_j \sigma_j$ ,  $a_j \in \mathcal{R}$ , such that  $\mathbf{a}_c = \sum a_j \mathbf{a}_{\sigma_j}$ .

We recall the algorithm to compute persistent cohomology in a single field  $\mathbb{F}$ . Initially,  $H(\mathbb{F}) = 0$ . For all  $j$  from 1 to  $m$ , let  $\mathbf{a}_{\partial\sigma_j}$  be the annotation of the boundary of  $\sigma_j$ , and let  $x = \mathbf{a}_{\partial\sigma_j}[k] \in \mathbb{F}$  be its non-zero element with maximal index  $k$  (if it exists). To each row  $\phi_i$  with  $i < j$ , we apply the Gaussian eliminations:  $\phi_i \leftarrow \phi_i - (x^{-1} \mathbf{a}_{\partial\sigma_j}[i]) \times \phi_k$ . If  $\mathbf{a}_{\partial\sigma_j}$  is null,  $x$  is not defined and we simply set  $\phi_j$  to the cocycle  $\sigma_j^*$ , which verifies  $\sigma_j^*(\sigma_i) = 1$  for  $i = j$  and 0 otherwise, with no further modification on  $H(\mathbb{F})$ . We call *reduction induced by  $\sigma_j$*  these modifications of the matrix  $H(\mathbb{F})$  at iteration  $j$ . The persistent cohomology pairing is deduced directly: if  $\mathbf{a}_{\partial\sigma_j} = 0$  then  $\sigma_j$  creates a cohomology class  $[\phi_j]$ ; if  $\mathbf{a}_{\partial\sigma_j} \neq 0$ ,  $\sigma_j$  destroys the cohomology class  $[\phi_k]$  (created by  $\sigma_k$ ) with  $k$  the maximal index of a non-zero element in  $\mathbf{a}_{\partial\sigma_j}$ , and produces a point  $(f(\sigma_k), f(\sigma_j))$  in the persistence diagram. Every creator  $\sigma_i$  which is not paired at the end of the reduction

produces a point  $(f(\sigma_i), +\infty)$  in the persistence diagram. We refer to [?] for the correction of the algorithm and its description in terms of cohomology.

In MF-persistent homology, we are interested in reducing the matrices  $H(\mathbb{Z}_{q_1}), \dots, H(\mathbb{Z}_{q_r})$  simultaneously. Define  $H^\times$  to be the cohomology matrix with coefficients in the ring  $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$ . We describe a simple reformulation of the separated reductions of the matrices  $H(\mathbb{Z}_{q_1}), \dots, H(\mathbb{Z}_{q_r})$  in terms of the reduction of the matrix  $H^\times$  with  $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$  coefficients, where one can think of  $H^\times$  as the product  $H(\mathbb{Z}_{q_1}) \times \dots \times H(\mathbb{Z}_{q_r})$  (see Appendix ??). Initially  $H^\times = 0$ . At the  $j^{\text{th}}$  iteration of the reduction of  $H^\times$ , we compute  $\mathbf{a}_{\partial\sigma_j} \in (\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r})^m$ , which is equivalent to the product of the annotations of the boundary of  $\sigma_j$  in the matrices  $H(\mathbb{Z}_{q_1}), \dots, H(\mathbb{Z}_{q_r})$ , say  $\mathbf{a}_{\partial\sigma_j}(\mathbb{Z}_{q_1}) \times \dots \times \mathbf{a}_{\partial\sigma_j}(\mathbb{Z}_{q_r})$ . We evaluate  $\mathbf{a}_{\partial\sigma_j}$  bottom-up and perform the reduction associated to  $\sigma_j$  in each matrix  $H(\mathbb{Z}_{q_s})$  separately. Let  $S \subseteq [r]$  be the set of indices  $s$  for which the reduction associated to  $\sigma_j$  has not been done yet in  $H(\mathbb{Z}_{q_s})$ , i.e. the set of indices  $s$  during the evaluation of  $\mathbf{a}_{\partial\sigma_j}$  for which the lowest element of  $\mathbf{a}_{\partial\sigma_j}(\mathbb{Z}_{q_s})$  has not been found yet. Initially  $S = [r]$ . For  $k$  from  $m$  to 1, let  $\mathbf{a}_{\partial\sigma_j}[k] = x = (u_1, \dots, u_r) \in \mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$  and  $T = \{s \in S : u_s \neq 0 \in \mathbb{Z}_{q_s}\}$ . Every  $u_t$  with  $t \in T$  corresponds to the non-zero element of  $\mathbf{a}_{\partial\sigma_j}$  with maximal index in the reduction of  $H(\mathbb{Z}_{q_t})$ . Let  $\tilde{x}^S$  be defined by  $(\tilde{u}_s^S)_{s=1, \dots, r}$  with  $\tilde{u}_s^S = u_s^{-1}$  if  $s \in T$  and 0 otherwise. We proceed to the Gaussian eliminations in  $H^\times$ :  $\forall i < j, \phi_i \leftarrow \phi_i - (\tilde{x}^S \mathbf{a}_{\partial\sigma_j}[i]) \times \phi_k$ . They correspond to the reductions associated to  $\sigma_j$  in each of the matrices  $H(\mathbb{Z}_{q_t})$  for  $t \in T$ . Finally we set  $S$  to  $S \setminus T$  and continue the evaluation of  $\mathbf{a}_{\partial\sigma_j}$ . At the end, if  $S \neq \emptyset$ ,  $\sigma_j$  is a creator in the fields  $\mathbb{Z}_{q_s}$  for  $s \in S$  and we set  $H_{j,j}^\times$  to  $(\delta_{1,S}, \dots, \delta_{r,S})$ , where the symbol  $\delta_{s,S}$  is 1  $\in \mathbb{Z}_{q_s}$  if  $s \in S$  and 0 otherwise.

We develop in the next sections tools for the modular reconstruction algorithm, which consists in manipulating a matrix  $H^*$  with coefficients in  $\mathbb{Z}_Q$  ( $Q = q_1 \dots q_r$ ), in order to simulate the reduction algorithm on  $H^\times$  by using the isomorphism between  $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$  and  $\mathbb{Z}_Q$ .

**Modular Reconstruction for Gaussian Elimination:** For a ring  $\mathbb{Z}_n$ , let  $\mathbb{Z}_n^\times$  be the multiplicative group  $(\mathbb{Z}_n^\times, \times)$  of invertible elements for  $\times$  in  $(\mathbb{Z}_n, +, \times)$ . We present a particular case of the chinese remainder theorem [?] that is sufficient for our study:

**Theorem 3 (Chinese Remainder Theorem)** *For a family  $q_1, \dots, q_r$  of  $r$  distinct prime numbers, there is a ring isomorphism  $\psi : \mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r} \rightarrow \mathbb{Z}_{q_1 \dots q_r}$  s.t. the restriction  $\psi^\times : \mathbb{Z}_{q_1}^\times \times \dots \times \mathbb{Z}_{q_r}^\times \rightarrow \mathbb{Z}_{q_1 \dots q_r}^\times$  is a group isomorphism.*

Let  $[r]$  refer to the set  $\{1, \dots, r\}$ . For a family of  $r$  distinct prime numbers  $\{q_1, \dots, q_r\}$ , and a subset of indices  $S \subseteq [r]$ ,  $Q_S$  refers to  $\prod_{s \in S} q_s$ , and we write simply  $Q = Q_{[r]}$ . We define the function  $\psi_S : \prod_{s \in S} \mathbb{Z}_{q_s} \rightarrow \mathbb{Z}_{Q_S}$  realizing the isomorphism of the Chinese Remainder Theorem for the subset  $\{q_s\}_{s \in S}$  of primes, and we write simply  $\psi$  for  $\psi_{[r]}$ . For a family of elements  $u_s \in \mathbb{Z}_{q_s}, s \in S$ , we denote the corresponding  $|S|$ -uplet  $(u_s)_{s \in S} \in \prod_{s \in S} \mathbb{Z}_{q_s}$ . We give a constructive proof of the chinese remainder theorem. The proof can be found in [?].

**Proof.** For all  $1 \leq s \leq r$ , there exists  $U_s$  such that  $U_s \bmod q_t = 1$  if  $s = t$  and 0 otherwise.

the function  $\psi : \begin{array}{ccc} \mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r} & \rightarrow & \mathbb{Z}_Q \\ (u_1, \dots, u_r) & \mapsto & (u_1 U_1 + \dots + u_r U_r) \bmod Q \end{array}$   
with inverse  $\psi^{-1} : (x \bmod q_1, \dots, x \bmod q_r) \leftarrow x$  realizes the isomorphism.  $\square$

$\square$  In the following, we consider the isomorphism of the former proof when referring to the isomorphism given by the chinese remainder theorem. For any integer  $z \in \mathbb{Z}$  and positive integer  $n > 0$ ,  $z \bmod n$  refers to the equivalence class of  $z$  in  $\mathbb{Z}_n$ . For simplicity, any element  $x \in \mathbb{Z}_n$  is identified with the smallest positive integer belonging to the class  $x$  in  $\mathbb{Z}_n$ . We also denote this integer  $x \in \mathbb{Z}, 0 \leq x < m$ . Consequently, for  $x \in \mathbb{Z}_n$ ,  $x \bmod n'$  refers to the class of  $\mathbb{Z}_{n'}$  to which belongs the integer  $x \in \mathbb{Z}$ , and  $(\bmod n')$  can be seen as a ring homomorphism  $\mathbb{Z}_n \rightarrow \mathbb{Z}_{n'}$ .

Finally, we recall *Bezout's lemma*: for two integers  $a$  and  $b$  there exist integers  $v$  and  $w$  such that  $va + wb = \gcd(a, b)$ , the greatest common divisor of  $a$  and  $b$ .

The Bezout's coefficients  $(v, w)$  can be computed with the extended Euclidean algorithm [?].

*Arithmetic Complexity Model:* During the reduction algorithm we perform arithmetic operations on big integers, for which we describe a complexity model [?]. Suppose that on our architecture, a memory word is encoded on  $w$  bits (on modern architectures,  $w$  is usually 64). Computer chips contains Arithmetic Logic Units that allow arithmetic operations on a 1-memory word integer in  $O(1)$  machine cycles. Let the *length* of an integer  $z$  be defined by:  $\lambda(z) = \lfloor \log_2 z/w \rfloor + 1$ , i.e. by the number of memory words necessary to

encode  $z$ . We express our arithmetic complexities as functions of the length. For any positive integer  $z$  of length  $\lambda(z) = B$ , operations in  $\mathbb{Z}_z$  cost  $A_+(z) = O(B)$  for addition,  $A_\times(z) = O(M(B))$  for multiplication and  $A_{\div}(z) = O(M(B) \log B)$  for (extended) Euclidean algorithm, inversion and division, where  $M(n)$  is a monotonic upper bound on the number of word operations necessary to multiply two integers of length  $B$  [?]. By a result of [?],  $M(B) = O(B \log B 2^{O(\log^* B)})$ .

We are interested in the value of  $\lambda$  for an element in  $\mathbb{Z}_Q$ ,  $Q = q_1 \cdots q_r$ , in the case where  $\{q_1, \dots, q_r\}$  are the first  $r$  prime numbers. We know [?] that  $\ln Q < 1.01624q_r$  and  $q_r < r \ln(r \ln r)$  for  $r \geq 6$ . Consequently,  $\lambda(Q) < \lfloor 1.46613r \ln(r \ln r)/w \rfloor + 1$ . Note that  $\lambda(Q) \ll r$  for  $r \ln r \ll e^w$ , which is a reasonable assumption. In the following, we replace  $r$  distinct arithmetic operations in the fields  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$  (taking  $\Omega(r)$  machine cycles) by one operation in  $\mathbb{Z}_Q$ .

*Gaussian Elimination:* In the following, we are given an integral matrix  $M$  and a family of distinct prime numbers  $\{q_1, \dots, q_r\}$  indexed on  $[r]$ , and their product  $Q$ . Let  $(M_{q_s})_{s=1, \dots, r}$  be the corresponding matrices in the fields  $(\mathbb{Z}/q_s\mathbb{Z})_{s=1, \dots, r}$ . In this section, we introduce algorithms to run Gaussian eliminations simultaneously on all matrices  $(M_{q_s})_{s=1, \dots, r}$ .

Specifically, consider the set of  $r$  matrices  $(M_{q_s})_{s=1, \dots, r}$  as one matrix  $M^\times$  with coefficients in  $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ . Gaussian eliminations are of three types (left) and admit a multi-field counterpart. In the matrix  $M^\times$  with  $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$  coefficients, we define *partial Gaussian eliminations* (right):

- |   |   |       |
|---|---|-------|
| (1) exchange row $k$ and row $\ell$                   | (1') exchange row $k$ and row $\ell$ in $(\mathbb{Z}_{q_s})_{s \in S}$ only                     |       |
| (2) multiply row $k$ by $-1$                          | (2') multiply row $k$ by $-1$ in $(\mathbb{Z}_{q_s})_{s \in S}$ only                            | for a |
| (3) replace row $k$ by (row $k$ ) + $x$ (row $\ell$ ) | (3') replace row $k$ by (row $k$ ) + $x_s$ (row $\ell$ ) in $(\mathbb{Z}_{q_s})_{s \in S}$ only |       |
- set of indices  $S \subseteq [r]$ , and  $x$  and the  $x_s, s \in S$ , elements of the base rings.

Define  $M^*$  as the matrix with coefficients in  $\mathbb{Z}_Q$  obtained by applying the chinese remainder isomorphism to the coefficients of  $M^\times$ . We describe how to simulate the partial Gaussian eliminations on  $M^\times$  by manipulating matrix  $M^*$ .

As the correspondance  $\psi : \mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r} \rightarrow \mathbb{Z}_Q$  is a ring homomorphism, it verifies the properties:  $\psi(u_1, \dots, u_r) + \psi(v_1, \dots, v_r) \times \psi(w_1, \dots, w_r) = \psi(u_1 + v_1 \times w_1, \dots, u_r + v_r \times w_r)$  and we can compute addition and multiplication componentwise in  $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$  using addition and multiplication in  $\mathbb{Z}_Q$ .

In order to compute partial Gaussian eliminations, we first introduce the set of *partial identities*, which are coefficients that allow to proceed to the Gaussian eliminations (1') and (2') on a subset of fields in the cartesian product  $\mathbb{Z}_{q_1} \times \cdots \times \mathbb{Z}_{q_r}$ . Secondly, as the rings  $\mathbb{Z}_{q_s}$  are fields, we need to compute the multiplicative inverse of an element, that is used as multiplicative coefficient  $x$  in Gaussian elimination rule (3). As  $\mathbb{Z}_Q$  is not a field, inversion is not possible, and we introduce the concept of *partial inverse* to overcome this difficulty.

*Partial Identity:* Given a subset of indices  $S \subseteq [r]$ , we define the *partial identities w.r.t. S* by  $L_S = \psi((\delta_{t,S})_{t=1, \dots, r})$  where the symbol  $\delta_{t,S} \in \mathbb{Z}_{q_t}$  is 1 if  $t \in S$  and 0 otherwise. Each  $L_S$  can be constructed in time  $O(rA_+(Q))$  with the explicit formula for  $\psi$ .

We need to access the partial identities during the computation; to avoid complexities depending on  $r$ , we preprocess a static dictionary  $D : Q_S \in \{Q_T : T \subseteq [r]\} \mapsto L_S$  using the data structure of Fredman et al [?].  $D$  has linear size in the number of elements stored  $O(2^r)$  and requires a constant number of arithmetic operations for an access query.

Knowing the partial identities, we can implement the partial Gaussian eliminations (1') and (2') for a set of indices  $S$ . Rule (1') is implemented by replacing row  $k$  by (row  $k \times L_{[r] \setminus S} +$  row  $\ell \times L_S)$  and row  $\ell$  by (row  $\ell \times L_{[r] \setminus S} +$  row  $k \times L_S)$ . Rule (2') is implemented by multiplying row  $k$  by  $L_{[r]} - 2 \times L_S$ .

*Partial Inverse:* We define now the partial inverse of an element in the ring  $\mathbb{Z}_Q$ :

**Definition 4 (Partial Inverse)** *Given a set  $S \subseteq [r]$  of indices, the partial inverse of  $x = \psi(u_1, \dots, u_r)$  with regard to  $S$  is the element  $\bar{x}^S \in \mathbb{Z}_Q$  defined by:*

$$\bar{x}^S = \psi(\bar{u}_1^S, \dots, \bar{u}_r^S), \quad \text{with } \bar{u}_s^S = \begin{cases} u_s^{-1} & \text{if } s \in S \text{ and } u_s \in \mathbb{Z}_{q_s}^\times \\ 0 & \text{o.w.} \end{cases}$$

We present an algorithm to compute the partial inverse of an element  $x \in \mathbb{Z}_Q$ . We prove first that:

**Proposition 5 (Partial Inverse Construction)** *For  $x = \psi(u_1, \dots, u_r) \in \mathbb{Z}_Q$  and  $S \subseteq [r]$ ,*

- (i)  $\gcd(x, Q_S) = Q_R$  with  $R \subseteq S$  and for all  $s \in S$ ,  $u_s$  is invertible in  $\mathbb{Z}_{q_s}$  iff  $s \notin R$ ; we denote  $T = S \setminus R$ .

(ii) The Bezout's identity for  $x$  and  $Q_T$  gives  $vx + wQ_T = 1$  with  $v \bmod Q_T = \psi_T((u_s^{-1})_{s \in T})$

(iii)  $\bar{x}^S = [\psi_T((u_s^{-1})_{s \in T}) \times L_T \bmod Q] \in \mathbb{Z}_Q$ , where  $L_T$  is the partial identity w.r.t  $T$ .

**Proof.** (i): The gcd of  $x$  and  $Q_S$  divides  $Q_S$  so  $\gcd(x, Q_S) = Q_R$  for some  $R \subseteq S$ , and for every index  $s \in S$ ,  $q_s$  divides  $x$  (denoted  $q_s \mid x$ ) iff  $s \in R$ . According to the chinese remainder theorem, for any  $s \in T = S \setminus R$ ,  $u_s = x \bmod q_s \neq 0$  as  $q_s \nmid x$ . As  $\mathbb{Z}_{q_s}$  is a field, its unique non invertible element is 0 and consequently  $u_s$  is invertible. Conversely, as  $q_t \mid x$  for  $t \in R$ ,  $x \bmod q_t = u_t = 0$  is non invertible.

(ii): First note that  $x \bmod Q_T = \psi_T((u_t)_{t \in T}) \in \mathbb{Z}_{Q_T}$ . Indeed, as  $q_t \mid Q_T, \forall t \in T$ , we have  $(x \bmod Q_T) \bmod q_t = x \bmod q_t = u_t$ . By definition of  $T$ ,  $\gcd(x, Q_T) = 1$  and so the Bezout's lemma gives  $vx + wQ_T = 1$ . Applying  $(\cdot \bmod Q_T)$  to both sides of the equality gives  $(v \bmod Q_T)\psi_T((u_t)_{t \in T}) = 1$  and consequently, for every  $q_t, t \in T$ , we have  $((v \bmod Q_T) \bmod q_t)u_t = 1$  and the result follows.

(iii): Let  $L_T$  be the partial identity w.r.t.  $T$ . We form the product  $\tilde{x} = [\psi_T((u_t^{-1})_{t \in T}) \times L_T \bmod Q]$  and evaluate it modulo  $q_s$ . For any index  $s \in [r]$ ,  $\tilde{x} \bmod q_s = [(\psi_T((u_t^{-1})_{t \in T}) \bmod q_s) \times (L_T \bmod q_s)] \bmod q_s$ . If  $s \notin T$ , then  $L_T \bmod q_s = 0$  and  $\tilde{x} \bmod q_s = 0$ . If  $s \in T$ , then  $L_T \bmod q_s = 1$ ,  $\psi_T((u_t^{-1})_{t \in T}) \bmod q_s = u_s^{-1}$  and consequently  $\tilde{x} \bmod q_s = u_s^{-1}$ . Thus,  $\tilde{x}$  verifies the definition of  $\bar{x}^S$ , the partial inverse of  $x$  w.r.t.  $S$ .  $\square \square$  We deduce directly an algorithm to compute the partial inverse of  $x$  from the product  $Q_S$  (see Appendix ??): compute  $Q_R = \gcd(x, Q_S)$  and  $Q_T = Q_S/Q_R$ , then  $v$  using the extended Euclidean algorithm and finally  $\bar{x}^S = (v \bmod Q_T) \times L_T \bmod Q$ .

In the sequel, we apply modular reconstruction to persistent cohomology. It should be noted, however, that modular reconstruction is a general tool which can be used to accelerate matrix reduction algorithms using Gaussian eliminations, like integer matrix reduction to Smith normal form and Hermite normal form [?, ?, ?] using the technique of Hafner and McCurley [?], consisting in reducing the matrix modulo primes to prevent "intermediate expression swell".

**Multi-Field Persistent Cohomology Algorithm:** In the following, we extend the term *isomorphic* to matrices. We say that two  $n \times m$  matrices with coefficients in  $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$  and  $\mathbb{Z}_Q$  respectively are isomorphic iff each elements  $i, j$  are pairwise isomorphic with the isomorphism  $\psi$ .

We now describe how to implement the reduction of  $H^\times$  by means of a "reduction" of  $H^*$ , which verifies that after each iteration  $j$ , from 1 to  $m$ , the matrices  $H^*$  and  $H^\times$  are isomorphic. Note that we do not perform a reduction of the matrix  $H^*$  with ring coefficients, but instead simulate multiple matrix reductions in the fields  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ . The output of the algorithm is the MF-persistence diagram.

Initially,  $H^* = 0$  and the isomorphism property is satisfied. Suppose the isomorphism property is satisfied at iteration  $j - 1$ ; at iteration  $j$ , we compute  $\mathbf{a}_{\partial\sigma_j} \in (\mathbb{Z}_Q)^m$  by performing the alternating sum of the annotations in  $H^*$  of the simplices in the boundary of  $\sigma_j$ :  $\mathbf{a}_{\partial\sigma_j}$  is isomorphic to the annotation of  $\partial\sigma_j$  in the reduction of  $H^\times$ . Let  $Q_S, S \subseteq [r]$ , be the product of prime numbers  $q_s, s \in S$ , for which the reduction induced by  $\sigma_j$  has not been done yet in the  $H(\mathbb{Z}_{q_s})$  "component" of  $H^\times$ . Initially  $Q_S = Q_{[r]}$ . For  $k$  from  $m$  to 1, let  $\mathbf{a}_{\partial\sigma_j}[k] = x = \psi(u_1, \dots, u_r) \in \mathbb{Z}_Q$  and  $\bar{x}^S = \psi(\tilde{x}^S)$ , the partial inverse of  $x$ . We proceed to the Gaussian eliminations on  $H^*$ :  $\forall i < j, \phi_i \leftarrow \phi_i - (\bar{x}^S \mathbf{a}_{\partial\sigma_j}[i]) \times \phi_k$ , which maintains the isomorphism with  $H^\times$  along the reduction. We set  $Q_S$  to  $Q_{S \setminus T} = Q_S/Q_T$  and continue until either  $k < 1$  or  $Q_S = 1$  (or equivalently  $S = \emptyset$ ). Finally, if  $k < 1$  and  $Q_S \neq 1$ , we set  $H_{j,j}^*$  to  $L_S$ . This algorithm maintains the isomorphism property after iteration  $j$ . The output of the algorithm is a set of triplets  $(f(\sigma_i), f(\sigma_j), Q_S)$  encoding the MF-persistence diagram; this representation has size  $O(|D(\mathbf{K}, \mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r})| \times \lambda(Q))$ .

We note that only arithmetic operations in  $\mathbb{Z}_Q$  are performed. In particular, the set  $S$  of indices of "pending" prime numbers is maintained implicitly with the product  $Q_S$  and the difference  $S \setminus T$  is done by a division  $Q_S/Q_T = Q_{S \setminus T}$ . We use the algorithm of Proposition ?? to compute the partial inverse of an element, which requires a constant number of arithmetic operations in  $\mathbb{Z}_Q$ . See Appendix ??.

*Output-Sensitive Complexity Analysis:* The algorithm for computing persistent cohomology admits an efficient sparse implementation [?]. Denote by  $\mathcal{C}_{\mathbf{a}_\sigma}$  and  $\mathcal{C}_G$  the complexity for computing the annotation  $\mathbf{a}_{\partial\sigma}$  of the boundary of a simplex  $\sigma$ , and the complexity for running the Gaussian eliminations associated to a simplex in the sparse cohomology matrix. The complexity of the persistent cohomology algorithm on a filtered complex with  $m$  simplices is  $O(m \times [\mathcal{C}_{\mathbf{a}_\sigma} + \mathcal{C}_G])$ . Note that usually  $\mathcal{C}_{\mathbf{a}_\sigma} \ll \mathcal{C}_G$  as the first depends on the size of a column and the second depends on the size of the matrix. Moreover, typically  $\mathcal{C}_G \ll m$  because the matrix represents the cohomology groups of  $(\mathbf{K}_i)_i$ , which are a compact encoding of the global

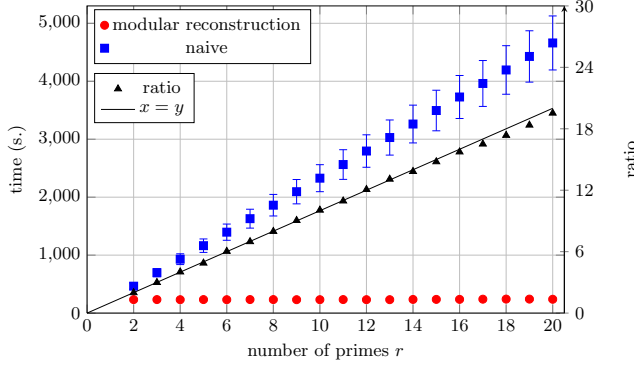


Figure 1: Timings for the modular reconstruction algorithm and the naive approach.

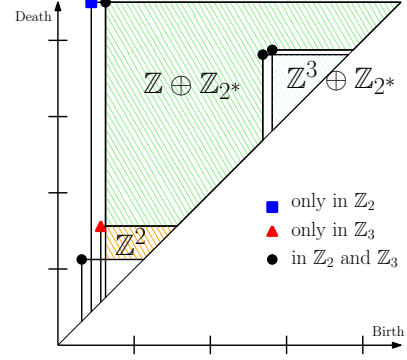


Figure 2: MF-persistence diagram of  $\mathbf{H}_1$  for a Rips complex reconstructing a Klein bottle.

topological features of the shape described by the simplicial complex. The output of the persistence algorithm is the persistence diagram. We express the complexity of the algorithm in terms  $|\mathbf{D}(\mathbf{K}, \mathbb{Z}_q)|$ . Note that  $|\mathbf{D}(\mathbf{K}, \mathbb{Z}_q)| = \Theta(m)$  for any field  $\mathbb{Z}_q$ . The output of the modular reconstruction algorithm is the MF-persistence diagram  $\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$ . As explained earlier,  $|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| \ll r \times |\mathbf{D}(\mathbf{K}, \mathbb{Z}_q)|$ .

**Theorem 6** *Given a filtered complex  $\mathbf{K}$ , the complexity for computing the multi-field persistence diagram  $\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  is:*

$$O(|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| \times [\mathcal{C}_{a_\partial} + \mathcal{C}_G] \times A(Q))$$

*and for the separate diagrams  $(\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_s}))_{s=1, \dots, r}$ :*

$$O(r \times |\mathbf{D}(\mathbf{K}, \mathbb{Z}_q)| \times [\mathcal{C}_{a_\partial} + \mathcal{C}_G] \times A(Q))$$

where  $A(Q)$  is an upper bound for  $A_+(Q)$ ,  $A_\times(Q)$  and  $A_\div(Q)$ .

**Proof.** We analyze the reduction algorithm of  $\mathbf{H}^*$ . For each simplex  $\sigma_j$ , we compute  $\mathbf{a}_{\partial\sigma_j}$  by proceeding to  $O(\mathcal{C}_{a_\partial})$  sums, for a total cost of  $O(m\mathcal{C}_{a_\partial}A_+(Q))$ . The annotation  $\mathbf{a}_{\partial\sigma}$  has length at most  $\mathcal{C}_{a_\partial}$ . We proceed to  $O(\mathcal{C}_{a_\partial})$  calls to the Euclidean algorithm when evaluating  $\mathbf{a}_{\partial\sigma}$  for a total cost of  $O(m\mathcal{C}_{a_\partial}A_\div)$ .

Every reduction of the form  $\forall i < j, \phi_i \leftarrow \phi_i - (\bar{x}^S \mathbf{a}_{\partial\sigma_j}[i]) \times \phi_k$  leads to a different point in the MF-persistence diagram (specifically the point with coordinates  $(f(\sigma_k), f(\sigma_j), Q_T)$ , with  $T$  the set of indices  $t$  for which the pair  $(\sigma_k, \sigma_j)$  is persistent with  $\mathbb{Z}_{q_t}$  coefficients). Consequently, we run  $O(|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})|)$  such reductions. Such operations requires to multiply  $\bar{x}^S$  with  $\mathbf{a}_{\partial\sigma}$  and then run  $O(\mathcal{C}_G)$  additions. The total cost is  $O(|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| \times [\mathcal{C}_{a_\partial}A_\times + \mathcal{C}_GA_+])$ . Summing up, the total complexity for computing the MF-persistent diagram is  $O(m\mathcal{C}_{a_\partial}A_\div + |\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| \times [\mathcal{C}_{a_\partial}A_\times + \mathcal{C}_GA_+])$ , which can be simplified as  $O(|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| \times [\mathcal{C}_{a_\partial} + \mathcal{C}_G] \times A(Q))$ . Observe nevertheless that the more costly arithmetic operations are less used. If the number  $r$  of primes is too big to preprocess the coefficients  $L_S$ , each computation of a partial inverse costs  $O(rA_+(Q))$  and we have to add a term  $O(|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})|rA_+(Q))$  to the complexity.  $\square$

**Experiments:** We report on the practical efficiency of the modular reconstruction algorithm. We use the sparse implementation of persistence cohomology with a compressed annotation matrix [?] and a simplex tree [?], which is one of the most efficient implementation to compute a persistence diagram. We use the GMP library [?] to represent big numbers. All timings are measured on a Linux machine with 3.00 GHz processor and 32 GB RAM, over 10 independent runs.

The input of the algorithm is a filtered simplicial complex, and the output is the MF-persistence diagram of the filtration, for a growing number  $r$  of fields  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ , where  $\{q_1, \dots, q_r\}$  are the first  $r$  prime numbers. We compare in Figure ?? the performance of the modular reconstruction algorithm and the approach consisting in computing persistence separately for each field. The algorithms run on a Rips complex on **Cy8**, a set of points in  $\mathbb{R}^{24}$  sampled from the space of conformations of the cyclo-octane molecule [?] and containing torsion. As expected, the timings for the naive approach follow a linear behavior. The modular reconstruction algorithm is almost insensitive to the number of primes used. The ratio between the timings of the two approaches shows that for  $r \leq 15$ , the modular reconstruction algorithm is precisely  $r$  times faster than the naive approach. For  $r > 15$ , the ratio is slightly under the line  $x = y$ , because numbers in  $\mathbb{Z}_{q_1 \dots q_r}$  do not fit in one memory word, and the arithmetic operations are more costly. However, the gain remains close

to a factor  $r$ : this is due to the fact that most operations are additions and that the compressed annotation matrix algorithm for persistent cohomology reduces the number of arithmetic operations.

## 4 Algorithm for Computing Multi-Field Bottleneck Distance

The definition of bottleneck distance involves the construction of a bijection  $\eta$ , which is computed using a graph matching [?]. We generalize the approach for computing MF-bottleneck distances.

**Multi-Field Matching:** For each triplet  $(x, y, Q_S)$  of an MF-persistence diagram  $D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$ , we denote by  $X^S$  the planar point  $X = (x, y) \in \mathbb{R}^2$  appearing in  $D(\mathbf{K}, \mathbb{Z}_{q_s})$ ,  $s \in S$ . For a filtration  $\mathbf{K}$  with MF-persistence diagram  $D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}) = \{X^S\}$ , we call an element  $X^S$  a *node*. Define the set of nodes  $U_0 = \{X^S\}$  and the corresponding set  $U'_0 = \{X'^S\}$ , where  $X'$  is the projection of  $X$  on the diagonal  $x = y$ . Define similarly  $V_0$  and  $V'_0$  for another filtration  $\widehat{\mathbf{K}}$ . Define the complete bipartite graph  $\mathcal{G} = (U \dot{\cup} V, E)$ , with  $U = U_0 \dot{\cup} V'_0$  and  $V = V_0 \dot{\cup} U'_0$ . Let the length of an edge  $e = (X^S, Y^R) \in E$ , denoted  $\ell(e)$ , be  $+\infty$  if  $S \cap R = \emptyset$ , 0 if both  $X^S \in U'_0$  and  $Y^R \in V'_0$  and  $\|X - Y\|_\infty$  otherwise. An *MF-matching* on  $\mathcal{G}$  is a set  $M$  of elements  $e^T \in E \times 2^{[r]}$ , with an edge  $e = (X^S, Y^R)$  and a set of indices  $T \subseteq [r]$ , verifying: 1.  $T \subseteq S \cap R$ , 2. every two edges  $e^{T_1}$  and  $e^{T_2}$  in  $M$  sharing an endpoint verify  $T_1 \cap T_2 = \emptyset$ , 3. no pair of distinct edges in  $M$  share the same two endpoints. The *size*  $\mathcal{S}(M)$  of an MF-matching  $M$  is defined by  $\sum |T|$  over all edges  $e^T \in M$ . A matching is *maximum* if there is no matching with bigger size. We call the *unmatched indices* of a node  $X^S$  the difference  $S \setminus (\dot{\cup}_{e^T \ni X^S} T)$  and we say  $X^S$  is *matched* if  $\dot{\cup}_{e^T \ni X^S} T = S$ , for the edges  $e^T \in M$ . A matching is *perfect* if all nodes are matched. We suppose for simplicity that, in an MF-matching, every edge  $e = (X^S, Y^R) \in E$  is represented by an element  $e^T$  with  $T$  potentially empty.

The distance  $d_{MF}$  between  $D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  and  $D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})$  is defined to be  $\inf_M \sup_{e^T \in M} \ell(e^T)$  over all MF-matchings  $M$ . We prove:

**Lemma 7**  $d_{MF}(D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}), D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}))$  is equal to the MF-bottleneck distance.

**Proof.** We construct a graph  $\widetilde{\mathcal{G}} = (\widetilde{U} \dot{\cup} \widetilde{V}, \widetilde{E})$  from  $\mathcal{G}$ . Each node  $X^S \in U$  (resp.  $Y^R \in V$ ) of  $\mathcal{G}$  participates to a set of nodes  $\{X^{\{s\}}\}_{s \in S}$  in  $\widetilde{U}$  (resp.  $\{Y^{\{t\}}\}_{t \in R}$  in  $\widetilde{V}$ ). An element  $e^T \in M$  between  $X^S$  and  $Y^R$  participates to a set of edges  $\{(X^{\{t\}}, Y^{\{t\}})\}_{t \in T}$ , with same length, in a matching  $\widetilde{M}$  of  $\widetilde{\mathcal{G}}$  (see Appendix ?? for an example). We observe that  $\widetilde{M}$  is a graph matching iff  $M$  is an MF-matching, and the correspondance maintains the property of being maximum, perfect, maintains the size, and the length of the longest edge. We see that the matching on  $\widetilde{\mathcal{G}}$  realizes the bijection  $\eta$  minimizing the the length of the longest edge in the definition of MF-bottleneck distance.  $\square \square$

**Algorithm for Multi-Field Bottleneck Distance:** We first generalize the maximum matching algorithm of [?] to compute a maximum MF-matching:

**Lemma 8** Given a bipartite graph  $\mathcal{G} = (U \dot{\cup} V, E)$  with nodes  $Z^S$ ,  $S \subseteq [r]$ , we can compute a maximum MF-matching of  $\mathcal{G}$  of size  $\mathcal{S}$  in time  $O(\sqrt{\mathcal{S}} \times |\mathcal{G}| \times A)$

**Proof.** Given a graph  $\mathcal{G}$  and an MF-matching  $M$ , an *augmenting path* is a sequence of nodes  $[Z_1^{S_1}, Z_2^{S_2}, \dots, Z_{2k-1}^{S_{2k-1}}, Z_{2k}^{S_{2k}}]$  with a set of indices  $T \neq \emptyset$ , where  $Z_1^{S_1} \in U$  and  $Z_{2k}^{S_{2k}} \in V$  are not matched by  $M$ , and there is an edge  $e^{T_i} \in M$  between each pair of nodes  $Z_i$  and  $Z_{i+1}$ , for all  $1 \leq i \leq 2k-1$ . Moreover  $T$  must verifies:  $T \subseteq (T_0 \cap T_{2k} \cap_{i=1, \dots, k-1} T_{2i}) \setminus (\bigcup_{i=1, \dots, k} T_{2i-1})$  where  $T_0$  and  $T_{2k}$  are the unmatched indices of  $Z_1$  and  $Z_{2k}$ , respectively. We say that two MF-augmented paths are disjoint if they share no node  $Z_i$  or if their sets of indices are disjoint. Given an augmenting path with edges  $e^{T_1}, \dots, e^{T_{2k-1}}$  and indices  $T$ , rebalancing the set of indices in the following way: for all  $1 \leq i \leq k$ ,  $T_{2i-1} \leftarrow T_{2i-1} \cup T$  and for all  $1 \leq i \leq k-1$ ,  $T_{2i} \leftarrow T_{2i} \setminus T$ , increases the size of the matching by  $|T|$ . Indeed:  $\mathcal{S}(M) = c + \sum_{i=1, \dots, 2k-1} |T_i| = c + \left[ \sum_{i=1, \dots, k-1} (|T_{2i-1}| + |T_{2i}|) \right] + |T_{2k-1}|$  and, by construction,  $T \subseteq T_{2i}$  and  $T \cap T_{2i-1} = \emptyset$ , so the sums  $(|T_{2i-1}| + |T_{2i}|)$  are unchanged by the rebalancing, and  $|T_{2k-1}|$  is increased by  $|T|$ .

The sets of indices  $S$  are encoded by the product of their corresponding prime numbers  $Q_S$ . The inclusion, disjoint union and set difference operations are implemented using gcd computation, product and division respectively, in time  $O(A \cdot (Q))$ . The standard algorithm [?] for finding a maximum matching in a bipartite

graph computes, at each step, a maximal set of vertex disjoint augmenting paths of minimal number of edges. Such a computation is done by traversing the graph, in time linear in its size. If the output maximum matching has size  $\mathcal{S}$ , this step is executed a  $O(\sqrt{\mathcal{S}})$  times. In our multi-field setting, for a graph of size  $|\mathcal{G}|$ , computing the maximal set of disjoint MF-augmenting paths with minimal number of edges consists in traversing the graph, and takes time  $O(|\mathcal{G}| \times \mathbf{A})$ , where  $\mathbf{A}$  is the arithmetic complexity for computing the set operations. If the set of indices is  $[r]$ , we obtain that the number of times the algorithm computes the sets of MF-augmenting paths is  $O(\sqrt{r\mathcal{S}})$  for computing a maximum MF-matching of size  $\mathcal{S}$ . Indeed, the maximum MF-matching computed corresponds to a maximum matching of a graph with at most  $r$  times more nodes (the graph where  $Z^{\mathcal{S}}$  is replaced by the nodes  $\{Z^{\{s\}}\}_{s \in \mathcal{S}}$ ).  $\square \square$  Given two

MF-persistence diagrams computed in the fields  $\mathbb{Z}_{q_1}, \dots, \mathbb{Z}_{q_r}$ , we preprocess the graph so as to only keep the indices  $T \subseteq [r]$  where the diagrams change. They correspond to the prime numbers  $(q_t)_{t \in T}$  dividing the torsion coefficients of the complexes. Let  $t = |T|$ . Let  $\mathcal{G}[\rho]$  be the subgraph of  $\mathcal{G}$  with the edges shorter than  $\rho$  only.

The best algorithm to compute the bottleneck distance between persistence diagrams is the one of [?], which takes time  $O(m^{3/2} \log m)$  for diagrams with less than  $m$  points. It uses a binary search over the values  $\rho$  in order to find the minimal  $\rho_0$  such that the graph admits a perfect matching. Apart from the maximum matching algorithm described earlier, all arguments are based on the geometry of the problem, i.e. the number of distinct geometric points. We can directly adapt the algorithm to our setting:

**Theorem 9** *Given two filtered complexes  $\mathbf{K}$  and  $\widehat{\mathbf{K}}$ , we write  $m'$  for the cumulated size of MF-persistence diagrams  $|\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})| + |\mathbf{D}(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})|$ . There is an algorithm to compute the MF-bottleneck distance  $d_{MF}(\mathbf{D}(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}), \mathbf{D}(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}))$  in time  $O(m'^{3/2} \log m' \sqrt{t\mathbf{A}})$ .*

## References

- [1] Gmp, the gnu multiple precision arithmetic library. <http://gmplib.org/>.
- [2] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. arXiv/1303.0477, 2013.
- [3] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In *ESA*, pages 695–706, 2013.
- [4] Jean-Daniel Boissonnat and Clément Maria. The simplex tree: An efficient data structure for general simplicial complexes. In *ESA*, pages 731–742, 2012.
- [5] Gunnar Carlsson, Tigran Ishkhanov, Vin Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *Int. J. Comput. Vision*, 76:1–12, January 2008.
- [6] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
- [7] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *CoRR*, abs/1107.5665, 2011.
- [8] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45(4):737–759, 2011.
- [9] Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. *CoRR*, abs/1208.5018, 2012.
- [10] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010.
- [11] Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001.
- [12] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with  $O(1)$  worst case access time. *J. ACM*, 31(3):538–544, June 1984.
- [13] Martin Fürer. Faster integer multiplication. *SIAM J. Comput.*, 39(3):979–1005, 2009.
- [14] Joachim Von Zur Gathen and Jurgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
- [15] James L. Hafner and Kevin S. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM J. Comput.*, 20(6):1068–1083, 1991.
- [16] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 1 edition, December 2001.
- [17] John E. Hopcroft and Richard M. Karp. An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.
- [18] Donald E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981.
- [19] S. Martin, A. Thompson, E.A. Coutsias, and J. Watson. Topology of cyclo-octane energy landscape. *J Chem Phys*, 132(23):234115, 2010.
- [20] James R. Munkres. *Elements of algebraic topology*. Addison-Wesley, 1984.
- [21] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. 6:64–94, 1962.



- [22] Arne Storjohann. Near optimal algorithms for computing smith normal forms of integer matrices. In *ISSAC*, pages 267–274, 1996.
- [23] Arne Storjohann and George Labahn. Asymptotically fast computation of Hermite normal forms of integer matrices. In *ISSAC*, pages 259–266, 1996.
- [24] Afra Zomorodian and Gunnar E. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

**Data:**  $\mathbf{K} = [\sigma_1, \dots, \sigma_m]$   
 $H^* \leftarrow 0$ ;  
 $m \times m$ -matrix with  $\mathbb{Z}_Q$  coefficients;  
**for**  $j = 1, \dots, m$  **do**  
     $\mathbf{a}_{\partial\sigma_j}^* \leftarrow$  annotation of the boundary of  $\sigma_j$ ;  
     $Q_S \leftarrow Q_{[r]}$ , product of all primes;  
    **for**  $k = m, \dots, 1$  **do**  
         $x \leftarrow \mathbf{a}_{\partial\sigma_j}^*[k]$ ;  
         $Q_R \leftarrow \gcd(x, Q_S)$ ;  
         $\tilde{x}^S \leftarrow$  partial inverse of  $x$  w.r.t  $S$ ;  
        **foreach**  $i < j$  **do**  
             $\phi_i \leftarrow \phi_i - \tilde{x}^S \mathbf{a}_{\partial\sigma_j}^*[i] \times \phi_k$ ;  
         $Q_S \leftarrow Q_R$ ;  
    **end**  
    **if**  $Q_S > 1$  **then**  $H_{j,j}^* \leftarrow L_S$ ;  
**end**

**Algorithm 1:** MODULAR RECONSTRUCTION

**Data:**  $\mathbf{K} = [\sigma_1, \dots, \sigma_m]$   
 $H^\times \leftarrow 0$ ;  
 $m \times m$ -matrix with  $\mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r}$  coefficients;  
**for**  $j = 1, \dots, m$  **do**  
     $\mathbf{a}_{\partial\sigma_j}^\times \leftarrow$  annotation of the boundary of  $\sigma_j$ ;  
     $S \leftarrow [r]$ , indices of all primes;  
    **for**  $k = m, \dots, 1$  **do**  
         $x \leftarrow \mathbf{a}_{\partial\sigma_j}^\times[k]$ ;  
         $R \leftarrow \{s \in S : \pi_s(x) = 0\}$ ;  
         $\tilde{x}^S \leftarrow$  partial inverse of  $x$  w.r.t  $S$ ;  
        **foreach**  $i < j$  **do**  
             $\phi_i \leftarrow \phi_i - \tilde{x}^S \mathbf{a}_{\partial\sigma_j}^\times[i] \times \phi_k$ ;  
         $S \leftarrow R$ ;  
    **end**  
    **if**  $S \neq \emptyset$  **then**  $H_{j,j}^\times \leftarrow (\delta_{s,S})_{s=1, \dots, r}$ ;  
**end**

**Algorithm 2:** NAIVE

Figure 3: The two approaches for computing the persistent cohomology of a filtered simplicial complex  $\mathbf{K}$  in a family of fields  $(\mathbb{Z}_{q_s})_{s=1, \dots, r}$ .

**Data:**  $x, Q_S$   
 $Q_R \leftarrow \gcd(x, Q_S)$ ;  
 $Q_T \leftarrow Q_S/Q_R$ ;  
 $v \leftarrow \text{EXTENDED-EUCLIDEAN-ALGORITHM}(x, Q_T)$ ;  
 $v \leftarrow v \bmod Q_T$ ;  
 $L_T \leftarrow D(Q_T)$ ;  
 $\tilde{x}^S \leftarrow (v \times L_T) \bmod Q$ ;  
return  $\tilde{x}^S$ ;

**via the euclidean algorithm:**  $O(A_\pm(Q))$ ;  
**such that**  $vx + wQ_T = 1$ :  $O(A_\pm(Q))$ ;  
**using the dictionary:**  $O(A_\pm(Q))$ ;

**Algorithm 3:** PARTIAL-INVERSE

Figure 4: Computation of the partial inverse of  $x$  w.r.t  $S$ .

## A Pseudo Code

For  $s \in \{1, \dots, r\}$ , define  $\pi_s : \mathbb{Z}_{q_1} \times \dots \times \mathbb{Z}_{q_r} \rightarrow \mathbb{Z}_{q_s}$  to be the projection to the  $s^{\text{th}}$  component of the cartesian product. Figure ?? presentes pseudo-code for the modular reconstruction algorithm for computing the multi-field persistence diagram and the naive approach consisting in computing separately the diagrams in each field.

We describe the subroutine to compute the partial inverse of an element:

## B Multi-Field Matching

We give in Figure ?? an explicit example of the correspondance between MF-matchings and graph matchings. The figure pictures a graph  $\mathcal{G}$  with a perfect MF-matching  $M$  and the corresponding graph  $\tilde{\mathcal{G}}$  obtained by splitting the nodes. The graph matching  $\tilde{M}$  obtained in  $\tilde{\mathcal{G}}$  is perfect, and has the same longest edge (labeled by index 2). The edges in the matching of  $\tilde{\mathcal{G}}$  are, by construction, between nodes  $X^{\{t\}}$  and  $Y^{\{t\}}$  with same index  $t$ , and consequently produce a perfect matching as well if we restrict the graph  $\tilde{\mathcal{G}}$  to the nodes  $X^{\{s\}}$  and  $Y^{\{s\}}$  for a given index  $s$ , and the edges of  $\tilde{M}$  to the edges between nodes with index  $s$ . The longest edge of  $\tilde{M}$  is consequently the maximal longest edge in each of the perfect matchings in the graphs restricted to

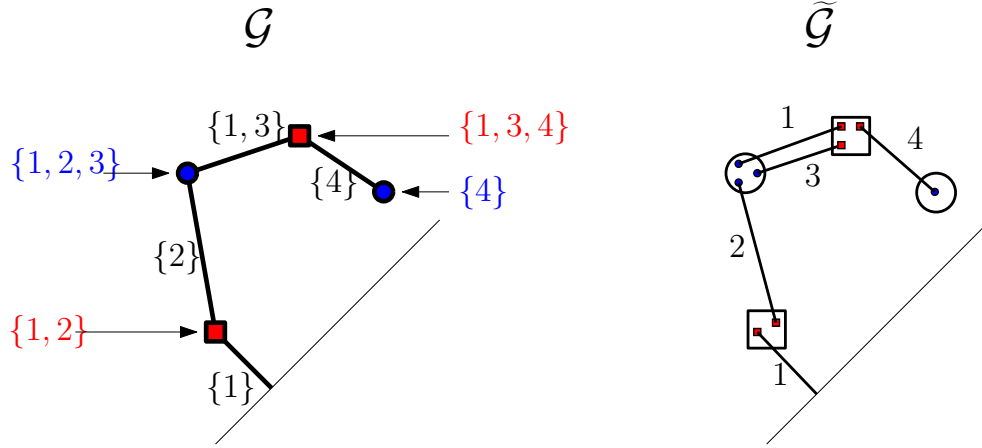


Figure 5: Correspondance between MF-matchings and graph matchings. The two matchings are perfect.

an index  $s$ . Recall the formula for the MF-bottleneck distance between two MF-persistence diagrams:

$$d_{\text{MF}}(D(\mathbf{K}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r}), D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_1} \cdots \mathbb{Z}_{q_r})) = \max_{1 \leq s \leq r} d_{\text{B}}(D(\mathbf{K}, \mathbb{Z}_{q_s}), D(\widehat{\mathbf{K}}, \mathbb{Z}_{q_s}))$$

with  $d_{\text{B}}(D(\mathbf{K}, \mathbb{F}), D(\widehat{\mathbf{K}}, \mathbb{F})) = \inf_{\eta: P \rightarrow \widehat{P}} \sup_{X \in P} \|X - \eta(X)\|_{\infty}$ . The matching restricted to an index  $s$  realizes the bijection  $\eta_s$  in  $\inf_{\eta_s: P^s \rightarrow \widehat{P}^s} \sup_{X^{\{s\}} \in P^s} \|X^{\{s\}} - \eta(X^{\{s\}})\|_{\infty}$ . The length of the longest edge over all these matchings realizes the max.



**RESEARCH CENTRE  
SOPHIA ANTIPOLIS – MÉDITERRANÉE**

2004 route des Lucioles - BP 93  
06902 Sophia Antipolis Cedex

Publisher  
Inria  
Domaine de Voluceau - Rocquencourt  
BP 105 - 78153 Le Chesnay Cedex  
[inria.fr](http://inria.fr)

ISSN 0249-6399