COREDUCTION HOMOLOGY ALGORITHM FOR INCLUSIONS AND PERSISTENT HOMOLOGY

MARIAN MROZEK AND THOMAS WANNER

Abstract. We present an algorithm for computing the homology of inclusion maps which is based on the idea of coreductions and leads to significant speed improvements over current algorithms. It is shown that this algorithm can be extended to compute both persistent homology and an extension of the persistence concept to two-sided filtrations. In addition to describing the theoretical background, we present results of numerical experiments, as well as several applications to concrete problems in materials science.

1. Introduction

In the mathematical literature, homology has long been the primary tool for studying topological properties of spaces, and its computability based on the classical Smith normal form algorithm has contributed to its applicability. Only fairly recently have homological methods found their way into the applied sciences, and in this context the Smith normal form algorithm quickly showed serious shortcomings due to its supercubical complexity [31]. For large data sets, performing the algorithm was simply not feasible, both in terms of memory usage and computational time. Not surprisingly, this fact has led to a surge in the development of faster homology algorithms. Some of these consider only specific types of topological spaces such as simplicial or cubical complexes, some only apply to spaces of certain dimensions, and some use probabilistic methods. For more details, we refer the reader to [4, 5, 6, 7, 10, 15, 16, 17, 23, 24], as well as the references therein. Several of the new homology algorithms are based around the idea of pre-processing. Rather than trying to apply the Smith normal form algorithm to the original data set, the idea is to employ a sequence of reduction steps which transforms the original input into one which is considerably smaller and still has the same homology, and then to apply the Smith normal form algorithm as a last step. A variety of different reduction mechanisms have been proposed, such as elementary reductions, acyclic subspace reductions, as well as coreductions. Implementations of these algorithms are available from [40, 41]. Again, we refer the reader to the references cited above for more details.

While the methods mentioned in the previous paragraph have proved to be very successful for the computation of the homology of topological spaces, computing the homology

2000 *Mathematics Subject Classification.* Primary **55-04, 55N35**; Secondary **52B99**.

Key words and phrases. Homology algorithm, cubical set, reduction methods.

The first author was partially supported by Polish MNSzW, Grant N201 037 31/3151 and N N201 419639. The second author was partially supported by NSF grants DMS-0639300 and DMS-0907818, as well as the U.S. Department of Energy under Contract DE-FG02-05ER25712.

of continuous maps between topological spaces — despite its importance — has been addressed to a much lesser extent. A recent approach [22] is based on nerves of coverings and Čech homology. Although the approach is very promising, it has not been implemented so far, because the implementation is not easy. Therefore, there is no evidence how it would perform in practice. An approach which has been implemented successfully [41] is described in [15, 19] and is formulated using the framework of cubical sets. One of the central steps in this approach is the realization that the homology of a map can be obtained by essentially computing the homology of two projections, as long as it is possible to keep track of the images of generators of the involved homology groups. The method has been applied to the computation of the Conley index [21] and to problems in image analysis [25].

Despite these early applications, the efficiency of the homology computation for maps is still lagging behind the efficiency of comparable algorithms for topological spaces. This is partially due to the fact that one has to keep track of the images of homology generators, which makes the adaptation of reduction methods more difficult. In the current paper, we propose an extension of the coreduction method [23] which allows one to compute the homology of inclusion maps efficiently and at the same time keeps track of the homology generators. While this is of independent interest, we also show that our method can be used to efficiently compute persistence for cubical complexes. Persistent homology was introduced in [9, 38], and previously in a zero-dimensional version in the context of shape analysis in [33, 35]. It is aimed at identifying and measuring the significant topological features in an increasing filtration of topological spaces. As one moves through the topological spaces in such a filtration, non-trivial homology generators will be generated and destroyed. Persistent homology keeps track of the birth and death of these generators by furnishing associated persistence intervals. These intervals allow one to study significant topological features which exist over many spaces in the filtration, i.e., over large persistence intervals. We will show that the persistence intervals can be computed efficiently using our coreduction method for inclusion maps. Furthermore, we also address the question of extended persistence, in which two back-to-back filtrations of topological spaces are used to gain insight into the so-called essential topology classes. Our method provides an alternative to the persistence algorithm developed in [39] for simplicial homology. While this algorithm has been adapted to cubical homology by Nanda [26] and Strömbom [32], numerical experiments indicate that our coreduction method provides significant efficiency gains.

The remainder of this paper is organized as follows. In Section 2 we recall the concept of S-complexes introduced in [23], which is fundamental for the justification of our algorithm. We also introduce the concepts of regular subsets of S-complexes and of homology reductions via S-reduction pairs. After that, Section 3 is devoted to the presentation of our coreduction algorithm, as well as a necessary discussion of homology models. Section 4 shows how the homology of inclusion maps can be used to compute persistence intervals, and it also addresses a useful extension of the persistence concept to two-sided filtrations.

Section 5 presents comparisons with other algorithms based on numerical experiments. Finally, Section 6 contains a few selected applications to problems in materials science.

2. S-Complexes and S-Reduction Pairs

In this section we recall and extend basic definitions and results concerning S-complexes. Most of these results are taken from [23]. In Section 2.1 we collect basic theorems on S-complexes, Section 2.2 addresses the central question of when subsets of an S-complex remain an S-complex, and Section 2.3 introduces the basic reduction methods associated with S-complexes. We will also show how one can keep track of homology generators throughout the reduction process.

2.1. **S-Complexes.**

To set the stage, let $\mathcal{C}$ denote a category and let X be an object in $\mathcal{C}$. Then a sequence $(X_q)_{q\in\mathbb{Z}}$ of objects of $\mathcal{C}$ is a *gradation* of X, if X decomposes as the direct sum of the objects X_q. In particular, in the case of the category of sets a gradation is the decomposition into a disjoint union, and in the category of moduli a gradation is the decomposition into the algebraic direct sum.

Now let R denote a ring with unity. Given a finite set A, let $R(A)$ denote the free module over R generated by A. In the following, let S denote a finite set, and denote its cardinality by $|S|$. Furthermore, let $(S_q)_{q\in\mathbb{Z}}$ be a gradation of S such that $S_q = \emptyset$ for all $q < 0$. Then $(R(S_q))_{q\in\mathbb{Z}}$ is a gradation of the module $R(S)$ in the category of moduli over the ring R. For every element $s \in S$ there exists a unique number q such that $s \in S_q$. This number will be referred to as the *dimension* of s and denoted by $\dim s$. We use the notation $\langle\cdot,\cdot\rangle : R(S) \times R(S) \to R$ for the scalar product which is defined on generators by

$$\langle t, s\rangle = \begin{cases} 1 & \text{for } t = s, \\ 0 & \text{otherwise,} \end{cases}$$

and extended bilinearly to $R(S) \times R(S)$.

Let $\kappa : S \times S \to R$ be a map such that

$$\kappa(s,t) \neq 0 \qquad \text{implies} \qquad \dim s = \dim t + 1 \ .$$

We say that the pair (S,κ) is an *S-complex*, if the derived pair $(R(S), \partial^\kappa)$, where the *boundary map* $\partial^\kappa : R(S) \to R(S)$ is defined on generators $s \in S$ by

$$\partial^\kappa(s) := \sum_{t\in S} \kappa(s,t)t \ ,$$

is a free chain complex with base S. The map κ will be referred to as the *coincidence index.* If $\kappa(s,t) \neq 0$, then we say that t is a *face* of s, and s is a *coface* of t. We also define the *coboundary map* δ^κ on generators $t \in S$ by

$$\delta^\kappa(t) := \sum_{s\in S} \kappa(s,t)s \ .$$

Using the notation introduced so far, it is straightforward to verify that for any pair of chains $c \in R_q(S)$ and $d \in R_{q-1}(S)$ we have

$$\langle \partial^\kappa c, d \rangle = \langle c, \delta^\kappa d \rangle \ .$$

To complete our introductory definitions, we define the *homology of an S-complex* (S, κ) as the homology of the associated chain complex $(R(S), \partial^\kappa)$, and denote it by $H(S, \kappa)$ or simply by $H(S)$. In the following, we will drop the superscript κ in ∂^κ whenever κ is clear from context.

Obviously, every basis S of a free chain complex C, with the map κ defined as the matrix of the boundary homomorphism, is an S-complex. Therefore, the concept of an S-complex is basically only a reformulation of the standard concept of a free chain complex. Nevertheless, by referring to an S-complex instead of a chain complex we emphasize the fact that in many applications the natural coding of S intrinsically carries all the information about κ, so there is no need at all to store κ in memory throughout the process of reductions — and the reductions may be performed on the coding of S only. In particular, throughout the paper we assume that the cost of evaluating $\kappa(s, t)$ is constant with respect to $|S|$. The two main examples of such S-complexes are simplicial complexes and cubical complexes, which will be briefly described now.

To begin with, we consider the classical case of simplicial complexes. Recall that a *q-simplex* $\sigma = [A_0, A_1, \ldots A_q]$ in $\mathbb{R}^d$ is the convex hull of $q+1$ affine independent points $A_0, A_1, \ldots A_q$ in $\mathbb{R}^d$, which are called the *vertices* of σ. The number q is then called the *dimension* of the simplex. A *face* of σ is a simplex whose vertices constitute a subset of $(A_0, A_1, \ldots A_q)$. Using this notation, a *simplicial complex* consists of a collection $\mathcal{S}$ of simplices such that every face of a simplex in $\mathcal{S}$ is contained in $\mathcal{S}$ as well, and that the intersection of two simplices in $\mathcal{S}$ is their common face. The simplicial complex $\mathcal{S}$ has a natural gradation $(\mathcal{S}_q)$, where $\mathcal{S}_q$ consists of all simplices of dimension q. Since a zero dimensional simplex is the singleton of its unique vertex, the set $\mathcal{S}_0$ may be identified with the collection of all vertices of all simplices in the simplicial complex $\mathcal{S}$. Finally, assume that an ordering of $\mathcal{S}_0$ is given, and that every simplex σ in $\mathcal{S}$ is coded as $[A_0, A_1, \ldots A_q]$, where the vertices $A_0, A_1, \ldots A_q$ are listed according to the prescribed ordering of $\mathcal{S}_0$. If we then define

$$\kappa(\sigma, \tau) := \begin{cases} (-1)^i & \text{if } \sigma = [A_0, A_1, \ldots A_q] \\ & \qquad \text{and } \tau = [A_0, A_1, \ldots A_{i-1}, A_{i+1}, \ldots A_q] \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

we obtain an S-complex whose chain complex is the classical simplicial chain complex used in simplicial homology.

Our second example is concerned with cubical complexes. For this, let $I = [k, l] \subset \mathbb{R}$ denote a compact interval, define its *length* as $l - k$, and denote the length by $\operatorname{length} I$. The interval I is called an *elementary interval*, if $\operatorname{length} I \in \{0, 1\}$ and both of its endpoints are integers. Elementary intervals of length one are called *nondegenerate*. We define the *left interval* of I by $I^- := [k, k]$, and the *right interval* of I by $I^+ := [l, l]$. Then an *elementary cube* in $\mathbb{R}^d$ is the Cartesian product $Q = I_1 \times \cdots \times I_d$ of d elementary intervals, and the *dimension* of Q is the number of nondegenerate intervals in the product decomposition. A *full elementary cube* is an elementary cube whose product decomposition consists only of nondegenerate intervals. For every elementary cube Q and every number $j \in \{1, 2, \dots d\}$ we define the j-th *nondegeneracy number* of Q by

$$\nu(Q,j) := \begin{cases} \operatorname{card}\{\, i < j \mid \operatorname{length} I_i = 1 \,\} & \text{if } \operatorname{length} I_j = 1 \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Using the above notation, a *cubical complex* $\mathcal{C}$ in $\mathbb{R}^d$ is a finite collection of elementary cubes in $\mathbb{R}^d$, and the associated *cubical set* is the union of this collection. The cubical complex $\mathcal{C}$ has a natural gradation $(\mathcal{C}_q)_{q\in\mathbb{Z}}$, where $\mathcal{C}_q$ consists of elementary cubes of dimension q. Finally, if we define

$$\kappa(Q,P) := \begin{cases} (-1)^{\nu(Q,j)} & \text{if } Q = I_1 \times \cdots \times I_j \times \cdots \times I_d \\ & \qquad \text{and } P = I_1 \times \cdots \times I_j^- \times \cdots \times I_d \ , \\ (-1)^{1+\nu(Q,j)} & \text{if } Q = I_1 \times \cdots \times I_j \times \cdots \times I_d \\ & \qquad \text{and } P = I_1 \times \cdots \times I_j^+ \times \cdots \times I_d \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

then a straightforward induction argument with respect to d may be used to show that a cubical complex is indeed an S-complex. The associated chain complex coincides with the cubical chain complex introduced in [15] for the definition of the homology of cubical sets, i.e., for finite unions of elementary cubes.

In both of the above examples the internal structure of the generators carries all the information needed to compute the coincidence index of two generators in constant time. Therefore, such complexes may be represented in memory by storing only the generators — there is no need to store the coincidence index. In the case of a cubical complex, the memory representation may be taken to be a bitmap, which is particularly advantageous.

2.2. **Regular Subsets of S-Complexes.**

In order to simplify a given S-complex through a reduction step, one has to replace the original set of generators S by a subset $S' \subset S$, and the original coincidence index κ by the restriction $\kappa' = \kappa|_{S'\times S'}$. This has to be done in such a way that (S', κ') is still an S-complex, and that $H(S) \cong H(S')$. Characterizations for both of these requirements were obtained in [23], and we briefly review these results. For this, we need the following notation. For any subset $A \subset S$ define

$$\begin{aligned} \operatorname{bd}_S A &:= \{\, t \in S \mid \ \kappa(s,t) \neq 0 \text{ for some } s \in A \,\} \ , \\ \operatorname{cbd}_S A &:= \{\, s \in S \mid \ \kappa(s,t) \neq 0 \text{ for some } t \in A \,\} \ . \end{aligned}$$

If the set A contains only one element s, then we simply write $\operatorname{bd}_S s$ or $\operatorname{cbd}_S s$. In addition, a subset $S' \subset S$ is called *closed* in S if we have $\operatorname{bd}_S S' \subset S'$, and it is called *open* in S if $S \setminus S'$ is closed in S. Using these definitions, one can now characterize when a subset of S gives rise to a new S-complex. The following results can be found in [23].

Theorem 2.1. *Let (S, κ) denote an S-complex over the ring R, and let $S' \subset S$. Then the following hold:*

(a) *If $S' \subset S$ is chosen in such a way that for all $s, u \in S'$ and $t \in S$ we have*

$$t \in bd_S s \quad \text{and} \quad u \in bd_S t \qquad \text{imply} \qquad t \in S' ,$$

then the pair (S', κ') with $\kappa' = \kappa|_{S' \times S'}$ is again an S-complex. Any subset $S' \subset S$ which satisfies the above implication is called regular*.*

(b) *If $S' \subset S$ is closed in S, then both S' and $S \setminus S'$ are regular.*

(c) *If S' is closed in S, then the boundary map $\partial^{\kappa'}$ satisfies $\partial^{\kappa'} = \partial^{\kappa}|_{R(S')}$. Furthermore, $R(S')$ is a subcomplex of $R(S)$.*

Notice that Theorem 2.1(c) does usually not hold if S' is not closed in S, since the sum defining $\partial^\kappa(s)$ for some $s \in S'$ might contain terms in $S \setminus S'$. Therefore, only for closed subsets S' of an S-complex S it is possible to define the relative homology group $H(R(S), R(S'))$. Nevertheless, even if S' is not closed, it is still possible that S' is regular, i.e., the pair (S', κ') with $\kappa' = \kappa|_{S' \times S'}$ is again an S-complex. In order to simplify notation we will write $H(S, S')$ instead of $H(R(S), R(S'))$ in the following.

The next theorem, which is also taken from [23], shows that the assumption of closedness implies the existence of an exact sequence of homologies.

Theorem 2.2. *Let (S, κ) denote an S-complex over the ring R, and let $S' \subset S$ be a closed subset. Then the following hold:*

(a) *Let $S'' = S \setminus S'$ and $\kappa'' = \kappa|_{S'' \times S''}$. Then both the inclusion ι and the projection π given by*

$$\iota \;:\; \left(R(S'), \partial^{\kappa'}\right) \to (R(S), \partial^\kappa) \quad \text{and} \quad \pi \;:\; (R(S), \partial^\kappa) \to \left(R(S''), \partial^{\kappa''}\right)$$

are chain maps. Moreover, we have the short exact sequence given by

$$0 \to R(S') \xrightarrow{\iota} R(S) \xrightarrow{\pi} R(S'') \to 0 ,$$

and the long exact sequence of homology modules given by

$$\ldots \xrightarrow{\partial_{q+1}} H_q(S') \xrightarrow{\iota_q} H_q(S) \xrightarrow{\pi_q} H_q(S'') \xrightarrow{\partial_q} H_{q-1}(S') \xrightarrow{\iota_{q-1}} \ldots$$

(b) *The homology of the chain complex $(R(S''), \partial^{\kappa''})$ satisfies*

$$H(S'') \cong H(S, S') .$$

Motivated by the above theorem, we call a regular subset $T \subset S$ a *nullset* of S, provided that T is closed or open in S and that $H(T) = 0$. The following result follows immediately from Theorem 2.2.

Corollary 2.3. *Let (S, κ) denote an S-complex over the ring R, and let $T \subset S$ be a nullset of S. Then the homologies $H(S)$ and $H(S \setminus T)$ are isomorphic.*

The corollary tells us that if one is able to locate a nullset in S, then it can be removed without changing the homology of S. In the next section we will indicate a simple method of locating nullsets in S. We conclude the current section with the following simple lemma, which will be useful in the sequel.

Lemma 2.4. *Let (S,κ) denote an S-complex over the ring R, and assume that three elements $s_0, t_0, u_0 \in S$ are chosen in such a way that $u_0 \in \operatorname{bd} t_0$ and $t_0 \in \operatorname{bd} s_0$. Then we have both* $\operatorname{card} \operatorname{bd} s_0 \geq 2$ *and* $\operatorname{card} \operatorname{cbd} u_0 \geq 2$.

As with all the other results of this section, the proof of Lemma 2.4 can be found in [23].

2.3. **S-Reduction Pairs.** In the current section we describe a simple method for locating nullsets in an S-complex. This method is based on the concept of a reduction pair of a finitely generated free chain complex C, which was introduced in [15, Section 4.4] and [16]. Given a free chain complex C with basis S we say that a pair (a,b) of elements of S is called *reduction pair* in C, if $\langle \partial b, a\rangle$ is invertible in the ring R. Notice that any reduction pair (a,b) satisfies $\dim b = 1 + \dim a$. For every such reduction pair one can define an associated chain complex $(\bar{C}, \bar{\partial})$ via

$$\bar{C}_q = \begin{cases} C_q & \text{for } q \notin \{\dim a, \dim b\}\ , \\ \{v \in C_q \mid \langle v, a\rangle = 0\} & \text{for } q = \dim a\ , \\ \{v \in C_q \mid \langle v, b\rangle = 0\} & \text{for } q = \dim b\ , \end{cases} \tag{1}$$

as well as

$$\bar{\partial}_q v = \begin{cases} \partial_q v & \text{for } q-1 \notin \{\dim a, \dim b\}\ , \\ \partial_q v - \dfrac{\langle \partial_q v, a\rangle}{\langle \partial_q b, a\rangle} \partial_q b & \text{for } q-1 = \dim a\ , \\ \partial_q v - \langle \partial_q v, b\rangle b & \text{for } q-1 = \dim b\ . \end{cases} \tag{2}$$

In other words, one has to remove all chains from C which have nontrivial coincidence with a or b and then update the boundary operator accordingly. One can show that the chain complexes (C,∂) and $(\bar{C},\bar{\partial})$ are chain equivalent [15, 16], and we call the chain complex $(\bar{C},\bar{\partial})$ a *reduction* of (C,∂) through the reduction pair (a,b).

As mentioned in the introduction, the fundamental philosophy behind reduction methods is the fact that it is often advantageous to perform a sequence of reductions before applying the Smith normal form algorithm. In fact, experiments show that frequently the decrease in size of the chain complex is significant [24, 23]. From a computational point of view it is desirable to be able to identify reduction pairs quickly and then to be able to perform the reduction efficiently. This naturally leads to considering only specific types of reduction pairs. In this paper, we concentrate on the notions of *elementary reduction pairs* and *elementary coreduction pairs*, which were introduced in [23].

Definition 2.5. *Let (S,κ) denote an S-complex. Then a reduction pair (a,b) of elements of S is called an* elementary reduction pair *if we have $cbd_S a = \{b\}$, and in this case a is called* free face *in S. Furthermore, the reduction pair (a,b) is called* elementary coreduction pair *if we have $bd_S b = \{a\}$, and in this case b is called a* free coface *in S. Finally, we will use the term* S-reduction pair *to denote either an elementary reduction pair or an elementary coreduction pair.*

It turns out that S-reduction pairs give rise to nullsets in an S-complex, and we can therefore use them as the basis of a reduction algorithm for S-complexes which preserves homology. The following result is established in [23].

Theorem 2.6. *Let (S, κ) denote an S-complex over the ring R, and let $a, b \in S$. Then the following hold:*

(a) *If (a, b) is an elementary reduction pair, then $\{a, b\}$ is open in S and a nullset.*
(b) *If (a, b) is an elementary coreduction pair, then $\{a, b\}$ is closed in S and a nullset.*

Together with Corollary 2.3 from the previous section, we can now immediately deduce the following central result.

Corollary 2.7. *Let (S, κ) denote an S-complex over the ring R, and let $a, b \in S$. If (a, b) is an S-reduction pair, then the homologies $H(S)$ and $H(S \setminus \{a, b\})$ are isomorphic.*

Corollary 2.7 lies at the heart of the coreduction homology algorithm presented in [23]. To make it useful one needs to find as many S-reduction pairs as feasible. The reader may have noticed that in the case of simplicial complexes and cubical complexes it is straightforward to provide examples which admit elementary reduction pairs, but elementary coreduction pairs are not possible. However, it is easy to observe that by removing a vertex one obtains an open subcomplex which admits elementary coreduction pairs. Moreover, the homology of this subcomplex coincides with the reduced homology of the original complex. Therefore, not only elementary reduction pairs, but also elementary coreduction pairs are useful when computing the homology of simplicial or cubical complexes. In fact, numerical experiments show that elementary coreduction pairs provide essentially deeper reductions. For details we refer the reader to [23, Section 5].

Since in the current paper we are interested in computing both generators and homology maps, we need to obtain explicit formulas for the isomorphism established in Corollary 2.7. For this, let (a, b) denote a reduction pair. For the chain complex $(\bar{C}, \bar{\partial})$ the relevant chain maps are given by

$$\psi_k^{(a,b)}(c) := \begin{cases} c - \dfrac{\langle c, a\rangle}{\langle \partial b, a\rangle}\partial b & \text{for } k = \dim b - 1 \ , \\ c - \langle c, b\rangle b & \text{for } k = \dim b \ , \\ c & \text{otherwise} \ , \end{cases} \tag{3}$$

and

$$\iota_k^{(a,b)}(c) := \begin{cases} c - \dfrac{\langle \partial c, a\rangle}{\langle \partial b, a\rangle} b & \text{for } k = \dim b \ , \\ c & \text{otherwise} \ , \end{cases} \tag{4}$$

and we have the following result.

Theorem 2.8. *Let (S, κ) denote an S-complex over the ring R, let $a, b \in S$ denote an arbitrary S-reduction pair, and consider the complex $\bar{C}$ defined in (1). Then the chain maps $\psi^{(a,b)} : C \to \bar{C}$ and $\iota^{(a,b)} : \bar{C} \to C$ defined in (3) and (4), respectively, are mutually inverse chain equivalences.*

Proof: The chain homomorphisms $\iota^{(a,b)}$ and $\psi^{(a,b)}$ coincide with the chain homomorphisms p and j defined in [16]. Therefore, the above theorem follows from the argument used in the proof of [16, Theorem 2]. Compare also [15, Theorem 4.24, Proposition 4.25]. □

It turns out that in the case of an S-reduction pair (a, b), we can use the maps $\psi^{(a,b)}$ and $\iota^{(a,b)}$ also in the context of the chain complex (C', ∂'), which is obtained from the regular subset $S' := S \setminus \{a, b\}$ and is explicitly given by $C' := R(S')$, as well as $\partial' := \partial^\kappa|_{C'}$. This is a consequence of the following result.

Theorem 2.9. *Let (S, κ) denote an S-complex over the ring R, let $a, b \in S$ denote an arbitrary S-reduction pair, and consider the complexes $\bar{C}$ and C' defined above. Then the chain complexes $(\bar{C}, \bar{\partial})$ and (C', ∂') coincide.*

Proof: Since C' is generated by $S' = S \setminus \{a, b\}$, it is immediate that $\bar{C}$ and C' are identical. To see that also the boundary maps $\bar{\partial}$ and ∂' coincide, first observe that for any $s \in S'$ we have

$$\partial^\kappa s = \partial^{\kappa'} s + \kappa(s, b)b + \kappa(s, a)a \; .$$

Therefore, we always have $\partial^\kappa s = \partial^{\kappa'} s$ as long as $\dim s \notin \{m, m+1\}$, where $m = \dim b$. Now assume that $\dim s = m$. In the case of an elementary reduction pair (a, b) we obtain for $s \in S'$ the identities

$$\bar{\partial} s = \partial^\kappa s - \frac{\langle \partial s, a\rangle}{\langle \partial b, a\rangle} \partial b = \partial^\kappa s = \partial^{\kappa'} s + \kappa(s, a)a = \partial^{\kappa'} s \; ,$$

since $\kappa(s, a) = 0$ for $s \neq b$ in this situation. On the other hand, in the case of an elementary coreduction pair (a, b) we obtain for $s \in S'$ the identities

$$\bar{\partial} s = \partial^\kappa s - \frac{\langle \partial s, a\rangle}{\langle \partial b, a\rangle} \partial b = \partial^\kappa s - \kappa(s, a)a = \partial^{\kappa'} s \; ,$$

since we have $\partial b = \kappa(b, a)a$ in this case. Finally, assume that we have $\dim s = m + 1$. Then one can readily show that

$$\bar{\partial} s = \partial^\kappa s - \langle \partial s, b\rangle b = \partial^{\kappa'} s + \kappa(s, b)b - \langle \partial s, b\rangle b = \partial^{\kappa'} s \; ,$$

and this completes the proof of the theorem. □

The basic usage of reduction pairs can be summarized in the following definition. For this definition, we do not specifically require the reduction pairs to be either elementary reduction or elementary coreduction pairs.

Definition 2.10. *A* reduction sequence *of a chain complex C is a sequence of generator pairs $\omega = \{(a_i, b_i)\}_{i=1,2,\dots n}$ in S such that (a_i, b_i) is a reduction pair in C^{i-1}, where the chain complexes (C^i, ∂^i) are defined recursively by letting $(C^0, \partial^0) = (C, \partial)$, and then letting (C^i, ∂^i) denote the reduction of $(C^{i-1}, \partial^{i-1})$ through (a_i, b_i), for $i = 1, 2, \dots n$. We then use the notation $(C^\omega, \partial^\omega)$ for the last chain complex in the sequence $\{(C^i, \partial^i)\}_{i=1,2,\dots n}$*

and call this chain complex an ω-reduction *of* (C, ∂)*. Given a reduction sequence* ω *in* S *we also let*

$$\begin{aligned} \iota^\omega &= \iota^{(a_1,b_1)} \circ \iota^{(a_2,b_2)} \circ \cdots \circ \iota^{(a_n,b_n)} , \\ \psi^\omega &= \psi^{(a_n,b_n)} \circ \psi^{(a_{n-1},b_{n-1})} \circ \cdots \circ \psi^{(a_1,b_1)} . \end{aligned}$$

Before closing this section, we would like to reiterate the main reason for introducing the concept of S-complexes and S-reduction pairs. At first glance it seems too restrictive to only consider these special cases of general reduction pairs. However, from an algorithmic point of view one has to be able to both quickly identify reduction pairs, and then to efficiently perform the reduction step. As we will see in Theorem 3.1 below, this is the main reason for considering only S-reduction pairs in practice. As a side benefit, the chain maps ι^ω and ψ^ω described above allow one easily to track generators as well.

3. Coreductions and the Homology of Inclusion Maps

After the preparations of the last section, we now turn our attention to the coreduction algorithm for inclusion maps. First recall that finitely generated abelian groups admit bases, i.e., minimal sets of generators which behave similarly to bases in finite-dimensional vector spaces [15, Theorem 3.61]. In particular, any two such minimal set of generators have the same cardinality and given a basis, every element of the group may be written as linear combination of basis elements with integer coefficients. Therefore, the homomorphisms of such groups may be studied in terms of integer matrices.

Let S denote an S-complex and let $T \subset S$ be an arbitrary closed subset. Then there is a chain map $\rho : T \to S$ which is induced by the inclusion map. The construction of the matrix of the map induced in homology by ρ is a two-step procedure.

Step 1: Construct bases $\{ [v_1], [v_2], \ldots [v_m] \}$ and $\{ [u_1], [u_2], \ldots [u_n] \}$ of the homology groups $H_q(T)$ and $H_q(S)$, respectively.

Step 2: Decompose each generator $[v_i]$ of $H_q(T)$ with respect to the set of generators given by $\{ [u_1], [u_2], \ldots [u_n] \}$ in $H_q(S)$.

Notice that both, constructing a basis as well as decomposing a homology class with respect to a given basis, are procedures of independent interest with applications going beyond computing homology of maps. From the algebraic point of view both procedures may easily be implemented by means of the Smith diagonalization but such an implementation has complexity $O(n^\alpha)$ with $\alpha = 3.376...$; see [31].

The goal of this section is to show that the reduction methods discussed in the last section may substantially speed up each of the two procedures and consequently also the construction of the homology of an inclusion map. To achieve this we first discuss in Section 3.1 and 3.2 the transporting of chains via the maps ι^ω and ψ^ω, then introduce the concept of homology model in Section 3.3, and finally apply these ideas to the homology of inclusion maps in Section 3.4.

3.1. **Transporting Chains.** To see that it is often possible to significantly speed up the process of constructing a homology basis, assume that some ω-reduction C^ω of the

S-complex C is significantly smaller. Then, constructing the generators in the reduced complex C^ω may be achieved much quicker. Now, applying the chain equivalence ι^ω to these generators one obtains the generators of the original S-complex. Notice, however, that this only makes sense if the involved generator transport can be implemented efficiently.

A slightly more complicated but essentially similar situation is the case of decomposing a homology class. Let C denote a chain complex and let $u_1, u_2, \dots u_n$ be cycles in C such that $\{\, [u_1], [u_2], \dots [u_n] \,\}$ is a set of generators of the homology group $H_q(C)$. Furthermore, let $[z] \in H_q(C)$ be an arbitrary homology class. The decomposition of the equivalence class $[z]$ in terms of the given generators consists in finding integral coefficients x_i such that

$$[z] = \sum_{i=1}^{n} x_i [u_i] \ ,$$

and this problem reduces to solving the equation

$$z = \sum_{i=1}^{n} x_i u_i + \partial c \tag{5}$$

for some unknown coefficients $x_1, x_2, \dots, x_n \in \mathbb{Z}$ and an unknown chain $c \in C_{q+1}(C)$. Now consider the case of an S-complex C and assume that we have $S_q = \{\, s^q_j \mid j = 1, 2, \dots, r_q \,\}$. Then the chains z, u_i, and c can all be represented as linear combinations in the form

$$z = \sum_{j=1}^{r_q} z_j s^q_j \ , \qquad u_i = \sum_{j=1}^{r_q} u_{ij} s^q_j \ , \qquad c = \sum_{k=1}^{r_{q+1}} y_k s^{q+1}_k \ .$$

Due to

$$\partial s^{q+1}_k = \sum_{j=1}^{r_q} a_{kj} s^q_j \ ,$$

where we use the abbreviation $a_{kj} = \kappa(s^{q+1}_k, s^q_j)$, one obtains

$$\partial c = \sum_{j=1}^{r_q} \left(\sum_{k=1}^{r_{q+1}} a_{kj} y_k \right) s^q_j \ ,$$

which implies that (5) can be rewritten as

$$z_j = \sum_{i=1}^{n} u_{ij} x_i + \sum_{k=1}^{r_{q+1}} a_{kj} y_k \qquad \text{for} \qquad j = 1, 2, \dots r_q \ . \tag{6}$$

This reformulation is a system of r_q equations in $n + r_{q+1}$ unknowns. We would like to point out that the worst case complexity of solving such a linear system is supercubical, and that in general it is difficult to solve the system significantly quicker. This poses a serious obstacle when the numbers r_q are large.

Again, if some ω-reduction C^ω of the S-complex C is significantly smaller, then it turns out to be advantageous to transport the generators u_i and the chain z to the reduced complex C^ω via the map ψ^ω, which was introduced at the end of the last section. Once this has been done, one can solve the equation (6) in the reduced complex C^ω, and then finally transport the solution back to the original complex C via the map ι^ω. As in the case of constructing homology generators, this only makes sense if the involved generator transport can be implemented efficiently.

3.2. **The Cost of Transporting Chains.** In order to assess the actual cost of the generator transport, define the *length of a chain* $c \in R(S)$ by

$$l_S(c) := |\{ s \in S \mid \langle c, s \rangle \neq 0 \}| ,$$

the *weight of a generator* $s \in S$ by

$$w_S(s) := \max(l_S(\partial s), l_S(\delta s)) ,$$

and finally the *weight of the set* S by

$$w(S) := \max \{ w_S(s) \mid s \in S \} .$$

In order to keep the notation as simple as possible, we will drop the index S in the definitions of l_S and w_S in the following, as long as the set S is clear from the context. With these notations, we can now assess the cost of the generator transport.

Theorem 3.1. *Consider the situation of Definition 2.10. Then the cost of computing the image of a chain under the maps ψ^ω or ι^ω is bounded by $O(|S|^2)$. If however, in addition, all reduction pairs are actually S-reduction pairs, then the cost of computing the image of a chain under the maps ψ^ω or ι^ω is bounded by $O(|S| \cdot w(S))$.*

Proof: We assume that chains are represented as vectors of their expansion coefficients. Under such a representation the cost of modifying a chain c by adding or subtracting another chain d is bounded by $O(l(d))$, the cost of evaluating $\langle c, s \rangle$ for a generator $s \in S$ is constant, and the cost of evaluating $\langle \partial c, a \rangle = \langle c, \delta a \rangle$ is bounded by $O(w(a))$. It then follows from (3) and (4) that the cost of finding the image of a chain c under either the map $\psi^{(a,b)}$ or the map $\iota^{(a,b)}$ is bounded by $O(\max(w(a), w(b)))$. Therefore, the cost of computing the image of a chain c under ψ^ω or ι^ω is bounded by

$$O(\max(w_1(a), w_1(b)) + \max(w_2(a), w_2(b)) + \cdots + \max(w_n(a), w_n(b))) ,$$

where $w_i(s)$ denotes the weight of s in the chain complex C_i. In the case of general reduction pairs, we then obtain the estimate

$$\max(w_i(a), w_i(b)) \leq |S| ,$$

which immediately furnishes the bound $O(|S|^2)$ for the total cost, due to $n \leq |S|$. On the other hand, if all involved reduction pairs are actually S-reduction pairs, one can readily see that

$$\max(w_i(a), w_i(b)) \leq \max(w_S(a), w_S(b)) \leq w(S) ,$$

and therefore in the case of S-reduction pairs, one obtains the upper bound $O(|S| w(S))$ as an estimate of the total cost. □

```
function HomologyModel(S-complex $S$)
begin
    $Q$ :=empty queue of generators;
    $M$ :=empty list of generator pairs;
    $s$ :=a generator in $S_0$;
    enqueue($Q$,$s$);
    while $Q \neq \emptyset$ do begin
        $s$ := dequeue($Q$);
        if $\mathrm{bd}_S\, s$ contains exactly one element $t$ then begin
            $S := S \setminus \{s\}$;
            foreach $u \in \mathrm{cbd}_S\, t$ do
                if $u \notin Q$ then enqueue($Q, u$);
            $S := S \setminus \{t\}$;
            enlist($M, (s, t)$);
        end
        else if $\mathrm{bd}_S\, s = \emptyset$ then
            foreach $u \in \mathrm{cbd}_S\, s$ do
                if $u \notin Q$ then enqueue($Q, u$);
    end;
    return $(M, S)$;
end;
```

Table 1. Coreduction homology model algorithm.

3.3. **Homology Models.** Based on the above discussion, we can now present the definition which lies at the center of the methods discussed in this section.

Definition 3.2. *Let (S, κ) denote an S-complex over a ring R. Then a* coreduction homology model *for S is a reduction sequence $M := \{(a_i, b_i)\}_{i=1,2,\ldots n}$ such that the pair (a_i, b_i) is an S-reduction pair in the complex S^{i-1}, where the S-complexes S^i are defined recursively by*

$$S^0 := S \ , \qquad \text{and} \qquad S^i := S^{i-1} \setminus \{a_i, b_i\} \quad \text{for} \quad i = 1, 2, \ldots, n \ .$$

*We use the abbreviation S^M for the last S-complex in the sequence $\{S^i\}$, and we call this S-complex an M-*reduction of *S in the following.*

The coreduction homology algorithm presented in [23] can now be adapted to lead to the construction of coreduction homology models. We present the resulting algorithm in Table 1, and deduce the following result concerning its validity. The proof is based on a straightforward adaptation of the proof of Theorem 6.2 in [23] and is left to the interested reader.

Theorem 3.3. *Let S denote a non-empty S-complex. If the algorithm presented in Table 1 is applied to S, then it returns a pair (M, S') where M is a coreduction homology model* *of S and $S' = S^M$. Moreover, assuming S is stored in a bitmap, the complexity of this algorithm is $O(|S|w(S)^2)$.*

3.4. **Coreduction Homology Algorithm for Inclusions.** Theorem 3.1 indicates that when an S-complex S admits a small homology model, then the approach proposed in Section 3.1 for the two steps discussed at the beginning of this section may indeed speed up computations. For the complexity results presented in the rest of the paper we denote the number of generators in a basis of $H(S)$ by $g(S)$. We also consider the class $\mathcal{S}_\delta$ of S-complexes such that for some constant $K > 0$ and every S-complex $S \in \mathcal{S}_\delta$

$$w(S) \leq K, \tag{7}$$

$$|S^M| \leq |S|^\delta, \tag{8}$$

where M is the homology model of S provided by the algorithm in Table 1. Note that (7) is satisfied for instance in the class of cubical complexes of fixed dimension and all their S-subcomplexes. A class of S-complexes satisfying (8) is discussed in [23]. Numerical experiments indicate that in many situations δ may be taken quite small. This is important because of the following straightforward proposition.

Proposition 3.4. *For $S \in \mathcal{S}_\delta$ the complexity of finding the homology model M of S followed by applying Smith diagonalization to the matrix of the boundary map of S^M or any other matrix of the size of the boundary matrix of S^M is*

$$O(|S|^{\gamma(\delta)}),$$

where

$$\gamma(\delta) := \max(1, \alpha\delta).$$

and $\alpha = 3.376..$ is the complexity exponent in the Smith diagonalization algorithm. In particular, if $\delta \leq 1/\alpha$ then the complexity is linear.

We are now ready to analyse the algorithms and their complexity for the two steps discussed at the beginning of this section. For the first step we recall that in order to obtain a basis of $H(S)$ it is enough to find such a basis U^M in $H_*(S^M)$, and then lift this basis back to $H_*(S)$ via the map ι^M. Therefore, Theorem 3.1 leads to the following result.

Theorem 3.5. *Given an S-complex $S \in \mathcal{S}_\delta$, a basis of $H(S)$ may be computed in time*

$$O(g(S)|S|^{\gamma(\delta)}).$$

Proof: Let $S \in \mathcal{S}_\delta$. By Theorem 3.3 the cost of constructing a homology model M of S does not exceed $C_1K^2|S|$ for some $C_1 > 0$. To find the basis of $H(S^M)$ it is enough to apply Smith diagonalization to the boundary map in S^M, so the cost does not exceed $C_2|S^M|^\alpha \leq C_2|S|^{\alpha\delta}$. By Theorem 3.1 the cost of transporting $g(S)$ homology generators is $g(S)C_3K|S|$. Therefore, the total cost does not exceed

$$C_1K^2|S| + C_2|S|^{\alpha\delta} + g(S)C_3K|S| \leq (C_1K^2 + C_2 + C_3K)g(S)|S|^{\gamma(\delta)},$$

which gives the required conclusion. □

```
function CoefficientVector(S-complex $S$, homology model $M$,
      homology basis $U^M$, chain $z$)
begin
   $z' := \psi^M(z)$;
   $D$ :=the matrix of the boundary operator in $S^M$;
   solve $z' = U^M x + Dy$ for $x, y$;
   return $x$;
end;
```

Table 2. Coefficient vector through a homology model.

```
function HomologyInclusion(S-integer $q$, S-complex $S$,
   homology model $M$, homology bases $V, U^M$)
begin
   $Q$ :=empty sequence of vectors;
   for $i := 1$ to $V$.length do
      $Q[i]$ :=CoefficientVector($S, M, U^M, V[i]$);
   endfor;
   return $Q$;
end;
```

Table 3. Coreduction homology algorithm for inclusions.

The second step is formalized in the algorithm presented in Table 2, and the following result confirms its correctness and provides its complexity. The proof of correctness is straightforward and the complexity analysis is similar to the proof of Theorem 3.5 and is left to the reader.

Proposition 3.6. *Let S denote a non-empty S-complex, let M denote a coreduction homology model for S, let U^M denote a basis for the homology group $H_q(S^M)$, and let z denote a cycle $z \in Z_q(R(S))$. Then the algorithm in Table 2, when applied to this input data, returns the coefficient vector of the decomposition of the homology class of z with respect to the basis $\iota^M(U^M)$ of $H_q(S)$. The complexity of the algorithm is $O(|S|^{\gamma(\delta)})$.*

Despite the simplicity of the two above algorithms, they form the core of the coreduction algorithm for the homology computation of inclusion maps, presented in Table 3. Altogether, we have now deduced the following result in which the complexity analysis is similar to that in Theorem 3.5 and is skipped.

Theorem 3.7. *Let S denote a non-empty S-complex and let M denote a coreduction homology model for S. Furthermore, let V denote a set of generators for a closed S-complex $T \subset S$, and let U^M denote a set of generators in S^M. Then the algorithm in Table 3, when applied to the listed input data, returns the matrix of the homology of the inclusion $\rho : T \to S$ with respect to the bases induced by V and $U := \iota^M(U^M)$. The cost of running this algorithm is $O(g(T)|S|^{\gamma(\delta)})$.*

Finally, taking into account the cost of constructing homology models and bases, we obtain the following corollary.

Corollary 3.8. *The cost of computing the map induced in homology by the inclusion $T \subset S$ of two S-complexes in $\mathcal{S}_\delta$ is*

$$O(\max(g(T), g(S))|S|^{\gamma(\delta)})).$$

4. Persistent Homology and An Extension

In this section we demonstrate how the coreduction algorithm for inclusion maps can be applied to the computation of persistence intervals. As mentioned in the introduction, the concept of persistent homology was introduced in [9, 33, 35, 38] and is aimed at identifying and measuring the significant topological features in an increasing filtration of topological spaces, and our persistence algorithm is described in Section 4.1. After that, Section 4.2 introduces an extension of the persistence idea to two-sided filtration, which will turn out to be useful for our applications in the next section.

4.1. **Persistence Intervals.**

The concept of persistence is aimed at identifying and measuring the significant topological features in an increasing filtration of topological spaces. See for example [8, 14] for an introduction. In our situation, we will consider an increasing sequence of S-complexes. As one moves through such a sequence, non-trivial homology generators will be generated and destroyed. Persistent homology keeps track of the birth and death of these generators by furnishing associated persistence intervals. These intervals allow one to study significant topological features which exist over many spaces in the filtration, i.e., over large persistence intervals. To briefly illustrate the main ideas, consider the example shown in Figure 1. In this figure we consider a filtration X_h which is indexed by the height h, and which collects, for a given height h, all blue elementary cubes (dark or light blue) below the level h. In the right half of the figure, the 0-dimensional persistence intervals are shown. Each interval starts at the birth height of a new component, and exists as long as the component persists. This implies that some of the intervals are of finite length; in fact, these intervals correspond exactly to components which merge with other components for increasing h. The remaining intervals are unbounded, and they correspond to *essential homology classes*, i.e., homology classes which are still present in the largest complex of the filtration.

In order to describe our approach to the computation of persistence intervals, we consider a sequence $S = (S^1, S^2, \ldots, S^n)$ of S-complexes. We say that S is a *filtration*, if $S^i \subset S^{i+1}$ and S^i is closed in S^{i+1} for all $i = 1, 2, \ldots, n-1$. Following Edelsbrunner, Letscher, and Zomorodian [9], for $i < j$ one can then define the (i,j)-persistent q-th homology group of S via

$$PH_q^{i,j}(S) := Z_q(S^i) / \left(B_q(S^j) \cap Z_q(S^i) \right) \ .$$

The (i,j)-persistent Betti number of the filtration S in dimension q is then the rank of the group $PH_q^{i,j}(S)$, and is denoted by $\beta_q^{i,j}$. The following proposition, which was originally

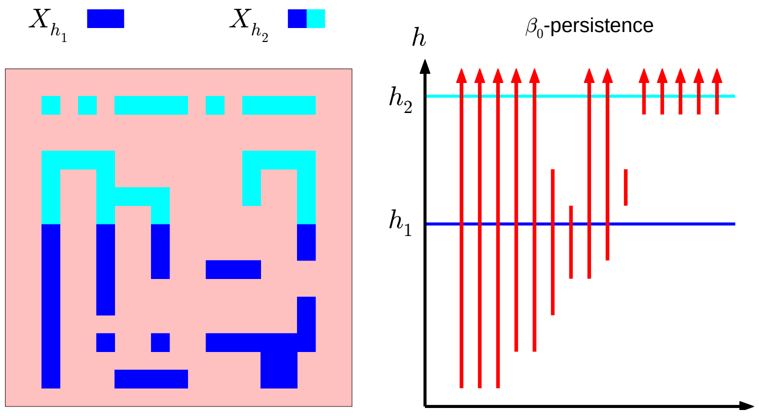

Figure 1. Regular β_0-persistence for an artificial cubical filtration X_h. If h denotes a fixed height, then X_h consists of all points in the light or dark blue region on the left which are at most at height h. Thus, the dark blue region is X_{h_1}, and the union of dark and light blue corresponds to X_{h_2}. The β_0-persistence intervals are shown on the right in red, essential persistence classes are indicated by arrows.

established by Zomorodian [38], can then easily be obtained for the case of arbitrary S-complexes, and we omit the straightforward proof. Similar results in the context of size functions can be found in [2].

Proposition 4.1. *Let $S = (S^1, S^2, \ldots, S^n)$ denote a filtration of S-complexes, and denote the inclusion maps corresponding to $S^i \subset S^j$ by $\rho^{i,j} : S^i \to S^j$. Then the (i,j)-persistent q-th homology group $PH_q^{i,j}(S)$ of S is isomorphic to the image of $H_q(\rho^{i,j})$.*

This result takes on a particular nice form if the ring R is actually a field. In this case, it can readily be seen that

$$\beta_q^{i,j} = \text{rank } \rho_q^{i,j} .$$

It was demonstrated in [39, Definition 3.2] that the collection of homology modules $H_*(S^j)_{j=1,2,\ldots n}$, together with the homomorphisms $\rho_*^{j,j+1} : H_*(S^j) \to H_*(S^{j+1})$, forms a so-called *persistence module*. For the specific case of R being a field, this module is in one-to-one correspondence with an $R[t]$-module of the form

$$\bigoplus_{l=1}^{m} t^{i_l} R[t] / \left(t^{j_l - i_l} \right) \ ,$$

for some uniquely determined sequence of intervals (i_l, j_l) with integer endpoints satisfying $i_l < j_l$. See [39, Theorem 3.1, Corollary 3.1] for more details. Given a pair of integers (i,j), one can then define the number of (i,j)-persistence intervals in S as

$$\text{pi}(S,i,j) := \text{pi}(i,j) := \text{card} \left\{ l \in \mathbb{Z} \mid 1 \le l \le m \text{ and } (i_l, j_l) = (i,j) \right\} .$$

In practice, the number $\text{pi}(S,i,j)$ gives the number of homology generators which start (or are born) at level i in the filtration, and which survive exactly until level $j-1$ (at which

```
function PersistenceIntervals(filtration of S-complexes $S$, integer $q$)
begin
    $n :=$ $S$.length;
    for $j := 1$ to n do
        $(M[j], S^M[j]) :=$HomologyModel($S[j]$);
        $U^M[j] :=$HomologyBasis($q, S^M[j]$);
        $U[j] := \iota_{M[j]}(U^M[j])$;
        if $j > 1$ then
            $A[j-1] :=$
                    HomologyInclusion($q, S[j], M[j], U[j-1], U^M[j]$);
        endif;
    endfor;
    $Q :=$ identity matrix;
    for $j := 2$ to n do
        $(A[j-1], Q) :=$RowEchelon($A[j-1] * Q$);
        $C :=$ identity matrix;
        for $i := j-1$ downto 1 do
            $C := C * A[i]$;
            $\beta[i,j] :=$rank $C$;
        endfor;
    endfor;
    for $j := 2$ to n do
        for $i := j-1$ downto 1 do
            $p[i,j] := \beta[i,j-1] - \beta[i-1,j-1] - (\beta[i,j] - \beta[i-1,j])$;
        endfor;
    endfor;
    return $p$;
end;
```

Table 4. Coreduction homology algorithm for persistence intervals.

point the homology generator dies); see also [38, Section 3.3]. Using the interpretation of $\beta_q^{i,j}$ in terms of homology generators, one can now easily verify that for a field R we have

$$\mathrm{pi}_q(i,j) = \left(\beta_q^{i,j-1} - \beta_q^{i-1,j-1}\right) - \left(\beta_q^{i,j} - \beta_q^{i-1,j}\right) , \tag{9}$$

see also [8, p. 152]. In other words, finding the collection of all persistence intervals of the filtration S can be reduced to computing the numbers $\beta_q^{i,j}$. Since this in turn reduces to determining the rank of the homologies of inclusion maps, the applicability of the coreduction algorithm of the previous section is evident. We would like to point out that the above function $\mathrm{pi}_q(i,j)$ has been studied in the context of size functions under the name of *multiplicity of cornerpoints*. See [11, 18] for more details.

The specific algorithm which applies coreduction homology models to the computation of persistence intervals is presented in Table 4. Apart from the algorithms described in this paper, it makes use of two standard algebraic algorithms. The first is `HomologyBasis`. It takes an integer q and an S-complex S on input and returns a basis of $H_q(S)$ in the form of a matrix in which basis elements are stored as columns of the matrix. The other algorithm is `RowEchelon`. It takes a matrix A on input and returns a pair of matrices (A', Q) such that A' is in row echelon form and

$$A' = Q^{-1}A. \tag{10}$$

Let us emphasize again that although the construction of basis and bringing to row echelon form are purely algebraic algorithms of cubical complexity, they are applied to the coreduction homology models. This results in efficiency gains if the homology models are small when compared with the size of the original problem. Altogether, we have the following result.

Theorem 4.2. *Let $S = (S^1, S^2, \dots, S^n)$ denote a filtration of S-complexes, and let q denote an integer. Then the algorithm in Table 4 applied to this input data returns a lower triangular matrix p such that $p[i,j]$ denotes the number of (i,j)-persistence intervals in S in dimension q. If all $S^i \in \mathcal{S}_\delta$, then the complexity of this algorithm is*

$$O\left(n^2 |S^n|^{\gamma(\delta)} \max\{\, g(S^i) \mid i = 1, 2, \dots, n \,\}\right). \tag{11}$$

Proof: Using (10) it is straightforward to verify that on the i-th pass of the internal **for** loop and j-th pass of the second external **for** loop the C variable contains the matrix of the inclusion map $\rho^{i,j}$, transformed into row-echelon form for quick determination of its rank. Therefore, the first part of the conclusion follows from (9). If all $S^i \in \mathcal{S}_\delta$, then by the results of Section 3.4 we can estimate the cost of one pass of the first, second and third external **for** loop respectively by

$$C_1|S^n| + C_2|S^n|^{\alpha\delta} + C_3|S^n| \max_i g(S^i) + C_4|S^n|^{\gamma(\delta)} \max_i g(S^i),$$
$$C_5|S^n|^{\gamma(\delta)} + C_6 n |S^n|^{\gamma(\delta)},$$
$$C_7 n.$$

Since each of the three external loops is executed at most n times, we obtain the total estimate (11). □

Our algorithm provides an alternative to the persistence algorithm developed in [39] for simplicial homology, which has been adapted to the case of cubical complexes in [26, 32]. In the context of zero-dimensional persistent homology, reduction methods have previously been introduced in [12].

4.2. **Extended β_0-Persistence.** Despite the fact that persistence has been applied successfully to many problems in the applied sciences, the information provided by the persistence intervals does not always suffice. To see this, consider again Figure 1, which gives a simple example of a "directed filtration" with respect to height, which is generated from the cubical complex X consisting of all light and dark blue elementary cubes.

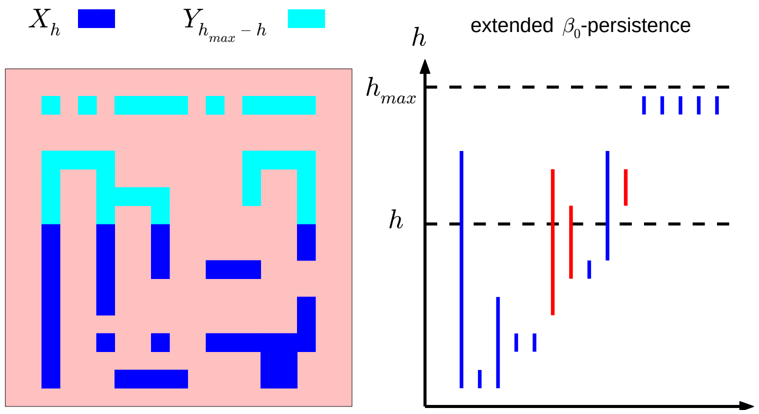

FIGURE 2. Extended β_0-persistence for an artificial cubical complex. If h denotes a fixed height, then X_h consists of all points in the light or dark blue region on the left which are at most at height h, and $Y_{h_{\max}-h}$ contains all points which are at least at height h. Thus, the dark blue region is X_h, and the light blue corresponds to $Y_{h_{\max}-h}$. The extended β_0-persistence intervals for the essential homology classes are shown on the right in blue, the regular ones for non-essential classes in red.

In this example, the persistence intervals clearly detect at which levels the components of X appear for the first time, but their disappearance levels remain unknown, unless a component merges with another component. Thus, the essential homology classes only provide one part of the interesting information. For certain applications, however, the missing piece of information is crucial; see Section 6.3 below.

Of course, finding the levels at which the components disappear can be achieved by considering a new filtration Y_h, which constructs the cubical set from top to bottom. To be more precise, assume that we are given two filtrations X_h and Y_h, for h-values between 0 (without loss of generality) and $h_{\max}$, and assume that these two filtrations furnish the same maximal complex $X = X_{h_{\max}} = Y_{h_{\max}}$. We are interested in the *essential homology classes*, i.e., the non-trivial homology classes of the final complex X. More precisely, we would like to determine the birth times of these classes in both filtrations. Obviously, finding these birth times can easily be accomplished by performing two persistence computations, one for each filtration. The problem is that in the end, we would like to be able to match up the two birth times for a given generator. In general, this cannot be easily accomplished, since the basis representations of the essential homology classes may not be compatible. One approach based on Poincaré duality was presented in [3]. For the applications which we have in mind, it is sufficient to consider the matching for the 0-dimensional homology groups. It can easily be seen that in this case, the basis transformation between the homology bases of $H_0(X)$ constructed via the X_h and the Y_h filtrations is always given by a permutation of a diagonal matrix. More precisely, the implementation of the algorithm for inclusion maps described in the previous section always returns bases for the 0-dimensional homology groups which are represented by a

Size	1152	2048	3200	4608	6272	8192
graph approach	6.72	24.6	117	453	600	2680
coreduction approach	0.047	0.063	0.109	0.141	0.187	0.235

Table 5. CPU time in seconds of the computation of the matrix of homology inclusion based on the graph approach [19] (middle line) and the coreduction approach (bottom line) for various rescalings (top line).

single vertex for each generator. Thus, the matching matrices which describe the basis change to the homology basis given by the homology model for a set in the filtration is a bijection which maps vertex subsets to vertex subsets — and therefore at every level these matrices are permutations on subsets of vertices. Furthermore, both runs of the persistence algorithm described above use the same homology model for the final complex $X = X_{h_{\max}} = Y_{h_{\max}}$. Thus, the matching can be done in a straightforward way by the product of one permutation matrix and the inverse of another one. We would like to point out that our notion of β_0-persistence is more general than the one described in [3], since in our situation the only requirement is the equality of the maximal complexes $X_{h_{\max}}$ and $Y_{h_{\max}}$. No other restrictions have to be imposed on the filtrations X_h and Y_h, which opens the door for a variety of different applications.

Rather than describing the implementation in detail, we illustrate this idea using our example from Figure 1. In Figure 2 we depict the definition of the two filtrations X_h and Y_h, which start at the bottom or top of the cubical set X, respectively. If we then pair the birth time $h_b = h_b^X$ of a component of X with respect to the filtration X_h with the level $h_d = h_{\max} - h_b^Y$, where h_b^Y denotes the birth time of the component in the filtration Y_h, then the interval $[h_b, h_d]$ contains the desired information for the essential homology classes. These intervals are shown in blue in Figure 2.

5. Numerical Experiments

In this section we discuss the performance of the coreduction homology algorithm for inclusions and persistence based on numerical experiments with the implementation by the first author [20].

5.1. **The Coreduction Homology Algorithm for Inclusions.** In order to test the coreduction homology algorithm for inclusions we compared the implementation [20] with the algorithm for homology maps based on [19] and implemented by P. Pilarczyk [41]. The input consisted of a pair of cubical sets $A \subset X$ in several rescalings with A homeomorphic to a torus and X homeomorphic to a torus with an internal wall. The outcome of the comparison performed on a 2.2 GHz Pentium processor with 2GB of RAM is presented in Table 5. The same data is presented graphically in Figure 3.

5.2. **The Coreduction Homology Algorithm for Persistence.** In order to test the coreduction homology algorithm for persistence we compared the algorithm in Table 4

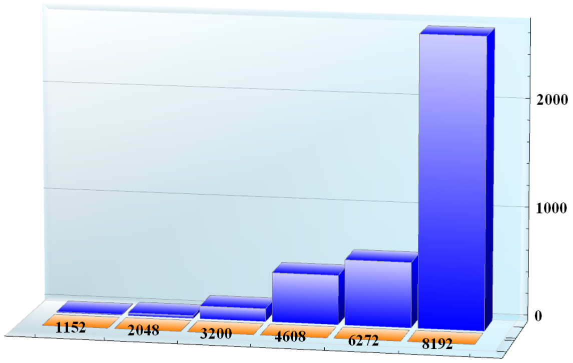

FIGURE 3. CPU time in seconds of the computation of the matrix of homology inclusion based on the graph approach [19] (blue) and the coreduction approach (yellow) for various rescalings. The bars are labelled with the input size.

grid 1024 × 1024						
Input Size	178 176	352 256	526 336	700 416	874496	1 048 576
classical approach	38	278	405	693	742	3299
coreduction approach	11	53	122	214	333	470
speedup	3.5	5.3	3.3	3.2	2.2	7.0
grid 2048 × 2048						
Input Size	712704	1409024	2105344	2801664	3497984	4194304
classical approach	259	6558	24465	26502	27513	36187
coreduction approach	229	904	2063	3653	5619	8012
speedup	1.1	7.3	12.	7.3	4.9	4.5
grid 100 × 100 × 100						
Input Size	200000	360000	520000	680000	840000	1000000
classical approach	291	1386	5061	17232	40474	60407
coreduction approach	25	139	571	1243	2601	4025
speedup	12.	10.	8.9	14.	16.	15.

TABLE 6. CPU time in seconds and the speedup factor for the computation of persistence intervals based on the classical approach [26] and the coreduction approach [20].

with the classical, purely algebraic algorithm for persistent homology [9, 38]. The tests were performed on a 3.0GHz Intel Xeon processor with 16GB of RAM.

We considered three data sets: A 17 level filtration on a 1024 × 1024 cubical grid, an 18 level filtration on a 2048 × 2048 cubical grid, and a 25 level filtration on a 100 × 100 ×

grid 1024×1024						
Input Size	178 176	352 256	526 336	700 416	874496	1 048 576
classical approach	0.43	0.85	1.3	1.7	2.1	2.5
coreduction approach	0.23	0.25	0.26	0.29	0.30	0.33
decrease	2	3	5	6	7	8
grid 2048×2048						
Input Size	712704	1409024	2105344	2801664	3497984	4194304
classical approach	1.7	3.4	5.1	6.7	8.4	10.
coreduction approach	0.29	0.39	0.50	0.63	0.79	0.94
decrease	6	9	11	11	11	11
grid $100 \times 100 \times 100$						
Input Size	200000	360000	520000	680000	840000	1000000
classical approach	1.3	2.3	3.3	4.3	5.2	6.2
coreduction approach	0.23	0.26	0.28	0.30	0.32	0.34
decrease	6	9	12	14	16	18

Table 7. Memory usage in GB and the decrease factor for the computation of persistence intervals based on the classical approach [26] and the coreduction approach [20].

100 cubical grid. For each data set we selected six subsets of varying size and ran the implementation [20] of the algorithm in Table 4 and the implementation by V. Nanda [26] of the classical persistence algorithm [9, 38]. The comparison of CPU time is gathered in Table 6 and is also presented graphically in the left column of Figure 4. The average speedup factor is respectively $4, 6$ and 12 for the case of the 1024×1024, 2048×2048 and $100 \times 100 \times 100$ cubical grid. Although there is not enough evidence to compare the two algorithms asymptotically, it is clear that in the case of the input sizes accessible by a typical or even top class present day personal computer, the coreduction algorithm for persistence should be expected to perform significantly better than the purely algebraic persistence algorithm. The coreduction algorithm also uses significantly less memory, as indicated in Table 7 and in the right column of Figure 4.

6. Applications

In this final section of the paper we present three materials science oriented applications based on the implementation [20] of the coreduction homology algorithm for persistent homology.

6.1. **Wavelength Characterization in Complex Patterns.** Complicated patterns that change with time can be observed throughout the applied sciences. In many cases, these patterns lack a clear regular geometric structure which makes their quantitative description and comparison difficult. This is particularly true for microstructures created

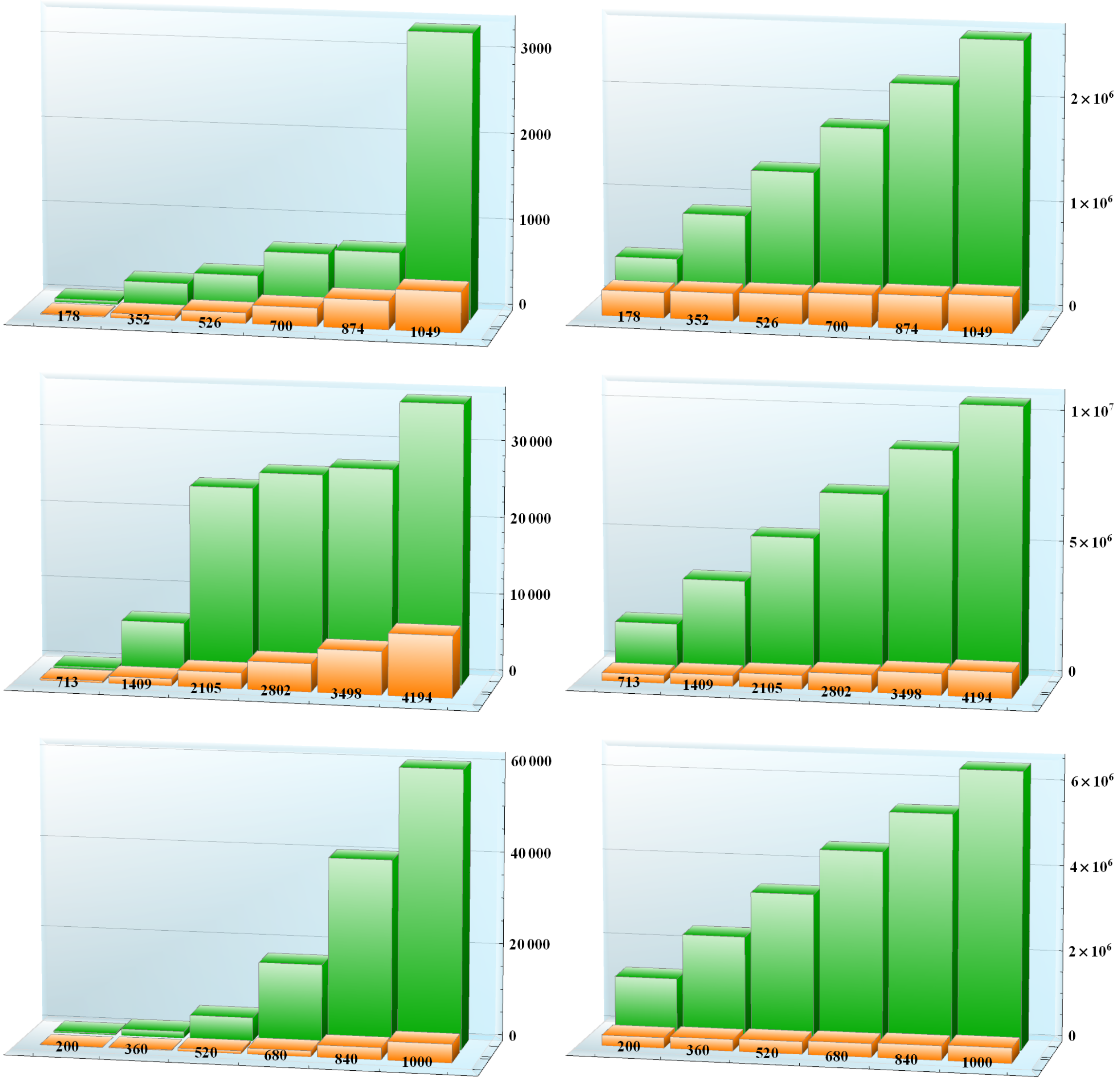

FIGURE 4. Comparison of the CPU time in seconds (left column) and the memory usage in kB (right column) for the grid 1024 × 1024 (top row), the grid 2048 × 2048 (middle row) and the grid 100 × 100 × 100 (bottom row) of the computation of persistence intervals based on classical approach implementation [26] (green) and the coreduction approach implementation [20] (orange). The bars are labelled with the input size counted in thousands.

during phase separation processes in alloys. From a mathematical point of view, algebraic topology is a natural tool for obtaining such a quantification, and recent advances in the algorithmic theory for homology have made it possible to determine homological invariants quickly, even for large data sets. The central question of course is whether homological techniques can offer additional insight.

Only recently a number of studies have appeared that apply homological analysis to problems in materials science, see for example [13], as well as the references therein. These studies have concentrated on describing the topological complexity of time-dependent patterns which are generated by partial differential equation models for phase separation. Many of these models are phenomenological in nature, and trying to identify similarities or — more importantly — differences between the microstructures obtained through simulations and experimental data lies at the heart of model verification. In addition, in many instances there are a variety of models that have been proposed for a specific phenomenon, and one needs to understand their validity from a pattern formation point of view.

Questions as the one described in the last section can be addressed successfully using computational homology, as has been demonstrated in [13]. In this paper, numerically computed microstructures are compared for two classical models of binary alloy phase separation — the deterministic Cahn-Hilliard and the stochastic Cahn-Hilliard-Cook model. The solution $u(t,x)$ to either partial differential equation determines the microstructure determined by one of the two alloy components through the nodal domains

$$N^{\pm}(t) \; = \; \{x \in \Omega \; : \; \pm u(t,x) \geq 0\}$$

where $u : \mathbb{R}_0^+ \times \Omega \to \mathbb{R}$. By computing the homology of these nodal sets for a random ensemble of solutions it was shown in [13] that the deterministic model differs significantly from the stochastic model during the early phase separation stages, and that recent experimental data compares favorably to the stochastic model.

For the above study it was sufficient to determine the homology of the nodal domains through the computation of Betti numbers. In many situations, however, more detailed topological information is necessary, and the persistence algorithm developed in the current paper can serve as such an improved tool in the cubical setting. As we will show below, the use of β_1-persistence turns out to be particularly useful. To illustrate this, consider the viscous Cahn-Hilliard model [1, 27] given by

$$\beta \cdot u_t - (1-\beta) \cdot \epsilon^2 \Delta u_t = -\Delta(\epsilon^2 \Delta u + f(u)) \; ,$$

on a square base domain $\Omega \subset \mathbb{R}^2$ and subject to homogeneous Neumann boundary conditions, where $\beta \in [0,1]$ is the viscosity parameter and ϵ is a small parameter modeling interaction length. This family of equations was proposed as a phase separation model which takes into account interfacial forces, and in fact it serves as a homotopy between the classical Cahn-Hilliard model for $\beta = 1$ and the nonlocal Allen-Cahn model for $\beta = 0$. Similar to [13], one is interested in a quantitative understanding of the nodal domains $N^{\pm}(t)$. Two sample nodal domains are shown in the leftmost column of Figure 5 — the top image corresponds to the Cahn-Hilliard model, while the bottom image was produced by the nonlocal Allen-Cahn equation. It is evident that these patterns differ significantly, and we now demonstrate how persistence can be used to quantify this difference. Fix a nodal domain N, for example the one shown in the first image of the first row in Figure 5 in

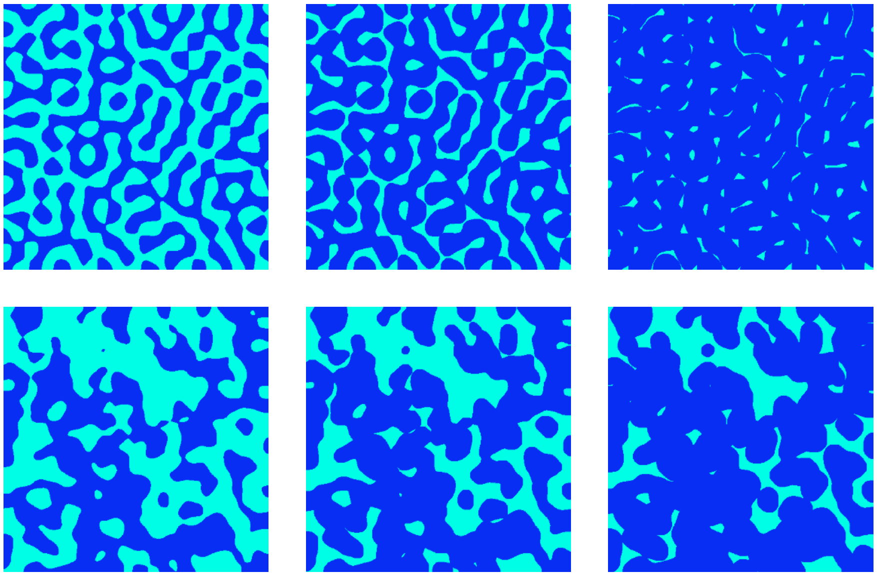

FIGURE 5. Sample Cahn-Hilliard (top row) and nonlocal Allen-Cahn (bottom row) microstructures. In each row, the leftmost image shows the actual microstructure, the next two images show dilated version with radii of $r = 5$ and $r = 10$ pixels, respectively.

dark blue. For $r > 0$ define the dilated nodal domain as the r-neighborhood

$$N_r \;=\; B_r(N)$$

of the original microstructure. The filtration $\{N_r \;:\; r \geq 0\}$ serves as the basis for our persistence computation. As r increases, new one-dimensional generators will be created whenever pinch-offs occur in the dilated patterns. By keeping track of these generators, one can determine quantitative size information on the holes in the microstructure.

For our application, the original nodal domains are given as a 512×512 pixel image, and we choose the radius r in pixel increments. The results of a persistence computation for the one-dimensional generators is shown in the top row of Figure 6 for the Cahn-Hilliard (left) and the nonlocal Allen-Cahn (right) pattern. Both of these persistence diagrams contain several hundred generator intervals, many of which correspond to short-lived generators. Such generators arise for small r-intervals due to the curved interface boundaries, and they will not contribute to our general understanding of the pattern geometry. If we now discard all generators with existence intervals of length three or less, we obtain the persistence diagrams in the second row of Figure 6. These diagrams show that the Cahn-Hilliard pattern exhibits an almost uniform pattern wavelength, i.e., most channels exhibit a thickness between 10 and 15 pixels. Furthermore, the large number of generators indicates snake-like features with many high-curvature turns. On the other hand, the nonlocal Allen-Cahn pattern lacks a clear well-defined wavelength and exhibits considerably larger size holes (note the change of scale on the vertical axis).

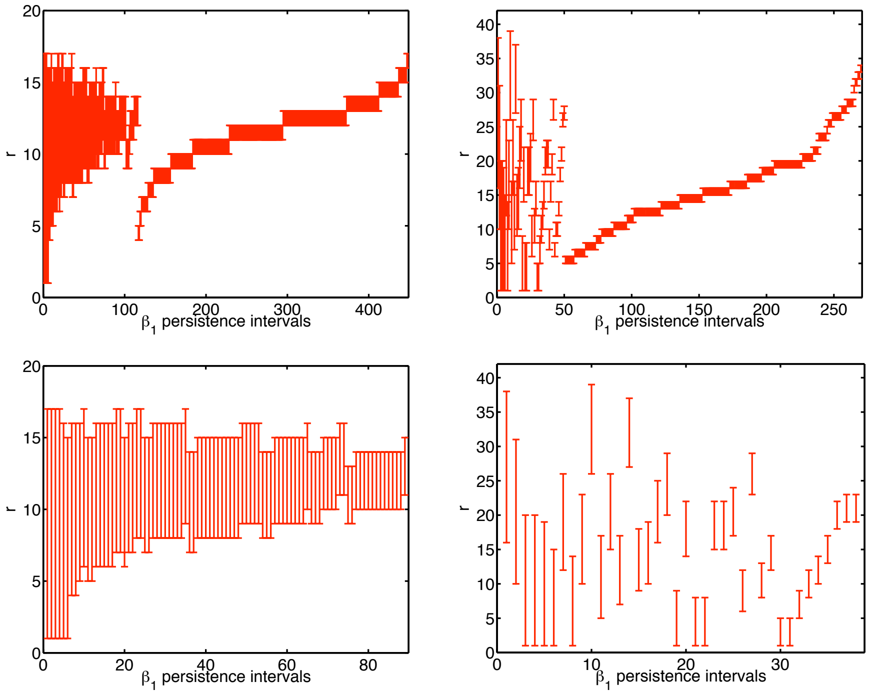

FIGURE 6. Dilation β_1-persistence for the patterns shown in the left column of Figure 5. In each case, the persistence parameter is the dilation radius r, which is represented by the vertical axis. In the two figures in the top row, all persistence intervals in dimension 1 are shown for the Cahn-Hilliard (left) and the nonlocal Allen-Cahn (right) pattern. The bottom row shows only the persistence intervals of length larger than 3. In all of these diagrams, the horizontal axis enumerates the persistence intervals, while the vertical axis represents the dilation parameter r.

6.2. **Stress Networks in Polycrystals.** Natural building stones such as marble are widely used in both structural and decorative applications. However, many marbles show limited durability. One such deterioration phenomenon is the bowing of marble panels, which for example has been observed at the university library in Göttingen [37] or at the Finland Hall in Helsinki [28] — in each case only a few years after the building's completion. This bowing process is accompanied by a reduction of strength properties, and can lead to the static failure of a facade with time. One major factor in such degradation phenomena is thermal heating and cooling. Due to the anisotropic thermal expansion of the underlying materials calcite and dolomite, stresses are created which lead to granular decohesion of the material with time. Similar effects have also been responsible for microcracking in polycrystalline ceramics [34]. In both cases, regions with high maximal principal stresses form complicated networks, and these networks and their topology play a significant role in the creation of microcracks. From a practical point of view, it is desirable to understand the mechanism responsible for the network creation, and to quantify

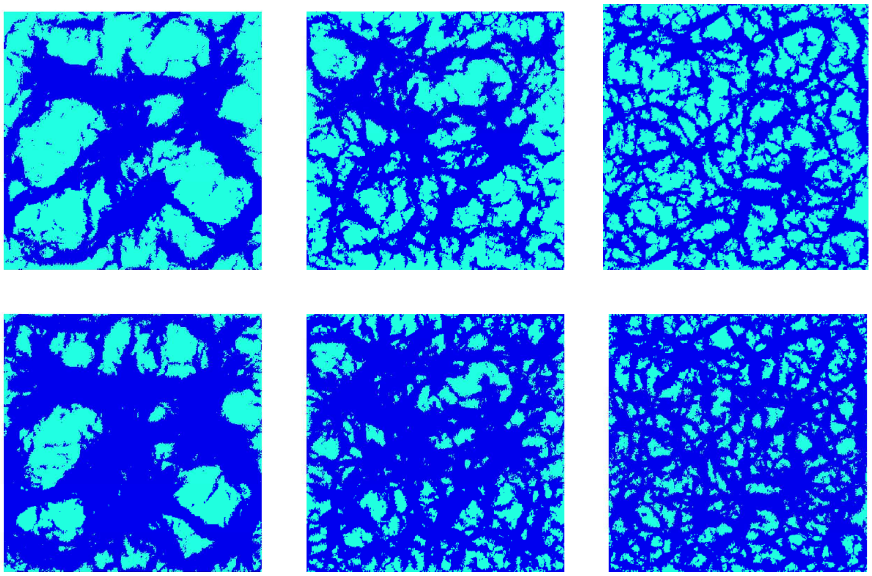

Figure 7. Maximal principal stress networks in alumina for different grain misorientation distribution functions. From left to right the figures correspond to misorientation distributions which result in predominantly low-angle, in uniform, and in predominantly high-angle grain misorientation. The dark blue regions in the top and bottom rows roughly correspond to stresses above 20 MPa (megapascals) and 12 MPa, respectively.

the topology of the resulting network. If the high stress network covers a large portion of the material, microcracking is more likely to occur.

From Figure 7 it appears that each network exhibits some characteristic width, or length scale, whose appearance is somewhat surprising due to the lack of a well-defined length scale in the underlying materials. Materials as the ones mentioned above are organized in the form of grains: Even though each grain consists of the same crystalline material, the grain orientations differ. Along the grain boundaries, the orientation mismatch of adjacent grains results in stress formation due to thermal anisotropy, i.e., due to the fact that the grains expand/contract differently depending on their orientation. Thus, one would expect that the topology of the grain microstructure and the specific distribution of the grain orientations are responsible for the stress network structure, and that the highest stresses occur at the grain boundaries. Yet, the three stress networks shown in Figure 7 are all for the same grain microstructure, and the specific distribution of the grain orientations is the same in all three cases! A qualitative explanation of the differences in the three networks has been given in [30]. Through rearrangement of the given set of grain orientations, Fuller and Saylor generated misorientation distributions with predominantly low angles, random angles, and predominantly high angles across the grain boundaries. The corresponding stress networks are the ones shown in Figure 7 from left to right.

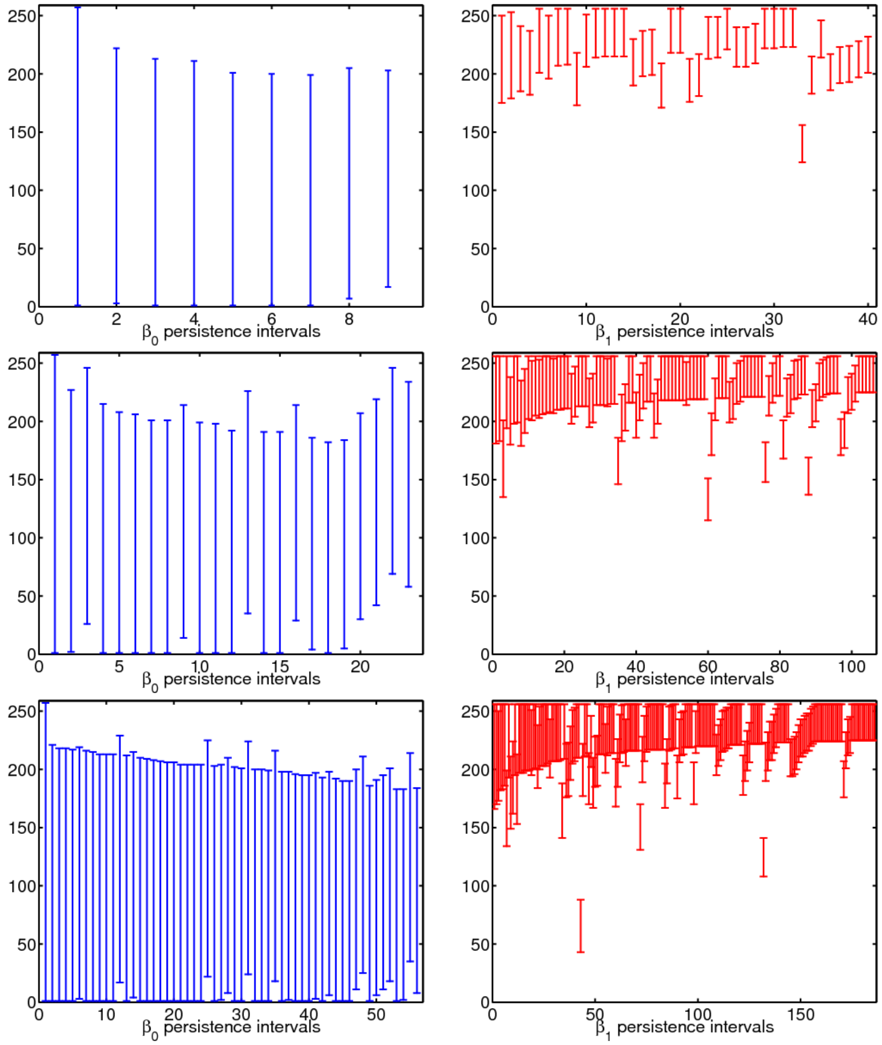

Figure 8. Persistence diagrams for the three stress networks from Figure 7. The vertical axis corresponds to the threshold level θ, which is related to the maximal principal stress $\sigma_{\max}$ in MPa (megapascals) via $\sigma_{\max} = 110 - 100\theta/255$.

More recently, computational homology was used in [36] to develop a metric for distinguishing between these different stress networks and for relating their appearance to the grain boundary misorientations. While we refer the reader to [36] for more details, the results in this paper were obtained by applying Betti number computations to the stress networks as a function of the stress level; see again Figure 7 for these networks at two different threshold levels θ, which are related to the maximal principal stress $\sigma_{\max}$ in MPa (megapascals) via $\sigma_{\max} = 110 - 100\theta/255$. It is evident from these images that the provided stress data contains a large number of numerical artifacts which could lead to spurious generators in the first dimension, and thus make a meaningful interpretation

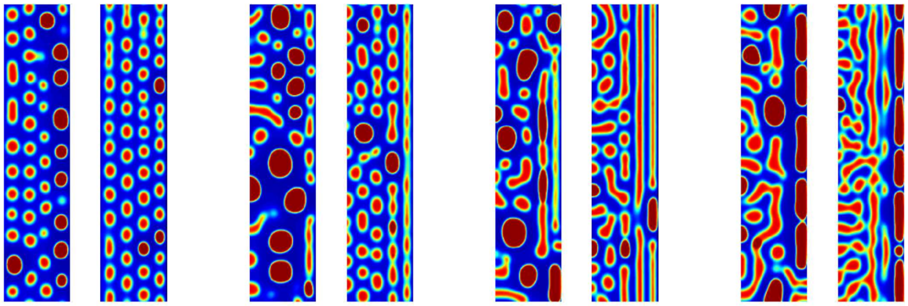

FIGURE 9. Eight sample drug polymer mixtures generated via phase field simulations. In each sample, the stent surface is on the right vertical boundary.

of the Betti number results impossible. In order to avoid this issue, the Betti numbers in [36] were not computed for the patterns shown in Figure 7, but for smoothed versions.

Smoothing the stress networks is somewhat ad hoc, and while it does provide interesting results, it would be desirable to work with the original data directly and dispense with the artificial generators through topological methods. Exactly this can be achieved with the persistence algorithm presented earlier. We apply this algorithm to cubical complexes S_θ which contain all image pixels of stress level at least θ, for $\theta = 0, \ldots, 255$. The resulting persistence diagrams, after the removal of topological noise (in this case, persistence intervals of lengths smaller than 175 for the zero-dimensional case and 30 for the one-dimensional one), are shown in Figure 8. Also in this case, the persistence diagrams can quantitatively distinguish between the different stress networks, thus avoiding the need for additional ad hoc smoothing techniques.

6.3. **Topology of Controlled Drug Release Systems.** Our final application is concerned with the development of topological metrics for controlled drug delivery systems such as drug-eluding arterial stents. In this setting, the drug is incorporated into a polymer matrix which acts as a diffusion barrier and slows the rate of the drug release. These drug-polymer coatings are usually fabricated by dissolving a homogeneous mixture of drug and polymer into a solvent, and subsequently letting the solvent evaporate. Depending on the materials and the precise manufacturing conditions the resulting drug-polymer microstructure can exhibit a variety of complicated time-dependent structures. From a practical point of view, one is particularly interested in developing metrics which can be used to link the processing stage to the resulting microstructure and ultimately to the drug release profiles. For this it is necessary to be able not only to distinguish between topological differences of the microstructure, but also to describe the topology in a "directional sense." Once a stent has been implanted, it will start to dissolve and release the incorporated drug from the surface of the mixture downwards, and one has to be able to characterize at which depths drug droplets appear, and up to which depths they extend.

In the recent paper [29] a model has been proposed for simulating the formation of the drug polymer microstructure. For this, a small number of crystal particles are grown

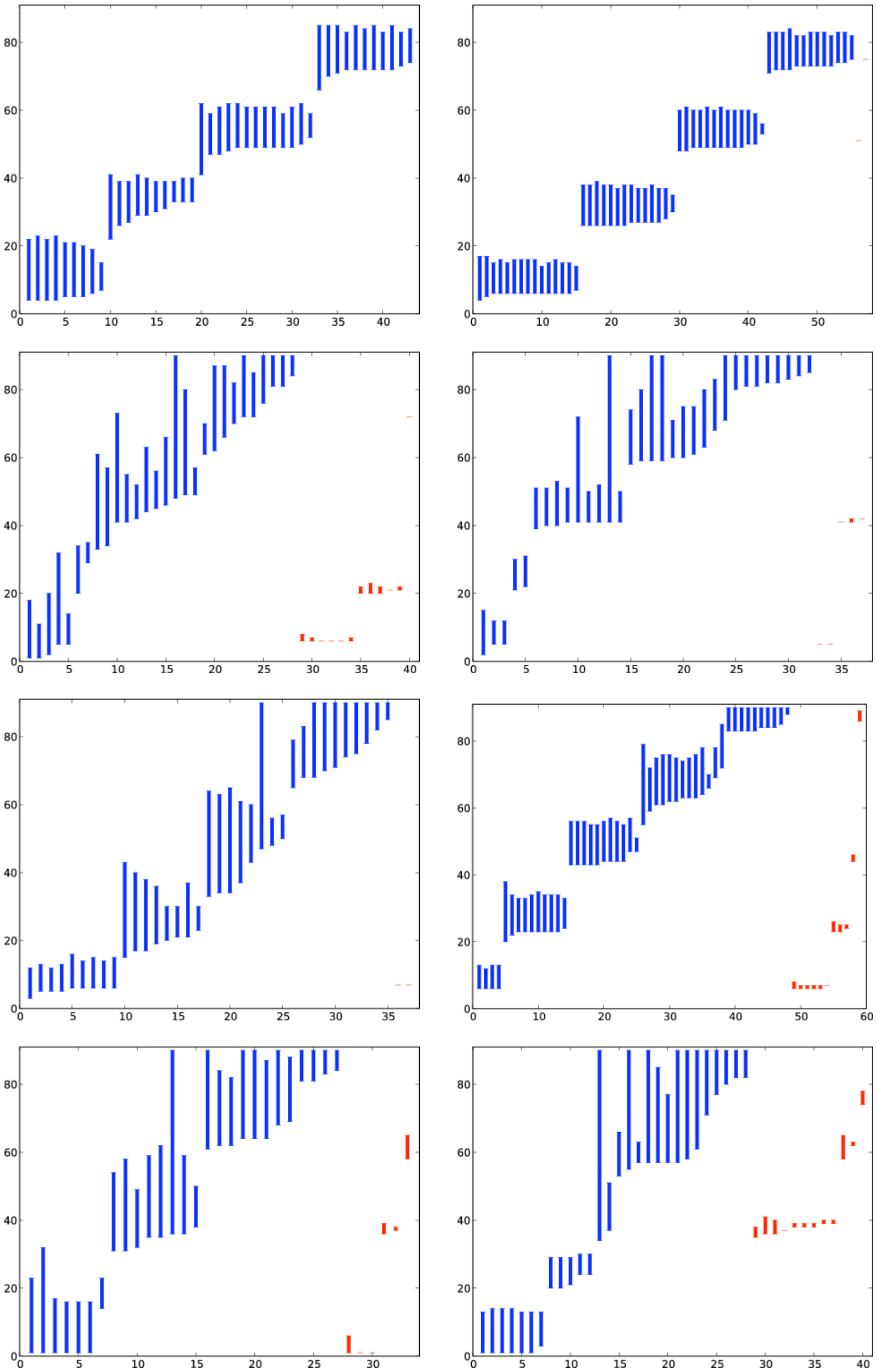

Figure 10. Extended β_0-persistence intervals for the drug polymer mixtures of Figure 9.

into a phase separated matrix. This crystal growth process can take place in the context of both nucleation and spinodal decomposition, depending on the concentrations of the involved components. The underlying model for this processing stage is a coupled system of partial differential equations of Cahn-Hilliard and phase-field type, which describes the evolution of the drug, polymer, and solvent concentrations, as well as the evolution of an order parameter for distinguishing between amorphous and crystalline states. Some of the resulting drug polymer microstructures are shown in Figure 9.

As we mentioned before, one can only hope to capture the topological information necessary to distinguish the microstructures and the possible resulting drug release profiles if the proposed metric does take into account the inherent directionality. We therefore propose to use the concept of extended β_0-persistence developed in Section 4.2, which is based on the usage of two filtrations of topological spaces, rather than one. In our setting, consider the increasing filtration X_h which consists of all points in the base domain which are less then distance h from the top of the stents. For example, in the images shown in Figure 9 this corresponds to all red pixels which are contained in a vertical slab of width h pixels which extends from the right-hand side of the image to the left. Thus, we have obtained an increasing filtration which extends up to a maximum width $h_{\max}$. If one computes persistence on this filtration, the zero-dimensional persistence interval for a given component extends from the depth of origination until $h_{\max}$, unless it merges with another component. Thus, persistence of X_h can detect the appearance of components as a function of height, but not the depth to which the component extends to.

In order to address this issue, extended persistence creates a second increasing filtration Y_d, which in our case also contains all of the red pixels in a vertical slab of width d, but this time the slab is aligned with the left edge of the image, i.e., the bottom side of the stent. Notice that we have $X_{h_{\max}} = Y_{h_{\max}}$, i.e., the largest cubical complexes in both filtrations coincide. Now the further proceeding is clear. Computing persistence on the filtration Y_d allows one to determine the height h (measured from the right edge) at which components disappear: If a component appears in for the first time in Y_d, this component disappears at height $h_{\max} - d$. All one has left to do is to match the generators of the zero-dimensional homology of the final complexes $X_{h_{\max}}$ and $Y_{h_{\max}}$, and this can easily be done using the framework presented in this paper. The resulting zero-dimensional extended persistence intervals for the microstructures of Figure 9 are shown in Figure 10. These diagrams allow one to not only distinguish between different types of microstructures, but also to analyze the depth information which is crucial for determining drug release profiles. A more detailed analysis of this will be presented elsewhere.

References

[1] F. Bai, C.M. Elliott, A. Gardiner, A. Spence, A.M. Stuart, The viscous Cahn-Hilliard equation. I. Computations, *Nonlinearity*, **8**(1995), 131–160.

[2] F. Cagliari, M. Ferri, P. Pozzi, Size functions from a categorial viewpoint, *Acta Applicandae Mathematicae*, **67**(2001), 225–235.

[3] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Extending persistence using Poincaré and Lefschetz duality, *Foundations of Computational Mathematics*, **9**(2009), 79–103.

[4] C.J.A. Delfinado and H. Edelsbrunner, An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere, *Computer Aided Geometric Design* **12**(1995), 771–784.

[5] B.R. Donald and D.R. Chang, On the complexity of computing the homology type of a triangulation. In *Proc. 32nd Ann. IEEE Sympos. Found. Comput. Sci.*(1991), 650-661.

[6] J.-G. Dumas, F. Heckenbach, D. Saunders and V. Velker, Computing simplicial homology based on efficient Smith normal form algorithms, In *Algebra, Geometry and Software Systems*(2003), 177-207.

[7] J.-G. Dumas, B.D. Saunders, G. Villard, On efficient sparse integer matrix Smith normal form computations, *J. Symbolic Computation* **32**(2001), 71–99.

[8] H. Edelsbrunner, J.L. Harer, *Computational Topology: An Introduction*, American Mathematical Society, Providence, 2010.

[9] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, *Discrete and Computational Geometry* **28**(2002), 511–533.

[10] J. Friedman, Computing Betti numbers via combinatorial Laplacians, In *Proc. 28th Ann. ACM Sympos. Theory Comput.*(1996), 386-391.

[11] P. Frosini, C. Landi, Size functions and formal series, *Applicable Algebra in Engineering, Communication and Computing* **12**(2001), 327–349.

[12] P. Frosini, M. Pittore, New methods for reducing size graphs, *International Journal of Computer Mathematics* **70**(1999), 505–517.

[13] M. Gameiro, K. Mischaikow, T. Wanner, Evolution of pattern complexity in the Cahn-Hilliard theory of phase separation, *Acta Materialia*, **53**(2005), 693–704.

[14] R. Ghrist, Barcodes: the persistent topology of data, *Bulletin of the American Mathematical Society* **45**(2008), 61–75.

[15] T. Kaczynski, K. Mischaikow, M. Mrozek, *Computational Homology*, Applied Mathematical Sciences **157**, Springer-Verlag, New York, 2004.

[16] T. Kaczynski, M. Mrozek, M. Ślusarek, Homology computation by reduction of chain complexes, *Computers and Math. Appl.* **35**(1998), 59–70.

[17] W. Kalies, K. Mischaikow, and G. Watson, Cubical approximation and computation of homology, in: *Conley Index Theory*, Banach Center Publications **47**(1999), 115–131.

[18] C. Landi, P. Frosini, New pseudodistances for the size function space, in *Proc. SPIE, 3168, Vision Geometry VI* (R.A. Melter, A.Y. Wu, L.J. Latecki, eds.), pp. 52–60, 1997.

[19] K. Mischaikow, M. Mrozek, P. Pilarczyk, Graph approach to the computation of the homology of continuous maps, *Foundations of Computational Mathematics* **5**(2005), 199–229.

[20] M. Mrozek, *Homology Software*,
http://www.ii.uj.edu.pl/˜mrozek/software/homology.html, 2006.

[21] M. Mrozek,, Index pairs algorithms, *Foundations of Computational Mathematics* **6**(2006), 457–493.

[22] M. Mrozek, Čech type approach to computing homology of maps, *Discrete and Computational Geometry* **44**(2010), 546–576.

[23] M. Mrozek, B. Batko, Coreduction homology algorithm, *Discrete and Computational Geometry* **41**(2009), 96–118.

[24] M. Mrozek, P. Pilarczyk, N. Żelazna, Homology algorithm based on acyclic subspace, *Computers and Mathematics with Applications* **55**(2008), 2395–2412.

[25] M. Mrozek, M. Żelawski, A. Krajniak, A. Gryglewski, S. Han, Homological methods in feature extraction of multidimensional images, *Proceedings of the 2009 2nd International Congress on Image and Signal Processing*, IEEE, pp. 1061-1066, 2009.

[26] V. Nanda, *Cubical Persistence Software*, personal communication.

[27] A. Novick-Cohen, On the viscous Cahn-Hilliard equation, In *Material instabilities in continuum mechanics (Edinburgh, 1985–1986)*, Oxford University Press (1988), 329–342.

[28] G. Royer-Carfagni, Some considerations on the warping of marble facades: The example of Alvar Aalto's Finland Hall in Helsinki, *Construction and Building Materials*, **13**(1999), 449–457.

[29] D.M. Saylor, C.-S. Kim, D.V. Patwardhan, J.A. Warren, Diffuse-interface theory for structure formation and release behavior in controlled drug release systems, *Acta Biomaterialia*, **3**(2007), 851–864.

[30] D.M. Saylor, E.R. Fuller Jr., T. Weiss, Thermal-elastic response of marble polycrystals: Influence of grain orientation configuration, *International Journal of Materials Research*, **2007**(2007), 1256–1263.

[31] A. Storjohann, Near optimal algorithms for computing Smith normal form of integer matrices, In *Proceedings of the 1996 international symposium on symbolic and algebraic computation, ISAAC 1996*, (1996), 267-274.

[32] D. Strömbom, *Persistent Homology in the Cubical Setting*, Master's Thesis, Lulea University of Technology, 2007.

[33] C. Uras, A. Verri, Computing size functions from edge maps, *International Journal of Computer Vision*, **23**(1997), 169–183.

[34] V.R. Vedula, S.J. Glass, D.M. Saylor, G.S. Rohrer, W.C. Carter, S.A. Langer, E.R. Fuller Jr., Residual-stress predictions in polycrystalline alumina, *Journal of the American Ceramic Society*, **84**(2001), 2947–2954.

[35] A. Verri, C. Uras, P. Frosini, M. Ferri, On the use of size functions for shape analysis, *Biological Cybernetics*, **70**(1993), 99–107.

[36] T. Wanner, E.R. Fuller Jr., D.M. Saylor, Homology metrics for microstructure response fields in polycrystals, *Acta Materialia*, **58**(2010), 102–110.

[37] T. Weiss, S. Siegesmund, E.R. Fuller Jr., Thermal degradation of marble: Indications from finite-element modelling, *Building and Environment*, **38**(2003), 1251–1260.

[38] A. Zomorodian, *Computing and Comprehending Topology: Persistence and Hierarchical Morse Complexes.* PhD thesis, University of Illinois at Urbana-Champaign, 2001.

[39] A. Zomorodian, G. Carlsson, Computing persistent homology, *Discrete and Computational Geometry*, **33**(2005), 249–274.

[40] Computer Assisted Proofs in Dynamics: http://capd.ii.uj.edu.pl

[41] Computational Homology Project: http://www.math.gatech.edu/˜chomp/

Marian Mrozek, Institute of Computer Science, Jagiellonian University, ul. St. Łojasiewicza 6, 30-348 Kraków, Poland and Division of Computational Mathematics, WSB-NLU, ul. Zielona 27, 33-300 Nowy Sącz, Poland

E-mail address: Marian.Mrozek@ii.uj.edu.pl

Thomas Wanner, Department of Mathematical Sciences, George Mason University, 4400 University Drive, MS 3F2, Fairfax, VA 22030, USA

E-mail address: twanner@gmu.edu