# On the expected complexity of matrix reduction for random complexes[⋆]

Michael Kerber[1] [0000−0002−8030−9299] and Hannah Schreiber[2] [0000−0002−8564−415X]

[1] Graz University of Technology, Graz, Austria
kerber@tugraz.at
[2] St. Louis University, St. Louis MO, USA
hschreiber@slu.edu

**Abstract.** A major computational task in applied topology is the computation of persistence diagrams. This is done by matrix reduction, which simply refers to apply Gaussian elimination to the boundary matrix of a cell complex over a finite field. While the worst-case complexity is known to be cubic, we are posing the question about the expected complexity of matrix reduction when drawing a simplicial complex from a fixed random model uniformly at random. We provide extended experimental evidence that the expected complexities are different for various random models, and we prove that the expected complexity is an order of magnitude smaller than the worst case for one particular model.

**Keywords:** Persistent Homology · Matrix Reduction · Computational Complexity

## 1 Introduction

This topic is motivated by *persistent homology*, an evolving research field of applied topology. It studies general data sets by tracking their topological properties across scales, and summarizes the information in a planar point plot called the *persistence diagram* [2, 5].

*Matrix reduction* forms the computational core of persistent homology. The goal is to apply column addition to the boundary matrix of a cell complex to arrive at an echelon form, out of which the persistence diagram can be read off. Compared to usual Gaussian elimination, there are some minor restrictions which we refer to in Section 2.

The well-known cubical worst-case bit complexity of matrix reduction can also be realized by boundary matrices of simplicial complexes – the initial sparsity of the matrix does not improve the worst-case bound [4]. On the other hand, the empirical performance of matrix reduction to realistic data sets scales much better than what the worst case predicts, as proved by countless applications of persistence to real-world data.

[⋆] Supported by the Austrian Science Fund (FWF) grant number P 29984-N35.

The apparent gap between theory and practice is usually explained by the fact that creating worst-case examples requires a careful design of the input complex with artificial features that simply do not occur in realistic data. It seems natural to restrict the input set to a subset of complexes which share structural properties with data sets of interest. Unfortunately, not much progress has been made so far; for instance, the worst-case complexity of matrix reduction for Vietoris-Rips or Cech complexes [2] over $n$ points is completely open.

*Average-case complexity* can provide further insight of how common "bad" examples are in matrix reduction. Fixing a random process to generate input complexes, we pose the question of what is the expected number of operations performed if an element from the model is chosen uniformly at random. While expected structural properties of simplicial complexes (e.g., its expected Betti number) have been studied [3, 1], the algorithmic direction seems to be unexplored. More generally, we were not able to find any prior work on the (worst-case or expected) complexity of Gaussian elimination for special cases of input classes.

*Our results.* Our first result is an extensive experimental study on the empirical performance of matrix reduction for various standard models of random simplicial complexes, which we briefly describe in Section 3. Fig. 1 shows the number of bit operations in the matrix in dependence of the number of rows and columns. We used linear regression on the log-log-scale to approximate the plots by polynomial functions. Generally, we observe the expected result that more structure in the input model (geometric or combinatorial) yields a better algorithmic performance.

Our second result is a theorem stating that for a particular random model, producing a matrix with $m$ rows and $\binom{m}{3}$ columns, and a slight variant of the standard matrix reduction algorithm, the expected number of bit operations is $O(m^{4+\varepsilon})$, where $\varepsilon > 0$ can be chosen arbitrarily small. This is in contrast to the naive $O(m^5)$ worst-case bound and is the first example (up to our knowledge) where a better complexity bound can be derived for a restricted input set, even though it refers to the average case instead of the worst case.

A more extensive treatment of this work can be found in [6, Ch.4].

## 2 Matrix reduction

For brevity, we restrict our attention to simplicial complexes of dimension 2, although all concepts can be generalized. Let $v$, $e$, $f$ denote the number of 0-, 1-, and 2-dimensional simplices, called vertices, edges, and faces. The *boundary matrix* is an $(e \times f)$-matrix where each row represents an edge and each column represents a triangle, and the entry $(i, j)$ is 1 if edge $i$ is in the boundary of triangle $j$, and 0 otherwise. We consider the matrix coefficients to be elements of $\mathbb{Z}_2$ throughout. We store the columns of the boundary matrix in a sparse representation, that is, as a dynamic array of non-zero row indices.

We call the *pivot* of a non-zero column in a matrix the largest row index that is not zero. The goal is to transform the matrix so that no two columns share

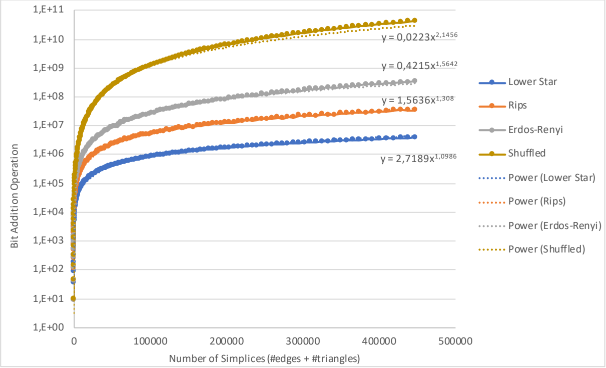

**Fig. 1.** Experimental performance of matrix reduction for various random models.

the same pivot. However, some special rules apply in the context of persistent homology: No swap of two rows or two columns is allowed, and only left-to-right column additions maybe performed. This comes from the fact that in persistent homology, the boundary matrix does not represent only one simplicial complex but rather a *filtration*, that is, a nested sequence of complexes defined by the prefixes of the matrix. This means that the order of simplices is important.

The standard reduction algorithm processes columns from left to right. On each column, it checks if its pivot appears in a previous column $c'$ and adds $c'$ to $c$ in that case. This is repeated until the column vanishes or has a novel pivot. The number of bit operations needed in this process is shown in Fig. 1.

## 3 Random models

We used 4 models in our experiments. In all of them, we fix the number of vertices $v$ and create a complete complex with $e = \binom{v}{2}$ edges and $f = \binom{v}{3}$ triangles. They only differ in the order in which the edges and triangles are added:

- Lower star filtrations: Using a total order on the vertices, every edge can be written as a pair of vertices, every triangle as a triple of vertices. Then, rows and columns are ordered lexicographically.
- Rips filtrations: For each vertex, we pick a random point in the unit square in $\mathbb{R}^2$. The order order is then determined by sorting the edges by increasing length. The triangles are inserted in lexicographic order with respect to the chosen edge order.
- Erdős-Rényi filtrations: A random order is chosen for the edges, and the triangles are inserted in lexicographic order with respect to the edge order.
- Shuffled filtrations: Both edges and triangles are inserted in a random order (chosen independently of each other).

Lower-star filtrations (of sub-complexes) and Rips filtrations (of non-uniform point sets) are standard techniques for applying persistent homology to real-world data; hence these models are not too far from realistic scenarios. The decrease of structure between the orders of edges and triangles leads to an increase in complexity as Fig. 1 shows. In [6, Ch.4], we also measured other properties such as number of column additions, size of the reduced matrix etc.

## 4 A first theoretical result.

We introduce a further random model: Consider a cell complex with one vertex, $m$ edges and $f := \binom{m}{3}$ triangles (note that $f$ is much larger than in the previous models). Order the edges and triangles in random order. Alternatively, the boundary matrix consists of all distinct columns with exactly 3 non-zero entries, randomly shuffled.

Standard reduction applied to this boundary matrix takes $O(m^5)$ time in the worst case. In order to derive an improved expected bound, we introduce a variant of the standard reduction: if we encounter a column $c$ with the same pivot as a previously reduced column $c'$, we swap the contents of $c$ and $c'$ and continue the reduction. The idea is that $c'$ might be re-used for later column additions, so reducing its size will save operations. Although this variant violates the rule of not swapping columns, the resulting reduction yields the same persistence diagram. This leads to the following result (see [6, Thm.4.8] for a proof):

**Theorem 1.** *Let $\epsilon > 0$ be any constant and $m$ be the number of edges. The bit complexity of the modified reduction algorithm is $O(m^{4+\epsilon})$ in expectation.*

## 5 Outlook

We made little progress so far extending our theoretical result to other models. The core problem seems to be the increased level of dependence between the columns during the reduction, making it difficult to bound the expected size of a column while the algorithm progresses. We are searching for techniques to analyze such stochastic processes.

## References

1. Bobrowski, O., Kahle, M.: Topology of random geometric complexes: a survey. Journal of Applied and Computational Topology **1**, 331–364 (2018)
2. Edelsbrunner, H., Harer, J.: Computational Topology. An Introduction. American Mathematical Society (2010)
3. Kahle, M.: Topology of random simplicial complexes: A survey. Algebraic Topology: Applications and New Directions **620** (01 2013)
4. Morozov, D.: Persistence algorithm takes cubic time in worst case. BioGeometry News, Duke University (2005)
5. Oudot, S.: Persistence Theory: From Quiver Representations to Data Analysis. American Mathematical Society (2015)
6. Schreiber, H.: Algorithmic Aspects in standard and non-standard Persistent Homology. Ph.D. thesis, Graz University of Technology (2019), `http://www.geometrie.tugraz.at/kerber/dissertation-schreiber-final.pdf`