Graph Approach to the Computation of the Homology of Continuous Maps

Konstantin Mischaikow *
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.
E-mail mischaik at math.gatech.edu

Marian Mrozek †
Division of Computational Mathematics,
Graduate School of Business,
ul. Zielona 27,
33-300 Nowy Sącz
and
Institute of Computer Science,
Jagiellonian University,
ul. Nawojki 11,
30-072 Kraków
E-mail: mrozek at wsb-nlu.edu.pl.

and

Paweł Pilarczyk‡
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.
E-mail: Pawel.Pilarczyk at ii.uj.edu.pl.

March 28, 2005

Abstract

We introduce an efficient algorithm to compute the homomorphism induced in (relative) homology by a continous map. The algorithm is based

*Research supported in part by NSF Grant 0107396

†Partially supported by the Polish Committee for Scientific Research (KBN), grant no. 2 P03A 041 24.

‡Partially supported by the Polish Committee for Scientific Research (KBN), grant no. 2 P03A 041 24. Permanent address: Jagiellonian University, Institute of Computer Science, ul. Nawojki 11, 30-072 Kraków.

on a cubical approximation of the map and the theory of multivalued maps. A software implementation of the algorithms introduced in the paper is available at [27].

1 Introduction

This paper provides an efficient algorithm to be used in the computation of the map on homology induced by a continuous function $f\colon (X,A) \to (Y,B)$. This work is motivated by a growing number of applications in which f is not treated analytically, but rather is obtained via rigorous numerical approximation [4, 5, 6, 15, 16, 19] or experimental observation [18]. As such, before describing the results presented here there are three essential issues that need to be addressed: the approximation of f, the representation of the spaces and the function in a combinatorial form that can be manipulated by a computer, and the requirement for dimension independent algorithms.

Beginning with the question of approximation, consider the case of a nonlinear function $f\colon \mathbf{R}^n \to \mathbf{R}^m$. Due to computational errors the best that one can expect is that a careful numerical estimation of f results in a different map f_{num} with the property that given $x \in \mathbf{R}^n$ one can construct $\varepsilon > 0$ such that $\|f(x) - f_{num}(x)\| < \varepsilon$, or equivalently $f(x) \in B_\varepsilon(f_{num}(x))$; that is the correct value of $f(x)$ lies in an ε-ball of the numerical approximation of f. It is this latter formulation that suggests the use of multivalued maps as a means of representing f.

To be more precise, a *multivalued map* $F\colon X \rightrightarrows Y$ is a function from X to the power set of Y, i.e. $F(x) \subset Y$ for every $x \in X$. We impose the additional assumption that $F(x) \neq \emptyset$. A continuous map $f\colon X \to Y$ is called a *selector* of $F\colon X \rightrightarrows Y$ if $f(x) \in F(x)$ for every $x \in X$.

We will use multivalued maps to approximate continuous functions on the level of topology. However, as was mentioned earlier, in order to use the computer we need a combinatorial means of representing these multivalued maps. For this purpose we make use of the cubical theory developed in [11]. As is made clear shortly, this is not an idiosyncratic choice—our algorithms strongly exploit the fact that the product of cubes is a cube and inversely the obvious projections map cubes to cubes.

Recall that an *elementary cube* Q in $\mathbf{R}^n$ is a d-dimensional face (for any d) of the usual integer (cubic) lattice cell complex in $\mathbf{R}^n$, which can be formally defined as

$$Q = I_1 \times I_2 \cdots \times I_n \subset \mathbf{R}^n$$

where $I_i = \{l_i\}$ or $I_i = [l_i, l_i + 1]$ and $l_i \in \mathbf{Z}$. The set of elementary cubes in $\mathbf{R}^n$ is denoted by $\mathcal{K}^n$. The *dimension* of Q is defined as

$$\dim Q := \operatorname{card}\{i \mid I_i = [l_i, l_i + 1]\}$$

and $\mathcal{K}^n_d$ indicates the set of d-dimensional elementary cubes in $\mathbf{R}^n$. Elements of $\mathcal{K}^n_n$ are called *full cubes*.

Let $\mathcal{X} \subset \mathcal{K}^n$, then its *geometric realization* is

$$|\mathcal{X}| := \bigcup \mathcal{X} \subset \mathbf{R}^n.$$

Consider finite sets of full cubes $\mathcal{X} \subset \mathcal{K}_n^n$ and $\mathcal{Y} \subset \mathcal{K}_m^m$. A *combinatorial multivalued map* is a multivalued map $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$. The *upper envelope* of $\mathcal{F}$ is the multivalued map $\lceil \mathcal{F} \rceil\colon |\mathcal{X}| \rightrightarrows |\mathcal{Y}|$ defined by

$$\lceil \mathcal{F} \rceil(x) = \bigcup \{ |\mathcal{F}(Q)| \mid x \in Q \in \mathcal{X} \} \subset |\mathcal{Y}|.$$

Observe that this provides us with a well defined procedure for passing from combinatorial data to topological information. To simplify the notation, we will implicitly define $F := \lceil \mathcal{F} \rceil$.

A set $X \subset \mathbf{R}^n$ is a *cubical set* if it is a finite union of elementary cubes. Note that a cubical set X is in fact a combinatorial object as it can be represented in a finite way by the set $\mathcal{X} \in \mathcal{K}^n$ such that $X = |\mathcal{X}|$. However, the representation of X is usually non-unique: Define $\mathcal{X}_{\max} := \{Q \in \mathcal{K}^n \mid Q \subset X\}$ and $\mathcal{X}_{\min} := \{Q \in \mathcal{X}_{\max} \mid \text{for every } R \in \mathcal{X}_{\max} \text{ if } Q \subset R \text{ then } Q = R\}$; then $X = |\mathcal{X}|$ for every $\mathcal{X} \subset \mathcal{K}^n$ such that $\mathcal{X}_{\min} \subset \mathcal{X} \subset \mathcal{X}_{\max}$. In the implementation of our algorithms we try and represent cubical sets as close to $\mathcal{X}_{\min}$ as possible. Since the technical complications arising from such optimization are inevitable, in the algorithms described in this paper we operate with cubical sets at the topological level, but the reader should keep in mind the fact that they are really combinatorial objects.

Because $\mathcal{F}$ is used to represent f, we are particularly interested in *full cubical sets*; that is, cubical sets of the form $X = |\mathcal{X}|$ where $\mathcal{X} \subset \mathcal{K}_n^n$. Observe that if X is a full cubical set, then there is a unique set of full cubes $\mathcal{X} \subset \mathcal{K}_n^n$ such that $X = |\mathcal{X}|$.

To simplify the notation we adopt the following convention. We use calligraphic letters to denote combinatorial objects and the corresponding capital letters to denote the corresponding topological objects. In particular, if a full cubical set in $\mathbf{R}^n$ is written using a capital letter, then the corresponding set of full cubes is denoted by the corresponding calligraphic letter.

An important notion used in the reduction algorithms is acyclicity. A topological set is called *acyclic* if its homology is isomorphic to the homology of a single-point space. Note that the acyclicity of set may depend on the ring of coefficients used to compute homology. A simple example is provided by the real projective plane.

Because of the intended applications we introduce two more concepts. A combinatorial multivalued map $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ is a *combinatorial representation* of a continuous map $f\colon X \to Y$ if f is a selector of F. It is *acyclic* if $F(x)$ is an acyclic set for each $x \in X$.

Assume $f\colon (X, A) \to (Y, B)$ is a continuous map of pairs and $X, A \subset \mathbf{R}^n$ and $Y, B \subset \mathbf{R}^m$ are full cubical sets. We are interested in an algorithm computing $f_*\colon H_*(X, A) \to H_*(Y, B)$. For this end we need to extend the concept of representation of a single valued map to the maps of pairs. We say that a combinatorial multivalued map $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ is a *representation* of $f\colon (X, A) \to (Y, B)$ if $\mathcal{F}$ is a representation of $f\colon X \to Y$ and $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$. (Note that if $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ is a combinatorial representation of $f\colon X \to Y$, then the condition $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ implies $f(A) \subset B$.) The reader may expect that given a representation $\mathcal{F}$ of $f\colon (X, A) \to (Y, B)$ we have $F(A) \subset B$, where $F = \lceil \mathcal{F} \rceil$. However, this is not true in general. In fact, as is indicated in Figure 6, in some cases there exist $x \in \partial A$ such that $F(x) \not\subset B$. Therefore, it is convenient to introduce another concept. A pair (F, G) of multivalued maps is a representation of $f\colon (X, A) \to (Y, B)$ if

$F\colon X \to Y$ is a representation of $f\colon X \to Y$ and $G\colon A \to B$ is a represenatation of $f|_A\colon A \to B$, and $G \subset F$. It is straightforward that if $\mathcal{F}$ is a combinatorial representation of $f\colon (X, A) \to (Y, B)$, then $(\lceil\mathcal{F}\rceil, \lceil\mathcal{F}|_{\mathcal{A}}\rceil)$ is a representation of f.

Observe that given a continuous map $f\colon X \to Y$ where X and Y are full cubical sets, finding a combinatorial representation $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ is a question of approximation. This is a topic in its own right (see [21], Th. 4.2, and the discussion in [23]) and is not the subject of this paper. Thus, we will limit ourselves to a few comments. The simplest approach to computing a rigorous enclosure is to use interval arithmetic [20] to evaluate the images of entire intervals or cubes by the map f. For simple nonlinearities more sophisticated approaches to obtaining bounds can also be used [5]. A more challenging example arises when f is the translation map of a continuous dynamical system induced by an ODE. In this setting one can use the method introduced in [12, 23, 33] (an implementation is available at [2]), as, for instance, was done in [22, 25, 26]. With this set of examples as justification our approach for the remainder of this paper is to assume that an appropriate combinatorial representation has been found.

The use of cubes, as opposed to simplicies, is contrary to the customary approach and thus deserves comment. At first it seems that the standard simplicial theory provides us with a good setting for algorithmic computation of maps in homology. Given a continuous map $f\colon |K| \to |L|$ of two simplicial complexes K and L, which satisfies the star condition one can construct a simplicial approximation of the map and from the simplicial approximation one can determine the map in homology. However, there are at least three problems with this approach.

- To verify the star condition one has to find good upper estimates of the images of the map on simplices. Unfortunately standard numerical algorithms for upper estimates of images are based on interval arithmetic, which leads to a significant overestimation when applied to simplices.
- In most cases a significant amount of subdivisions of K is needed in order to guarantee that the star condition is met even for simple maps. The large subdivision implies heavy numerical computations to verify the star condition. Additionally, we know of no practical a priori formula for determining what the optimal subdivision should be, which forces one to loop in the search of an optimal subdivision.
- Last but not least is the overhead caused by the necessity of matching the vertices in the star condition needed to define the simplicial approximation.

A possible alternative is to construct a covering $\mathcal{A} := \{A_w \mid w \in L\}$ of $|K|$ such that $A_w \subset f^{-1}(\operatorname{St} w)$, find the Lebesgue number, λ, of $\mathcal{A}$ and replace K with $\operatorname{sd}^N K$, where N is chosen so that each simplex in $\operatorname{sd}^N K$ has diameter less than $\lambda/2$. Unfortunately this approach appears to be even worse, because finding A_w requires the construction of approximations of $f^{-1}(\operatorname{St} w)$ from below, which is hard even for simple sets like balls or rectangles.

Of course these arguments do not imply that an algorithm based on simplicial homology is not possible, merely that we were not able to overcome these obstacles. However, we do believe that the approach presented in this paper, based on cubical homology and graphs of multivalued maps, when applied to maps available only through numerical computations is more natural and practical.

The final point which needs to be addressed is the justification for the development of a dimension independent algorithm. As was mentioned earlier, the origins of this work lie in the analysis of numerical and experimental data. In particular, the common strategy for these applications is to use the computer to identify an isolating neighborhood and compute its homology Conley index which involves computing the relative homology of a map (see [17, 14] for an introduction to this theory in the context of computations). For the earliest applications [15, 16, 19, 18] the computation of the homology map was greatly simplified by the fact that the maps of interest were defined on subsets of the plane and only the first homology groups were involved. This meant that the computation could be reduced to a question involving the connectedness of graphs. However, recent applications to infinite dimensional problems [4, 5] require that these computations be performed in higher dimensional spaces (dimension 6 for the specific example discussed at the end of this introduction). Furthermore, the higher homology groups come into play. At the moment, it appears that the techniques described in this paper are essential to these applications in the sense that they can handle relatively high dimensional data in an efficient manner both in time and memory.

Our main result is Algorithm 5.1 (see Section 5) whose validity is justified by the following theorem.

Theorem 1.1 *Let $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$ be full cubical sets. Let the combinatorial multivalued map $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ be a representation of*

$$f\colon (X, A) \to (Y, B).$$

*Assume that $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ and that both $\mathcal{F}$ and $\mathcal{F}|_{\mathcal{A}}$ are acyclic. Then the homomorphism returned by Algorithm 5.1 invoked with $\mathcal{F}$, $\mathcal{A}$, $\mathcal{B}$ and "incl" set to **false** coincides with $f_*\colon H_*(X, A) \to H_*(Y, B)$ in the sense that the domain D of this homomorphism is isomorphic to $H_*(X, A)$, the codomain C of it is isomorphic to $H_*(Y, B)$, and the following diagram, in which φ denotes the returned homomorphism, commutes*

$$\begin{array}{ccc} D & \xrightarrow{\varphi} & C \\ \downarrow \simeq & & \downarrow \simeq \\ H_*(X, A) & \xrightarrow{f_*} & H_*(Y, B) \end{array}$$

*Moreover, if $\mathcal{X} \subset \mathcal{Y}$, $\mathcal{A} \subset \mathcal{B}$ and the inclusion $i\colon (X, A) \hookrightarrow (Y, B)$ induces an isomorphism in homology, then the homomorphism returned by Algorithm 5.1 invoked with $\mathcal{F}$, $\mathcal{A}$, $\mathcal{B}$ and "incl" set to **true** coincides with the endomorphism $(i_*)^{-1} \circ f_*\colon H_*(X, A) \to H_*(X, A)$.*

While necessary, the validity of an algorithm is not sufficient. To be of practical value it must also be efficient. Though we will not present a formal analysis of the complexity, our experience suggests that the two predominant factors in the cost of computing homology are the number and dimensions of the elements of $\mathcal{X}$ and $\mathcal{Y}$. For this reason much of the algorithm focuses on reducing theses quantities before computing homology. Since $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$, we cannot manipulate elements of $\mathcal{X}$ and $\mathcal{Y}$ in a completely independent manner. Thus, we have adopted the following strategy modelled on [9, 7, 8] which allows us to simultaneously keep track of the modifications to $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{F}$.

Given a continuous map $f: X \to Y$ one always has the commutative diagram

$$\begin{array}{ccc} & & \Gamma_f \\ & \nearrow \iota & \downarrow q \\ X & \xrightarrow{f} & Y \end{array} \tag{1}$$

where $\Gamma_f := \{(x, f(x)) \mid x \in X\} \subset X \times Y$ is the graph of f, ι is the embedding map $\iota(x) = (x, f(x))$, and q is the projection onto Y. Observe that ι is a homeomorphism whose inverse is the projection $p: \Gamma_f \to X$. Thus, $f = q \circ p^{-1}$ and in particular $f_* = q_* \circ (p_*)^{-1}$; that is the homology map of f can be computed in terms of the homology maps of two projections.

This same idea carries over to the multivalued setting. More precisely, let $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$ be an acyclic combinatorial multivalued represtentation of $f: X \to Y$. Then we can construct a corresponding diagram

$$\begin{array}{ccc} & & \Gamma_F \\ & \swarrow p & \downarrow q \\ X & \overset{F}{\rightrightarrows} & Y \end{array} \tag{2}$$

where $\Gamma_F := \{(x, y) \mid x \in X,\ y \in F(x)\} \subset X \times Y$ is the graph of F. Of course, in this case the projection p may not be invertible. However, because F is acyclic valued, p_* is an isomorphism (see Proposition 2.4) and hence $(p_*)^{-1}$ is well defined. In particular, as we will show, $f_* = q_* \circ (p_*)^{-1}$.

Since our goal is to compute $f_*: H_*(X, A) \to H_*(Y, B)$, we have two related diagrams,

$$\begin{array}{ccc} & & \Gamma_F \\ & \swarrow p_F & \downarrow q_F \\ X & \overset{F}{\rightrightarrows} & Y \end{array} \qquad \begin{array}{ccc} & & \Gamma_G \\ & \swarrow p_G & \downarrow q_G \\ A & \overset{G}{\rightrightarrows} & B \end{array} \tag{3}$$

that need to be considered, where $G = \lceil \mathcal{F}|_A \rceil$.

Because p and q are simple projection maps, the computational cost of this approach to computing homology is determined mainly by the number of elementary cubes in $\Gamma_F \setminus \Gamma_G$. This is due to the fact that the elementary cubes in Γ_G do not become part of the relative chain complex of the graph of (F, G) which is used for the homology computation (consult Proposition 3.2 and the definitions that precede it for details).

As was suggested earlier, the efficiency of Algorithm 5.1 arises from preprocessing the sets of elementary cubes before proceeding with algebraic computations. This is done using a variety of other algorithms three of which we briefly mention here.

The first, `reduceF` (see Algorithm 4.3), is used to reduce the number of elements of $\mathcal{X}$ that need to be considered. More precisely, `reduceF` takes as input the sets $\mathcal{X}$ and $\mathcal{A}$ and produces sets $\widetilde{\mathcal{X}} \subset \mathcal{X}$ and $\widetilde{\mathcal{A}} \subset \mathcal{A}$ with the property that $H_*(\widetilde{X}, \widetilde{A}) \cong H_*(X, A)$ and both $\widetilde{\mathcal{F}} := \mathcal{F}|_{\widetilde{\mathcal{X}}}$ and $\widetilde{\mathcal{F}}|_{\widetilde{\mathcal{A}}}$ are acyclic.

Another way to simplify the computations is to enlarge the set $\mathcal{A}$ since its content is in essence ignored during the homology computation. This is done using `expandF` which produces sets $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{B}}$ satisfying $\mathcal{A} \subset \widetilde{\mathcal{A}} \subset \mathcal{X}$ and $\mathcal{B} \subset \widetilde{\mathcal{B}} \subset \mathcal{Y}$ such that $\mathcal{F}|_{\widetilde{\mathcal{A}}}$ is still acyclic and $\mathcal{F}(\widetilde{\mathcal{A}}) \subset \widetilde{\mathcal{B}}$.

The final algorithm which we wish to mention here is `collapse` which whenever possible eliminates the highest dimensional cubes. The importance of this is that in general the cost of homology computations increases rapidly as a function of the dimension of the cubes and by construction Γ_F consists of $(n+m)$-dimensional cubes. However, it is intuitively clear that on the level of homology all the relevant information of the map should be carried by a collection of n-dimensional cubes in Γ_F. `collapse` is used to perform a reduction to such a set of elements.

To put the previous discussion into perspective, let us consider an essential nontrivial application of the techniques introduced in this paper. In [5], Day, Junge, and one of the authors of this paper, reduce the problem of obtaining rigorous results concerning the dynamics of an infinite dimensional map to the computation of the homology of continuous maps $f\colon (X, A) \to (Y, B)$. The justification of this reduction and the details concerning the dynamics can be found in [5]. For the purpose of this paper it is sufficient to remark that the homology computations were performed using combinatorial representations $\mathcal{F}\colon (\mathcal{X}, \mathcal{A}) \rightrightarrows (\mathcal{Y}, \mathcal{B})$, where $\mathcal{Y} := \mathcal{X} \cup \mathcal{F}(\mathcal{X})$, $\mathcal{B} := \mathcal{A} \cup \mathcal{F}(\mathcal{A})$, $X = |\mathcal{X}|$, $Y = |\mathcal{Y}|$, etc.

Algorithm 5.1 is used to compute the following endomorphism induced in homology: $(i_*)^{-1} \circ f_*\colon H_*(X, A) \to H_*(X, A)$, where $i\colon (X, A) \hookrightarrow (Y, B)$ is the inclusion map. Obviously, this result is only valid if i induces an isomorphism in homology, and this condition is verified during the homology computation. Let us now explain step by step how the actual program available in [27] proceeds. Note that some additional actions not listed in Algorithm 5.1 are undertaken by the program, which is motivated mainly by efficiency reasons. Moreover, the map in question has convex values which implies that all its restrictions are acyclic, and therefore the program skips some time-consuming verifications. The numbers we quote correspond to Example 3 in [5], but in the other two cases the steps undertaken by the program are essentially the same.

The program first reads the sets $\mathcal{X}$ and $\mathcal{A}$ from the initial data files (10,330 and 6,683 full cubes in $\mathbf{R}^6$) and stores the disjoint sets $\mathcal{X} \setminus \mathcal{A}$ and $\mathcal{A}$ in the memory. The first reduction step applied to the data is the removal of cubes from $\mathcal{A}$ which do not have neighbors in $\mathcal{X} \setminus \mathcal{A}$ (part of Algorithm 4.1). Then the program reads $\mathcal{Y}$ and $\mathcal{B}$ from the disk (25,737 and 22,090 cubes, respectively), stores $\mathcal{Y} \setminus \mathcal{B}$ and $\mathcal{B}$ in the memory, and verifies (just in case) that $\mathcal{X} \setminus \mathcal{A} \subset \mathcal{Y}$ and $\mathcal{A} \subset \mathcal{B}$ to make sure that the inclusion map $i\colon (X, A) \hookrightarrow (Y, B)$ is well defined. Then the program reduces $(\mathcal{X}, \mathcal{A})$ with the `reduce` procedure (Algorithm 4.1). This reduction decreases the data very significantly, with only 699 cubes remaining. At this point the program considers the map $\mathcal{F}$, but it only reads its restriction to $\mathcal{X} \setminus \mathcal{A}$ for the moment. It verifies that $\mathcal{F}(\mathcal{X} \setminus \mathcal{A}) \subset \mathcal{Y}$ (just in case) and runs `expandF` (Algorithm 4.7) followed by `reduce` (Algorithm 4.1); this step leaves 332 cubes in $\mathcal{X}$, 197 of which are in $\mathcal{A}$, and adds 2,222 cubes to $\mathcal{B}$. Now the program considers the map $\mathcal{F}$ on the entire set $\mathcal{X}$ and reads all the necessary data from the disk. It verifies that $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ to make sure the data is correct. Then it applies `expandA` (Algorithm 4.5) to $(\mathcal{Y}, \mathcal{B})$, which increases $\mathcal{B}$ by 1,091 cubes. This step is followed by applying the `reduce` procedure (Algorithm 4.1) to $(\mathcal{Y}, \mathcal{B})$ in such a way that the cubes in $\mathcal{F}(\mathcal{X}) \cup \mathcal{X} \subset \mathcal{Y}$ (7,588 cubes) are left intact. In our example this decreases the number of cubes in $\mathcal{Y}$ to 7,610.

At this point the full cubes are transformed into elementary cubes (called ‘cells’ for short), because a different data type is used to represent them, and the procedure `collapse` (Algorithm 4.9) is applied to $(\mathcal{X}, \mathcal{A})$. It leaves only 9,103 elementary cubes in $\mathcal{X} \setminus \mathcal{A}$. There are still 66,757 elementary cubes left in $\mathcal{A}$ which will be used in the next step, and the dimension of $\mathcal{X} \setminus \mathcal{A}$ decreases from 6 to 2. Now `reducemap` (Algorithm 4.12) is run to determine the images of the cells in $\mathcal{X}$ and in $\mathcal{A}$ by $|\mathcal{F}|$ and to `collapse` them to lower-dimensional cubical sets if possible. This results in the graph of $\widetilde{F}$, a replacement for $F|_{X \setminus A}$, consisting of 217,929 cells. The last geometric reduction is eventually applied to (Y, B). The full cubes that represent them are transformed into elementary cubes, and `collapse` is applied in such a way that $\widetilde{F}(X \setminus A)$ is preserved. This reduces the number of elementary cubes in $\mathcal{Y} \setminus \mathcal{B}$ which are relevant for homology computation to just 5,945, and the dimension decreases to 3.

To enter the last stage of the computations, the geometric sets are transformed into algebraic data. The elementary cubes in $\Gamma_{\widetilde{F}}$ and $Y \setminus B$ are used as generators of the appropriate chain complexes, and the chain maps corresponding to the projection $q \colon (\Gamma_{\widetilde{F}}, \Gamma_G) \to (Y, B)$ and to the composition of the projection $p \colon (\Gamma_{\widetilde{F}}, \Gamma_G) \to (X, A)$ with the inclusion $i \colon (X, A) \hookrightarrow (Y, B)$ are created, as explained in Section 3. The algebraic homology computation over $\mathbf{Z}$ reveals that $H_1(\Gamma_{\widetilde{F}}, \Gamma_G) \simeq H_1(Y, B) \simeq \mathbf{Z}$, $H_2(\Gamma_{\widetilde{F}}, \Gamma_G) \simeq H_2(Y, B) \simeq \mathbf{Z}^{18}$, and all the remaining homology groups are trivial. Some generators of $H_2(\Gamma_{\widetilde{F}}, \Gamma_G)$ and $H_2(Y, B)$ are fixed, and the matrices M_* and N_* of the homomorphisms induced by q and $i \circ p$ on the first and second homology groups with respect to these generators are as follows:

$$M_1 = \begin{bmatrix} 0 \end{bmatrix}, \qquad N_1 = \begin{bmatrix} 1 \end{bmatrix},$$

$$M_2 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix},$$

$$N_2 = \left[\begin{array}{cccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right].$$

Since both N_1 and N_2 are invertible, it follows that the inclusion $i\colon (X,A) \hookrightarrow (Y,B)$ induces an isomorphism in homology, and therefore $M_* N_*^{-1}$ is in fact the matrix of $(i_*)^{-1} \circ f_*$ with respect to some generators of $H_*(X,A)$. All these computations take about 68 minutes on a PC with a 2.4 GHz processor, and use almost 100 MB of memory.

The outline of this paper is as follows. In Section 2 we discuss the class of multivalued maps that are used for the homology computations. Although we are working in a different context, the reasoning is motivated by Górniewicz [7]. We also present Corollary 2.6 which guarentees that computing the homology map of an appropriate multivalued function produces the homology map of its continuous selector.

Section 3 recalls the cubical theory developed in [11]. In particular, it is indicated how given a combinatorial multivalued map one can construct a chain map from which the homology map can be computed.

Section 4 describes the reduction algorithms indicated above. As was mentioned above the purpose of these algorithms is to preprocess the data so as to minimize the cost of the homology computations. As such they are essential elements of Algorithm 5.1. However, for the sake of continuity of presentation we delay presenting the proofs of their validity to Section 7.

In Section 5 we state Algorithm 5.1 and prove Theorem 1.1. In Section 6 we present several examples indicating the applicability of this method.

2 Multivalued maps

As was indicated in the introduction, in this section we delve into the class of multivalued maps used for computing the homology of continuous functions. In particular, we define homomorphisms induced in homology by such maps. We begin our discussion on a fairly general level; postponing to the next section the restriction to the setting of cubical complexes.

Definition 2.1 Let $X \subset \mathbf{R}^n$ and $Y \subset \mathbf{R}^m$. A continuous map $f: X \to Y$ is a *Vietoris map* if the following conditions are satisfied:

(i) f is proper, that is, $f^{-1}(C)$ is compact for every compact set $C \subset Y$,

(ii) $f^{-1}(y)$ is acyclic for every $y \in Y$.

Since we are going to restrict our attention to cubical sets, X and Y will be assumed compact thus making the map f automatically proper. Therefore, in what follows we often simplify matters by assuming that all the sets we consider are compact. Moreover, the restriction to cubical sets will guarantee that we can use the cubical homology theory (as in [11]) without loss of generality.

The following two theorems allow us to use graph projections to compute the homology of multivalued maps. The first is a special case of [30, Theorem 6.9.15] and the second is a straightforward extension.

Theorem 2.2 (Vietoris-Begle Mapping Theorem) *Let X and Y be compact. If $f: X \to Y$ is a Vietoris map, then the induced map $f_*: H_*(X) \to H_*(Y)$ is an isomorphism.*

Proposition 2.3 *Let $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$ be compact sets. If $f: (X, A) \to (Y, B)$ is a continuous map such that both $f: X \to Y$ and its restriction $f|_A: A \to B$ are Vietoris maps, then $f_*: H_*(X, A) \to H_*(Y, B)$ is an isomorphism.*

Proof: Since f is a Vietoris map, the induced homomorphism $f_*: H_*(X) \to H_*(Y)$ is an isomorphism. For the same reason, $(f|_A)_*: H_*(A) \to H_*(B)$ is also an isomorphism. Applying the five lemma to the following commutative diagram whose rows are the exact sequences for the pairs (X, A) and (Y, B):

$$\begin{array}{ccccccccccccc}
\cdots & \to & H_k(A) & \to & H_k(X) & \to & H_k(X,A) & \to & H_{k-1}(A) & \to & H_{k-1}(X) & \to & \cdots \\
 & & \downarrow{\scriptstyle (f|_A)_k} & & \downarrow{\scriptstyle f_k} & & \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle (f|_A)_{k-1}} & & \downarrow{\scriptstyle f_{k-1}} & & \\
\cdots & \to & H_k(B) & \to & H_k(Y) & \to & H_k(Y,B) & \to & H_{k-1}(B) & \to & H_{k-1}(Y) & \to & \cdots
\end{array}$$

we conclude that $f_*: H_*(X, A) \to H_*(Y, B)$ is an isomorphism. ■

Let us now introduce the notion of upper semi-continuity for multivalued maps which corresponds to the notion of continuity for (ordinary) maps in our case. Note that there also exists the notion of lower semi-continuity (see [11] for details), but we will not use it here.

Consider a multivalued map $F: X \rightrightarrows Y$. It is *upper semi-continuous* if for every $x \in X$ the set $F(x)$ is compact and for every open set $V \subset Y$ the set $F^{-1}(V) := \{x \in X \mid F(x) \subset V\}$ is an open subset of X. By [7, Proposition 1.2] if $F: X \rightrightarrows Y$ is upper semi-continuous then the image $F(A)$ of every compact set $A \subset X$ under F is compact.

A multivalued map $G: X \rightrightarrows Y$ is a *submap* of a multivalued map $F: X \rightrightarrows Y$, if $G(x) \subset F(x)$ for all $x \in X$. Observe that a selector f of F is a particular example of a submap.

The following proposition indicates how we will make use of Vietoris maps in the context of upper semi-continuous multivalued maps. Recall that a multivalued map $F: X \rightrightarrows Y$ is *acyclic* if $F(x)$ is acyclic for every $x \in X$.

Proposition 2.4 *Consider compact sets $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$. Let $F: X \rightrightarrows Y$ and let $G: A \rightrightarrows B$ be a submap of $F|_A$. If F and G are acyclic upper semi-continuous maps, then the natural projection $p: (\Gamma_F, \Gamma_G) \to (X, A)$ induces an isomorphism in homology.*

Proof: Since the image of a compact set under an upper semi-continuous map is compact, the pre-image of every compact set by each of the projections $p_F: \Gamma_F \to X$ and $p_G: \Gamma_G \to A$ is compact. This property combined with the acyclicity of F and G implies that p_F and p_G are Vietoris maps. Moreover, p_G is a restriction of p_F. Proposition 2.3 completes the proof. ■

We define the map induced in homology by a pair of multivalued maps (F, G) satisfying the assumptions of Proposition 2.4 in the following way:

$$(F, G)_* := q_* \circ (p_*)^{-1}: H_*(X, A) \to H_*(Y, B),$$

where q is the natural projection $(\Gamma_F, \Gamma_G) \to (Y, B)$. Note that by Proposition 2.4, p induces an isomorphism in homology, so this map is well-defined. Moreover, it is easy to see that if $F = f$ (that is, F is a single-valued map), then $(F, G)_* = (f, f|_A)_* = f_*$.

Proposition 2.5 *Consider compact sets $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$. Let $F: X \rightrightarrows Y$ and G be a submap of $F|_A: A \rightrightarrows B$. Assume that F and G are acyclic upper semi-continuous maps. If $\widetilde{F}$ and $\widetilde{G}$ are acylic upper semi-continuous submaps of F and G, respectively, and $\widetilde{G}$ is a submap of $\widetilde{F}$, then $(F, G)_* = (\widetilde{F}, \widetilde{G})_*$.*

Proof: Denote the natural projections for the map F by p, q, and for the map $\widetilde{F}$ by $\widetilde{p}, \widetilde{q}$. Consider the following commutative diagram:

$$\begin{array}{ccccc} & & (\Gamma_F, \Gamma_G) & & \\ & \swarrow p & & \searrow q & \\ (X, A) & & \uparrow \iota & & (Y, B) \\ & \nwarrow \widetilde{p} & & \nearrow \widetilde{q} & \\ & & (\Gamma_{\widetilde{F}}, \Gamma_{\widetilde{G}}) & & \end{array}$$

where $\iota: (\Gamma_{\widetilde{F}}, \Gamma_{\widetilde{G}}) \hookrightarrow (\Gamma_F, \Gamma_G)$ is the inclusion. Apply the homology functor to this diagram and notice that

$$(F, G)_* = q_* \circ (p_*)^{-1} = q_* \circ \iota_* \circ (\widetilde{p}_*)^{-1} = \widetilde{q}_* \circ (\widetilde{p}_*)^{-1} = (\widetilde{F}, \widetilde{G})_*.$$

■

Corollary 2.6 *Let F and G be as in Proposition 2.4. Let $f: (X, A) \to (Y, B)$ be a continuous map. If f is a selector of F and $f|_A$ is a selector of G, then $(F, G)_* = f_*$.*

3 Representable sets and maps

In this section we return to the discussion of the combinatorial representation of sets and maps in terms of elementary cubes. We begin by introducing some additional terminology and then turn to the relation between these combinatorial objects and the topological constructs of the previous section. We conclude with a description of the formulas for the chain maps of the graph projections.

If $P \subset Q \subset \mathbf{R}^n$ are two elementary cubes, then P is a *face* of Q. It is a *proper face* of Q if, in addition, $P \neq Q$. Given an elementary cube Q, define

$$\overset{\circ}{Q} := Q \setminus \bigcup \{P \mid P \text{ is a proper face of } Q\}.$$

Observe that if P and Q are elementary cubes such that $P \neq Q$, then $\overset{\circ}{P} \cap \overset{\circ}{Q} = \emptyset$. Since by definition every cubical set is the finite union of elementary cubes, it is compact and, moreover, is a disjoint union of $\overset{\circ}{Q}$ over all the elementary cubes Q it contains.

A multivalued map $F: X \rightrightarrows Y$, where $X \subset \mathbf{R}^n$ and $Y \subset \mathbf{R}^m$ are cubical sets, is called a *cubical multivalued map* if Γ_F is a cubical set in $\mathbf{R}^{n+m}$. It follows that $F(x)$ is a cubical set in $\mathbf{R}^m$ for every $x \in X$ and F is constant on $\overset{\circ}{Q}$ for every elementary cube $Q \subset X$. Note that since Γ_F is compact, F is upper semi-continuous.

We would like to stress that a cubical multivalued map is in fact a combinatorial object and can be represented in a finite way by the set of assignments $\left\{\overset{\circ}{Q} \mapsto F(\overset{\circ}{Q}) \mid \overset{\circ}{Q} \subset X\right\}$. In particular, in order to define such a map in an algorithm, it is enough to define each $F(\overset{\circ}{Q})$, and this is done in Algorithm 4.12, although the assignment "$F(\overset{\circ}{Q}) := D$" may look strange at first glance.

As an immediate consequence of Corollary 2.6 we have the following

Theorem 3.1 *Let $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}^n_n$, $\mathcal{B} \subset \mathcal{Y} \subset \mathcal{K}^m_m$, and $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$. Assume that $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ and $\mathcal{F}$ is a representation of a continuous map $f: X \to Y$ (note that then $f(A) \subset B$). Let $\mathcal{G} := \mathcal{F}|_{\mathcal{A}}$. If $\mathcal{F}$ and $\mathcal{G}$ are acyclic, then $(F, G)_* = f_*: H_*(X, A) \to H_*(Y, B)$.*

Computing the homology of f with the use of a pair of multivalued maps (F, G) instead of using $F: (X, A) \rightrightarrows (Y, B)$ directly (as is suggested in [7]) may, at first glance, appear somewhat artificial. However, it should be kept in mind that the actual computations are performed using $\mathcal{F}$ and by definition $F = \lceil \mathcal{F} \rceil$. Because of this $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ does *not* imply that $F(A) \subset B$ (see $F(x)$ in Figure 6 for a counterexample). In fact, one can check that $F(A) \subset B$ if and only if $\mathcal{F}(o_{\mathcal{X}}(\mathcal{A})) \subset \mathcal{B}$, where $o_{\mathcal{X}}(\mathcal{A})$ is the set $\mathcal{A}$ together with all its neighbors in $\mathcal{X}$, that is,

$$o_{\mathcal{X}}(\mathcal{A}) := \{Q \in \mathcal{X} \mid Q \cap P \neq \emptyset \text{ for some } P \in \mathcal{A}\}.$$

Note that even the identity map $\mathcal{I}: \mathcal{X} \rightrightarrows \mathcal{X}$ given by $\mathcal{I}(Q) = \{Q\}$ does not in general satisfy this assumption.

We would also like to explain why we assume that *both* maps $\mathcal{F}$ and $\mathcal{F}|_{\mathcal{A}}$ are acyclic in Theorem 3.1. The reason is that a restriction of an acyclic combinatorial multivalued map need not be acyclic, as one of the examples in [27] proves.

In the remainder of this section we introduce explicit formulas for the chain maps of the projections used to compute the homomorphism induced in homology by a pair of multivalued maps.

Given a pair of cubical sets (K, L) let $C(K, L)$ denote the associated cubical chain complex. This is a free chain complex whose generators correspond to the elementary cubes $Q \subset K$ such that $Q \not\subset L$. The generator corresponding to Q is denoted by $\widehat{Q}$. See [11] for further details.

Let $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$ be cubical sets. Let $F: X \rightrightarrows Y$ and $G: A \rightrightarrows B$ be acyclic cubical multivalued maps such that G is a submap of $F|_A$. The chain map $\varphi: C(\Gamma_F, \Gamma_G) \to C(X, A)$ of the projection $p: (\Gamma_F, \Gamma_G) \to (X, A)$ is defined on generators $\widehat{Q}$ of C_k in the following way. If the corresponding cube Q is reduced by the projection map (i.e., $\dim p(Q) < \dim Q$), then $\widehat{Q}$ is mapped to zero. Otherwise it is mapped to the generator of $C_k(X, A)$ corresponding to $p(Q)$ (which still can be zero if $p(Q) \subset A$). Formally, this definition can be written as

$$\varphi_k(\widehat{Q}) = \begin{cases} \widehat{p(Q)} & \text{if } \widehat{p(Q)} \in C_k(X, A), \\ 0 & \text{otherwise.} \end{cases}$$

The chain map $\psi: C(\Gamma_F, \Gamma_G) \to C(Y, B)$ of the projection $q: (\Gamma_F, \Gamma_G) \to (Y, B)$ is defined similarly.

Proposition 3.2 (see [11]) *The homomorphisms induced in homology by the chain maps φ and ψ defined above coincide with the homomorphisms induced in homology by the projections $p: (\Gamma_F, \Gamma_G) \to (X, A)$ and $q: (\Gamma_F, \Gamma_G) \to (Y, B)$, respectively.*

Corollary 3.3 *If $f: (X, A) \to (Y, B)$ is a selector of F and $f|_A$ is a selector of G, then*

$$f_* = (F, G)_* = (\psi)_* \circ \big((\varphi)_*\big)^{-1}.$$

Based on the discussion above, in Section 5 we will assume that we have the following algorithms which compute the chain maps of the projections p and q, respectively, and whose details are left to the reader:

Algorithm 3.4 Chain Map of the Projection p
function `proj_p` (F, G: cubical multivalued map; X, A: cubical set):
chain map;

Algorithm 3.5 Chain Map of the Projection q
function `proj_q` (F, G: cubical multivalued map; Y, B: cubical set):
chain map;

For the homology computation of the chain maps φ and ψ of the projections $(\Gamma_F, \Gamma_G) \to (X, A)$ and $(\Gamma_F, \Gamma_G) \to (Y, B)$, respectively, one can use the algorithm introduced in [10] or its generalization [24]. Our interface to this algorithm is as follows:

Algorithm 3.6 Homology of Chain Maps
function `homchain` (Γ_F, Γ_G, X, A, Y, B: cubical set; φ, ψ: chain map):
(φ_*, ψ_*: homomorphism);

At this point we are able to compute $f_*: H_*(X, A) \to H_*(Y, B)$. Unfortunately, the method introduced so far is of limited use in practice, since the amount of algebraic data to process can be extremely large due to the size of the chain complex of (Γ_F, Γ_G), as illustrated in Section 6. Therefore, it is necessary to replace the pair (Γ_F, Γ_G) with a smaller one. For this end, we decrease in size the domain and codomain of f and we construct a possibly small cubical submap of F such that the homomorphism induced in homology after the reduction is the same as for the original map. Effective algorithms which we use for this kind of the reduction are discussed in Section 4.

4 Geometric cubical reduction

In this section we introduce algorithms for the reduction of a pair of cubical sets (X, A) in such a way that the homology of (X, A) is preserved. The reduction is done either on the level of full cubical sets or cubical sets. We also introduce an algorithm for the construction of a possibly small cubical submap of a cubical multivalued map. For the sake of clarity of presentation, proofs of the results are postponed to Section 7.

The first algorithm in this section removes cubes from $\mathcal{X}$ whenever it does not affect the homology of (X, A). Moreover, it does not remove cubes which belong to $\mathcal{S}$ (this feature is used in Algorithms 4.7 and 5.1).

Algorithm 4.1 Reduce Cubes
procedure `reduce` (**var** $\mathcal{X}$, $\mathcal{A}$: finite subset of $\mathcal{K}_n^n$; $\mathcal{S}$: finite subset of $\mathcal{K}_n^n$);
begin
  while exists $Q \in \mathcal{X} \setminus \mathcal{S}$
    such that $\big(Q \notin \mathcal{A}$ **and** $Q \cap |\mathcal{X} \setminus \{Q\}|$ is acyclic $\big)$
    or $\big(Q \in \mathcal{A}$ **and** $Q \cap |\mathcal{A} \setminus \{Q\}|$ is acyclic
      and $Q \cap |\mathcal{X} \setminus \{Q\}|$ is acyclic $\big)$
    or $\big(Q \in \mathcal{A}$ **and** $Q \cap |\mathcal{X} \setminus \mathcal{A}| = \emptyset \big)$ **do**
  begin
    $\mathcal{X} := \mathcal{X} \setminus \{Q\}$;
    $\mathcal{A} := \mathcal{A} \setminus \{Q\}$
  end
end.

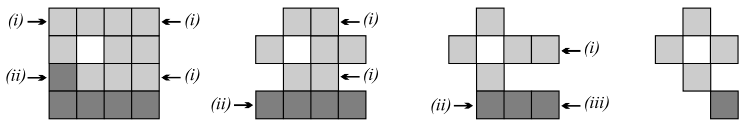

Figure 1: Reduction with Algorithm 4.1. Cubes in $\mathcal{A}$ are dark-grey, cubes in $\mathcal{X} \setminus \mathcal{A}$ are light-gray, $\mathcal{S} = \emptyset$. Cubes selected for removal are indicated with arrows and labeled with the corresponding condition from Lemma 7.2

Proposition 4.2 *Consider the finite subsets $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$. Let $\mathcal{S} \subset \mathcal{X}$. Then Algorithm 4.1 transforms $(\mathcal{X}, \mathcal{A})$ in a finite number of steps into the pair $(\widetilde{\mathcal{X}}, \widetilde{\mathcal{A}})$ such that the inclusion $(\widetilde{X}, \widetilde{A}) \hookrightarrow (X, A)$ induces an isomorphism in homology. Moreover, $\mathcal{S} \subset \widetilde{\mathcal{X}}$.*

Let $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$ be an acyclic combinatorial multivalued map. Assume that $\mathcal{A} \subset \mathcal{X}$ and $\mathcal{F}|_{\mathcal{A}}$ is also acyclic. In order to make sure that the restrictions of $\mathcal{F}$ to $\mathcal{X} \setminus \{Q\}$ as well as $\mathcal{A} \setminus \{Q\}$ are acyclic at each step, we propose the following, enhanced version of Algorithm 4.1.

Algorithm 4.3 Reduce Mutlivalued Map
procedure `reduceF` (**var** $\mathcal{X}$, $\mathcal{A}$: finite subset of $\mathcal{K}_n^n$;
    $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$: combinatorial multivalued map);
begin
    while exists $Q \in \mathcal{X}$
        such that $\Big[$ $\big($ $Q \notin \mathcal{A}$ **and** $Q \cap |\mathcal{X} \setminus \{Q\}|$ is acyclic $\big)$
            or $\big($ $Q \in \mathcal{A}$ **and** $Q \cap |\mathcal{A} \setminus \{Q\}|$ is acyclic
                and $Q \cap |\mathcal{X} \setminus \{Q\}|$ is acyclic $\big)$
            or $\big($ $Q \in \mathcal{A}$ **and** $Q \cap |\mathcal{X} \setminus \mathcal{A}| = \emptyset$ $\big)$ $\Big]$
        and $\big[$ **for each** proper face P of Q
            $\big($ the set $\bigcup \{|\mathcal{F}(R)| \mid R \in \mathcal{X}, R \neq Q, P \subset R\}$ is acyclic
            and if $P \subset |\mathcal{A}|$ then $\bigcup \{|\mathcal{F}(R)| \mid R \in \mathcal{A}, R \neq Q, P \subset R\}$
            is also acyclic $\big)$ $\big]$ **do**
    begin
        $\mathcal{X} := \mathcal{X} \setminus \{Q\}$;
        $\mathcal{A} := \mathcal{A} \setminus \{Q\}$
    end
end.

Proposition 4.4 *Let $\mathcal{X}$ and $\mathcal{A}$ be finite subsets of $\mathcal{K}_n^n$ such that $\mathcal{A} \subset \mathcal{X}$. Then Algorithm 4.3 transforms $(\mathcal{X}, \mathcal{A})$ in a finite number of steps into the pair $(\widetilde{\mathcal{X}}, \widetilde{\mathcal{A}})$ such that the inclusion $(\widetilde{X}, \widetilde{A}) \hookrightarrow (X, A)$ induces an isomorphism in homology. Moreover, if $\mathcal{F}$ and $\mathcal{F}|_{\mathcal{A}}$ are acyclic then so are $\mathcal{F}|_{\widetilde{\mathcal{X}}}$ and $\mathcal{F}|_{\widetilde{\mathcal{A}}}$.*

The following algorithm increases the set $\mathcal{A}$ within $\mathcal{X}$ in such a way that this does not change the homology of (X, A).

Algorithm 4.5 Expand Relative Set
procedure `expandA` ($\mathcal{X}$: finite subset of $\mathcal{K}_n^n$, **var** $\mathcal{A}$: finite subset of $\mathcal{K}_n^n$);
begin
    while exists $Q \in \mathcal{X} \setminus \mathcal{A}$ **such that** $Q \cap |\mathcal{A}|$ is acyclic **do**
        $\mathcal{A} := \mathcal{A} \cup \{Q\}$
end.

Proposition 4.6 *Let $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$. Then Algorithm 4.5 transforms $(\mathcal{X}, \mathcal{A})$ in a finite number of steps into the pair $(\mathcal{X}, \widetilde{\mathcal{A}})$ such that the inclusion $(X, A) \hookrightarrow (X, \widetilde{A})$ induces an isomorphism in homology.*

If a combinatorial multivalued map $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$ is given and $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$, then after obtaining the pair $(\mathcal{X}, \widetilde{\mathcal{A}})$ from $(\mathcal{X}, \mathcal{A})$ with Algorithm 4.5, it can turn out that the inclusion $\mathcal{F}(\widetilde{\mathcal{A}}) \subset \mathcal{B}$ is not valid. Therefore, whenever a cube Q is added to $\mathcal{A}$, one must also modify the set $\mathcal{B}$ so that the inclusion $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ is preserved and the homology of (Y, B) remains unchanged. The latter holds true, for example, if $\mathcal{B} \cup \mathcal{F}(Q)$ can be reduced to $\mathcal{B}$ with Algorithm 4.1 (note that this is not an "if and only if" condition). Moreover, like in Algorithm 4.3, we must be cautious not to spoil the acyclicity of $\mathcal{F}|_{\mathcal{A}}$. With this in mind, we propose the following modification of Algorithm 4.5:

Algorithm 4.7 Expand Relative Part of Map
procedure `expandF` ($\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$: combinatorial multivalued map;
  var $\mathcal{A}$: finite subset of $\mathcal{K}_n^n$; **var** $\mathcal{B}$: finite subset of $\mathcal{K}_m^m$);
begin
  while exists $Q \in \mathcal{X} \setminus \mathcal{A}$ **such that** $Q \cap |\mathcal{A}|$ is acyclic
    and `reduce` ($\mathcal{B} \cup \mathcal{F}(Q)$, $\emptyset$, $\mathcal{B}$) $= \mathcal{B}$
    and for each face $P \subset |\mathcal{A}|$ of Q the set
    $\bigcup\{|\mathcal{F}(R)| \mid R \in \mathcal{A}, R \neq Q, P \subset R\}$ is acyclic **do**
  begin
    $\mathcal{A} := \mathcal{A} \cup \{Q\}$;
    $\mathcal{B} := \mathcal{B} \cup \mathcal{F}(Q)$
  end
end.

Proposition 4.8 *Let* $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$ *and* $\mathcal{B} \subset \mathcal{Y} \subset \mathcal{K}_m^m$. *Let* $\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$ *be a combinatorial multivalued map such that* $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$. *Let* $\mathcal{G} := \mathcal{F}|_{\mathcal{A}}$. *Then Algorithm 4.7 modifies* $(\mathcal{A}, \mathcal{B})$ *in a finite number of steps into* $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}})$ *such that the inclusions* $i\colon (X, A) \hookrightarrow (X, \widetilde{A})$ *and* $j\colon (Y, B) \hookrightarrow (Y, \widetilde{B})$ *induce isomorphisms in homology. Moreover,* $\mathcal{F}(\widetilde{\mathcal{A}}) \subset \widetilde{\mathcal{B}}$, *and if* $\mathcal{F}|_{\mathcal{A}}$ *is acyclic, then so is* $\mathcal{F}|_{\widetilde{\mathcal{A}}}$.

Algorithms 4.1, 4.3, 4.5 and 4.7 provide a variety of methods for reducing the number of highest dimensional cubes that need to be considered in the computation of homology. Thus, before turning to algorithms which reduce the dimension we include some technical remarks concerning possible modifications and their effect on runtime.

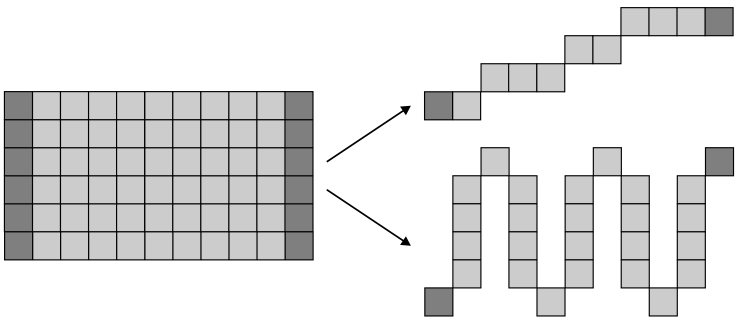

Figure 2: Two different results of reduction

In Algorithm 4.1 it is worth to make an additional effort to choose for reduction those elements of $\mathcal{K}_n^n$ which have the smallest number of neighbors in $\mathcal{X}$. Figure 2 shows two possible results of reduction of a pair of cubical sets in $\mathbf{R}^2$. The upper result was obtained with the use of this improvement, the lower one is an example of what one can obtain without it. Note that the gain is not only in the smaller number of cubes to process, but also the chain complexes and the generators of homology obtained in this way are smaller.

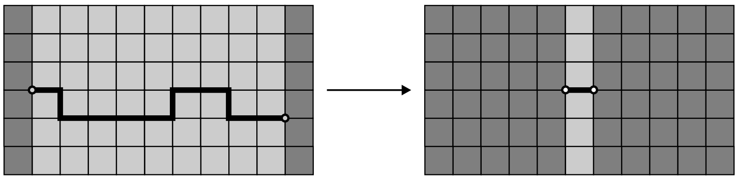

Figure 3: A homology generator obtained without and with Algorithm 4.5

Algorithm 4.5 usually reduces the computations significantly, but it causes the loss of the information about the actual generators of homology, as illustrated in Figure 3.

Note that if $(\mathcal{X}, \emptyset)$ can be reduced with Algorithm 4.1 to a set containing exactly one grid element, then $\mathcal{X}$ is acyclic. However, the converse is not true. There exist acyclic sets $\mathcal{X} \subset \mathcal{K}_n^n$ such that $\operatorname{card} \mathcal{X} > 1$, but no element of $\mathcal{X}$ can be removed without causing the change in the homology of $|\mathcal{X}|$ (consult [27] for examples).

Algorithms 4.3 and 4.7 can perform more efficiently if one cancels the verification whether the acyclicity of $\mathcal{F}$ is preserved, and verify this condition only on the final sets of cubes $\widetilde{\mathcal{X}}$ and $\widetilde{\mathcal{A}}$ (in dimension 3 our experiments suggest that the computations run about 3 times faster). However, in some cases acyclicity may be lost (an example is available at [27]). On the other hand, if F has convex values then we know *a priori* that every restriction of $\mathcal{F}$ is acyclic.

If $X \subset \mathbf{R}^k$ is a cubical set, then an elementary cube Q is called a *free face* in X if there exists exactly one elementary cube $P \subset X$ such that $Q \subset P$ and $\dim P - \dim Q = 1$.

The following algorithm removes pairs of elementary cubes from a cubical set with the use of so-called *free face collapses* (see [11]).

Algorithm 4.9 Collapse Free Faces
procedure `collapse` (**var** X: cubical set in $\mathbf{R}^n$; A, K: cubical set in $\mathbf{R}^n$);
begin
    for $k := n - 1$ **downto** 0 **do**
    while exists a k-dimensional free face Q in X
        such that $Q \not\subset A \cup K$ **do**
        begin
            let $P \subset X$ be the $(k+1)$-dimensional elementary cube
                such that $Q \subset P$;
            $X := X \setminus (\overset{\circ}{Q} \cup \overset{\circ}{P})$
        end
end.

At this point we would like to make a remark that Algorithm 4.9 works on more general data than Algorithm 4.1 and in our computations it is supposed to be the continuation of the latter, as shown in Figure 5. However, one should expect to obtain a similar result of reduction even if one does not run Algorithm 4.1 prior to Algorithm 4.9, but such computations use more resources, as one can see in Table 3 (Example 1 and 5).

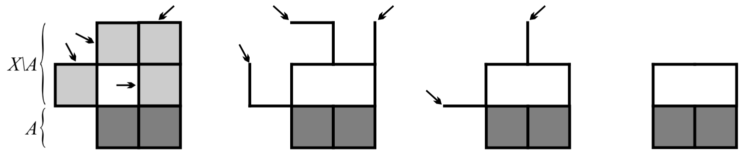

Figure 4: Reduction with Algorithm 4.9. Free faces are indicated with arrows

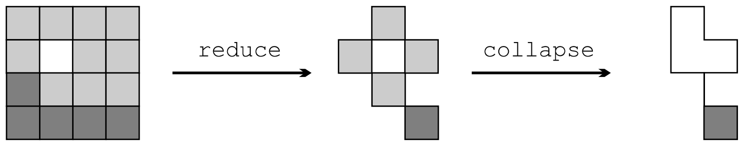

Figure 5: Two stages of reduction of cubical sets—with Algorithm 4.1 and Algorithm 4.9.

Proposition 4.10 *Let $A \subset X \subset \mathbf{R}^n$ and $K \subset X$ be cubical sets. Then Algorithm 4.9 transforms X in a finite number of steps into $\widetilde{X}$ such that $K \subset \widetilde{X} \subset X$ and the inclusion $(\widetilde{X}, A) \hookrightarrow (X, A)$ induces an isomorphism in homology.*

Like in the case of Algorithm 4.1, if $(X, \emptyset)$ can be reduced with Algorithm 4.9 to a single point, then X is acyclic, but the converse is not true (see [27] for an example).

In addition to the reduction by Algorithm 4.9, a considerable amount of data can often be removed in a very simple manner, as shown in the following result which follows directly from the excision property.

Proposition 4.11 *Let $A \subset X \subset \mathbf{R}^n$ be cubical sets. Take $\widetilde{X} := \mathrm{cl}\,(X \setminus A)$ and $\widetilde{A} := \widetilde{X} \cap A$. Then the inclusion $(\widetilde{X}, \widetilde{A}) \hookrightarrow (X, A)$ induces an isomorphism in homology.*

Since computing the intersection and the closure of difference of cubical sets is obvious from the algorithmic point of view, we do not write an explicit algorithm for computing $(\widetilde{X}, \widetilde{A})$ as defined above. In Algorithm 5.1 we refer directly to Proposition 4.11 instead.

The last algorithm introduced in this section constructs a possibly small upper semi-continuous cubical submap $\widetilde{F}$ of a given cubical multivalued map $F|_{\widetilde{X}}$ for the purpose of homology computation.

Algorithm 4.12 Reduce Map
function `reducemap` ($\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$: combinatorial multivalued map;
  $\mathcal{A}$: finite subset of $\mathcal{K}^n_n$, $\widetilde{X}$, $\widetilde{A}$: cubical set): cubical multivalued map;
begin
  $\widetilde{F} := \emptyset$;
  for $k := n$ **downto** 0 **do**
    for each elementary cube $Q \subset \widetilde{X}$ of dimension k **do**
    begin
      $D := \lceil \mathcal{F} \rceil (\overset{\circ}{Q})$;

$K := \bigcup\{\widetilde{F}(\mathring{P}) \mid P$ is an elementary cube, $Q \subset P \subset \widetilde{X}$,
and $\dim P - \dim Q = 1\}$;
if $Q \subset \widetilde{A}$ **then**
$K := K \cup \lceil \mathcal{F}|_{\mathcal{A}} \rceil(\mathring{Q})$;
`collapse` $(D, \emptyset, K)$;
$\widetilde{F}(\mathring{Q}) := D$; [see explanation in Section 3]
end;
return $\widetilde{F}$
end.

Proposition 4.13 *Let $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$ and $\mathcal{B} \subset \mathcal{Y} \subset \mathcal{K}_m^m$. Let $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$ be a combinatorial multivalued map. Assume that $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$. Let $\mathcal{G} := \mathcal{F}|_{\mathcal{A}}$. Let $\widetilde{A} \subset \widetilde{X} \subset \mathbf{R}^n$ be cubical sets such that $\widetilde{A} \subset A$ and $\widetilde{X} \subset X$. Let $i: (\widetilde{X}, \widetilde{A}) \hookrightarrow (X, A)$ denote the inclusion map. Then Algorithm 4.12 applied to $\mathcal{F}$, $\mathcal{A}$, $\widetilde{X}$, $\widetilde{A}$ returns $\widetilde{F}$ in a finite number of steps, such that $\widetilde{F}$ is an upper semi-continuous cubical multivalued map which is a submap of F, $\widetilde{G} := G|_{\widetilde{A}}$ is a submap of $F|_{\widetilde{A}}$ and if F is acyclic then so is $\widetilde{F}$.*

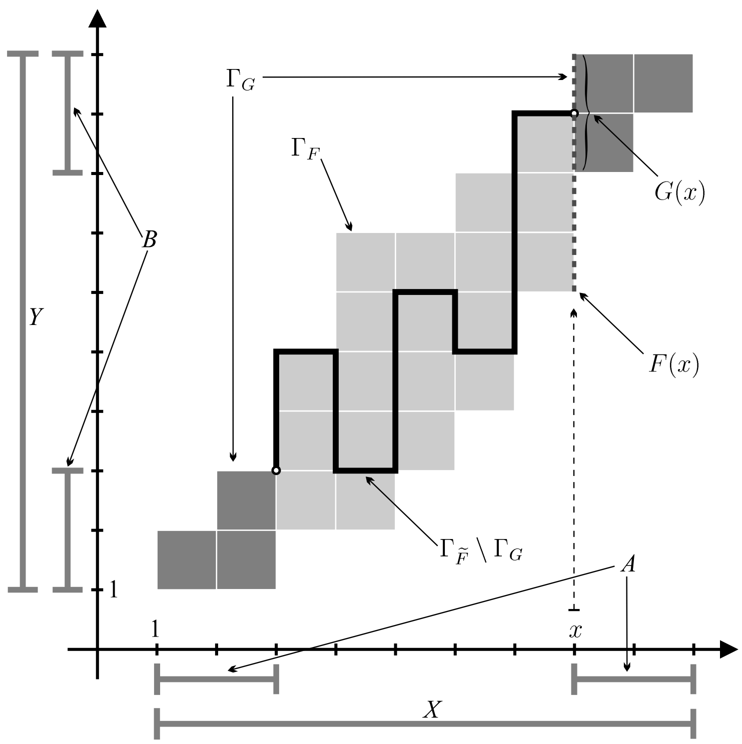

Figure 6: The graph of F and the graph of $\widetilde{F}$; note that $F(x)$ (indicated with a dashed line) is not contained in B, although $x \in A$

We would like to point out that Algorithm 4.12 is crucial for the effectiveness of our approach. This is due to the fact that if $X, Y \subset \mathbf{R}^n$, then the graph of F

is a subset of $\mathbf{R}^{2n}$. However, Algorithm 4.12 can usually replace this graph with a subset that is essentially n-dimensional, as illustrated in Figure 6. Note that if complicated acyclic cubical sets which cannot be reduced by means of free face collapses appear in Algorithm 4.12, then the dimension of the created graph is higher. This impacts the effectiveness of the algorithm since the associated algebraic computations become more complicated. Observe that the graph of G does not need to be reduced at all, because for relative homology computation all the generators of the cubical chain complex of Γ_G are neglected.

5 Homology computation of maps

In this section we gather the algorithms introduced in the previous sections in order to compute the homology of a continuous map, given its representation. We also repeat the statement of Theorem 1.1 and we prove it.

Before we proceed, we would like to explain the meaning of the parameter "incl" appearing in Algorithm 5.1: If it is set to **true**, then the algorithm assumes $(X, A) \subset (Y, B)$, and takes into consideration the inclusion $i\colon (X, A) \hookrightarrow (Y, B)$ to find the endomorphism $(i_*)^{-1} \circ f_*\colon H_*(X, A) \to H_*(X, A)$. Otherwise the homomorphism $f_*\colon H_*(X, A) \to H_*(Y, B)$ is computed and its matrix is given with respect to some generators of $H_*(X, A)$ and $H_*(Y, B)$ which are unrelated to each other even if $(X, A) \subset (Y, B)$.

Algorithm 5.1 Computation of Homology Map
function `homology` ($\mathcal{F}\colon \mathcal{X} \rightrightarrows \mathcal{Y}$: combinatorial multivalued map;
    $\mathcal{A}$: finite subset of $\mathcal{K}^n_n$, $\mathcal{B}$: finite subset of $\mathcal{K}^m_m$, **bool** incl):
    homomorphism;
begin
    `expandF` ($\mathcal{F}$, $\mathcal{A}$, $\mathcal{B}$); [Algorithm 4.7]
    `reduceF` ($\mathcal{X}$, $\mathcal{A}$, $\mathcal{F}$); [Algorithm 4.3]
    $\mathcal{S} := \mathcal{F}(\mathcal{X})$;
    if incl **then** $\mathcal{S} := \mathcal{S} \cup \mathcal{X}$;
    `reduce` ($\mathcal{Y}$, $\mathcal{B}$, $\mathcal{S}$); [Algorithm 4.1]
    `expandA` ($\mathcal{Y}$, $\mathcal{B}$); [Algorithm 4.5]
    $\widetilde{X} := |\mathcal{X}|$; $\widetilde{A} := |\mathcal{A}|$;
    `collapse` ($\widetilde{X}$, $\widetilde{A}$, $\emptyset$); [Algorithm 4.9]
    $\widetilde{X} := \operatorname{cl}(\widetilde{X} \setminus \widetilde{A})$; $\widetilde{A} := \widetilde{A} \cap \widetilde{X}$; [Proposition 4.11]
    $\widetilde{F} :=$ `reducemap` ($\mathcal{F}$, $\mathcal{A}$, $\widetilde{\mathcal{X}}$, $\widetilde{\mathcal{A}}$); [Algorithm 4.12]
    $\mathcal{G} := \mathcal{F}|_{\mathcal{A}}$;
    $\widetilde{G} := G|_{\widetilde{A}}$;
    $K := q(\Gamma_{\widetilde{F}})$;
    if incl **then** $K := K \cup \widetilde{X}$;
    $\widetilde{Y} := |\mathcal{Y}|$; $\widetilde{B} := |\mathcal{B}|$;
    `collapse` ($\widetilde{Y}$, $\widetilde{B}$, K); [Algorithm 4.9]
    if incl **then**
        $\varphi :=$ `proj_p` ($\widetilde{F}$, $\widetilde{G}$, $\widetilde{Y}$, $\widetilde{B}$); [Algorithm 3.4]
    else $\varphi := 0$;
    $\psi :=$ `proj_q` ($\widetilde{F}$, $\widetilde{G}$, $\widetilde{Y}$, $\widetilde{B}$); [Algorithm 3.5]
    $(\bar{\varphi}, \bar{\psi}) :=$ `homchain` ($\Gamma_{\widetilde{F}}$, $\Gamma_{\widetilde{G}}$, $\widetilde{Y}$, $\widetilde{B}$, $\widetilde{Y}$, $\widetilde{B}$, φ, ψ); [Algorithm 3.6]

if incl **then return** $\bar{\psi} \circ (\bar{\varphi})^{-1}$
else return $\bar{\psi}$
end.

Theorem 5.2 (Theorem 1.1) *Let $A \subset X \subset \mathbf{R}^n$ and $B \subset Y \subset \mathbf{R}^m$ be full cubical sets. Let the combinatorial multivalued map $\mathcal{F}: \mathcal{X} \rightrightarrows \mathcal{Y}$ be a representation of*

$$f: (X, A) \to (Y, B).$$

*Assume that $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$ and that both $\mathcal{F}$ and $\mathcal{F}|_{\mathcal{A}}$ are acyclic. Then the homomorphism returned by Algorithm 5.1 invoked with $\mathcal{F}$, $\mathcal{A}$, $\mathcal{B}$ and "incl" set to **false** coincides with $f_*: H_*(X, A) \to H_*(Y, B)$ in the sense that the domain D of this homomorphism is isomorphic to $H_*(X, A)$, the codomain C of it is isomorphic to $H_*(Y, B)$, and the following diagram, in which φ denotes the returned homomorphism, commutes*

$$\begin{array}{ccc} D & \xrightarrow{\varphi} & C \\ \downarrow \simeq & & \downarrow \simeq \\ H_*(X, A) & \xrightarrow{f_*} & H_*(Y, B) \end{array}$$

*Moreover, if $\mathcal{X} \subset \mathcal{Y}$, $\mathcal{A} \subset \mathcal{B}$ and the inclusion $i: (X, A) \hookrightarrow (Y, B)$ induces an isomorphism in homology, then the homomorphism returned by Algorithm 5.1 invoked with $\mathcal{F}$, $\mathcal{A}$, $\mathcal{B}$ and "incl" set to **true** coincides with the endomorphism $(i_*)^{-1} \circ f_*: H_*(X, A) \to H_*(X, A)$.*

Proof: At the beginning of Algorithm 5.1, Algorithm 4.7 transforms $\mathcal{A}$, $\mathcal{B}$ to $\mathcal{A}_1$, $\mathcal{B}_1$ such that by Proposition 4.8 the inclusions $i_1: (X, A) \hookrightarrow (X, A_1)$ and $j_1: (Y, B) \hookrightarrow (Y, B_1)$ induce isomorphisms in homology. Moreover, the map $\mathcal{G}_1 := \mathcal{F}|_{\mathcal{A}_1}$ is acyclic.

Next, Algorithm 4.3 transforms $(\mathcal{X}, \mathcal{A}_1)$ to $(\mathcal{X}_2, \mathcal{A}_2)$ such that by Proposition 4.4 the inclusion $i_2: (X_2, A_2) \hookrightarrow (X, A_1)$ induces an isomorphism in homology and the maps $\mathcal{F}_2 := \mathcal{F}|_{\mathcal{X}_2}$ and $\mathcal{G}_2 := \mathcal{F}|_{\mathcal{A}_2}$ are acyclic.

Afterwards, Algorithm 4.1 transforms $(\mathcal{Y}, \mathcal{B}_1)$ to $(\mathcal{Y}_2, \mathcal{B}_2)$ such that the inclusion $j_2: (Y_2, B_2) \hookrightarrow (Y, B_1)$ induces an isomorphism in homology. Note that $\mathcal{F}_2(\mathcal{X}_2) \subset \mathcal{Y}_2$ and $\mathcal{F}_2(\mathcal{A}_2) \subset \mathcal{B}_2$, which implies that $F_2(X_2) \subset Y_2$ and $G_2(A_2) \subset B_2$. Moreover, if "incl" is set to **true**, then also $\mathcal{X}_2 \subset \mathcal{Y}_2$ and $\mathcal{A}_2 \subset \mathcal{B}_2$, and therefore $X_2 \subset Y_2$ and $A_2 \subset B_2$.

In the next step, Algorithm 4.5 transforms $\mathcal{B}_2$ to $\mathcal{B}_3$ such that the inclusion $j_3: (Y_2, B_2) \hookrightarrow (Y_2, B_3)$ induces an isomorphism in homology.

Then Algorithm 4.9 and the two assignments that follow it transform (X_2, A_2) to $(\widetilde{X}, \widetilde{A})$ such that the inclusion $i_3: (\widetilde{X}, \widetilde{A}) \hookrightarrow (X_2, A_2)$ induces an isomorphism in homology by Propositions 4.10 and 4.11. Note that the maps $\widetilde{F}_2 := F_2|_{\widetilde{X}}$ and $\widetilde{G} := G_2|_{\widetilde{A}}$ are acyclic as restrictions of acyclic cubical multivalued maps F_2 and G_2, respectively.

Next, Algorithm 4.12 constructs the submap $\widetilde{F}: \widetilde{X} \rightrightarrows Y_2$ of $\widetilde{F}_2$ and the two assignments that follow it construct $\widetilde{G}$ as above. Proposition 4.13 implies that $\widetilde{F}$ is acyclic. Moreover, Proposition 2.5 implies that $(\widetilde{F}, \widetilde{G})_* = (\widetilde{F}_2, \widetilde{G})_*$.

In the next step, Algorithm 4.9 transforms (Y_2, B_3) to $(\widetilde{Y}, \widetilde{B})$ such that the inclusion $j_4: (\widetilde{Y}, \widetilde{B}) \hookrightarrow (Y_2, B_3)$ induces an isomorphism in homology. Since

$\widetilde{F}(\widetilde{X}) \subset \widetilde{Y}$, the multivalued maps $\widetilde{F}: \widetilde{X} \rightrightarrows \widetilde{Y}$ and $\widetilde{G}: \widetilde{A} \rightrightarrows \widetilde{B}$ are well-defined. Moreover, if "incl" is set to **true**, then $\widetilde{X} \subset \widetilde{Y}$.

Consider the following diagram which gathers most of the sets and maps discussed so far:

$$\begin{array}{ccccccccc}
(X,A) & \overset{i_1}{\hookrightarrow} & (X,A_1) & \overset{i_2}{\hookleftarrow} & (X_2,A_2) & \overset{i_3}{\hookleftarrow} & (\widetilde{X},\widetilde{A}) & & \\
\downarrow\downarrow (F,G) & & \downarrow\downarrow (F,G_1) & & \downarrow\downarrow (F_2,G_2) & & \downarrow\downarrow (\widetilde{F}_2,\widetilde{G}) & \searrow\searrow (\widetilde{F},\widetilde{G}) & \\
(Y,B) & \overset{j_1}{\hookrightarrow} & (Y,B_1) & \overset{j_2}{\hookleftarrow} & (Y_2,B_2) & \overset{j_3}{\hookrightarrow} & (Y_2,B_3) & \overset{j_4}{\hookleftarrow} & (\widetilde{Y},\widetilde{B})
\end{array}$$

This is not a commutative diagram, but it becomes one after applying the homology functor. Then the horizontal arrows correspond to isomorphisms. Therefore, $f_* = (F,G)_* \approx (\widetilde{F},\widetilde{G})_*$. In addition to this, if "incl" is set to **true**, then the inclusion map $\widetilde{i} : (\widetilde{X},\widetilde{A}) \hookrightarrow (\widetilde{Y},\widetilde{B})$ is well-defined and $i_* \approx \widetilde{i}_*$.

In the remaining computations programmed in Algorithm 5.1, either the homomorphism $\widetilde{q}_*: H_*(\Gamma_{\widetilde{F}}, \Gamma_{\widetilde{G}}) \to H_*(\widetilde{Y},\widetilde{B})$ induced in homology by the natural projection q, or the homomorphism $\widetilde{q}_* \circ (\widetilde{i}_*)^{-1}: H_*(\widetilde{Y},\widetilde{B}) \to H_*(\widetilde{Y},\widetilde{B})$ is computed, which corresponds either to f_* or $(i_*)^{-1} \circ f_*$, respectively. ■

6 Examples

In this section several examples of the applications of the algorithms introduced in this paper are discussed and the issue of computational complexity is briefly adressed. Some possible improvements of the algorithms are also indicated.

A software implementation of the algorithms introduced in this paper is available to the public at the website [27]. In particular, a computer program for the computation of the homomorphism induced in homology by a combinatorial multivalued map $\mathcal{F}: (\mathcal{X},\mathcal{A}) \rightrightarrows (\mathcal{Y},\mathcal{B})$ is available there, as well as a program which verifies whether a given map $\mathcal{F}$ satisfies the assumptions of Theorem 3.1.

To the best of our knowledge the first and only other dimension independent algorithm for computing homology of maps is due to M. Allili and T. Kaczynski [1]. Therefore, a comparison is appropriate. To begin with, the algorithm of [1] requires that the upper representation F of the combinatorial multivalued map has convex as opposed to acyclic values for each point in the domain. Moreover, the issue of relative homology is not addressed there. In addition to that, no geometric reduction is performed, which usually results in much larger algebraic data that needs to be processed. Last but not least, the algorithm in [1] produces only a chain map φ and one needs to continue the computations further in order to find the homomorphism induced by this chain map in homology. These algebraic computations are included in our algorithm. An actual comparison of effectiveness of the computer program [27] based on our algorithm with the implementation of [1] introduced in [13] proves the superiority of our approach (consult a discussion in [27] for details).

In order to illustrate the effectiveness of the algorithm introduced in this paper, we would like to mention a few example maps which we computed for benchmarking and testing purposes (see Table 1). The first combinatorial map is a representation of a Conley index map for an unstable periodic trajectory

[29], the second arises from a Conley index map for a finite-dimensional approximation of the Kot-Shaffer map [5], and the remaining three are rigorous enclosures of various index maps for an attracting periodic trajectory in the Rössler equations [25].

All the running times are measured accurately and refer to a PC with a 1 GHz processor running Linux. The memory measurements are only approximate. In Table 1 we indicate the size of the data in terms of the dimension of the space and the number of cubes in the domain of the map. The topological complexity of the examples is indicated by the homology module (over the ring of integers) of the map's domain. In all the cases the homomorphism induced in homology was computed together with the homomorphism induced by the inclusion. Note that the program easily handles relatively large sets of cubes, but the computation time and memory requirements increase significantly with the dimension.

The latter observation is clearly illustrated in Table 2, which contains a benchmark comparison of the computation of the homomorphism induced in homology by an example combinatorial multivalued map arising from the Conley index map for an attracting periodic trajectory. The domain of the map taken for the tests contains 814 two-dimensional squares and was embedded in higher-dimensional spaces in order to determine how the space dimension increases the need for the computational resources. We also remark that the algebraic stage of the homology computation usually requires far more memory than the geometric reduction; therefore, the effort put into the latter pays off in the final stage of computations.

Ex. no.	space dimension	no. of cubes in $X \setminus A$ and A	$H_*(X, A)$ over $\mathbf{Z}$	computation time	memory used
1	3	2,136 and 1,016	$(0, \mathbf{Z}, \mathbf{Z})$	0.33 min	9 MB
2	6	3,647 and 6,683	$(0, \mathbf{Z}, \mathbf{Z}^{18})$	192 min (3.2 h)	100 MB
3	3	122,178 and 0	$(\mathbf{Z}, \mathbf{Z}, \mathbf{Z})$	2.1 min	28 MB
4	3	840,303 and 0	$(\mathbf{Z}, \mathbf{Z}^4, \mathbf{Z}^{44})$	245 min (4.1 h)	204 MB
5	3	1,372,328 and 0	$(\mathbf{Z}, \mathbf{Z}^8, \mathbf{Z}^{24})$	770 min (12.8 h)	616 MB

Table 1: Some example computation benchmarks

space dimension	computation time	memory used
2	0.005 min	< 2 MB
3	0.019 min	< 2 MB
4	0.074 min	5 MB
5	0.32 min	12 MB
6	1.9 min	32 MB
7	8.3 min	80 MB
8	72 min	211 MB

Table 2: A comparison of time and memory complexity for various space dimensions

For yet another benchmark we computed an endomorphism induced in homology by a simple combinatorial multivalued map on a 3-dimensional pair of cubical sets arising from a Conley index map for a repelling periodic trajectory in the plane and embedded in R^3 as in the previous example. We compared how the speed and memory usage change if we skip some of the algorithms. In Table 3 each column corresponds to one example computation. In each row, a '+' indicates which reductions were used, and a '−' shows which were disabled. The last two rows show the computation time and approximate memory usage. Notice that the lack of some reductions is compensated to a certain extent by other reductions. As one should expect, without any geometric reduction the program is very inefficient: it needs 3.7 hours and over 500 MB RAM to perform the computations that can normally be done in 22 seconds within less than 10 MB RAM.

Example no.	1	2	3	4	5	6	7
reduce (Alg. 4.1)	+	+	+	+	+	−	−
reduceF (Alg. 4.3)	+	+	+	+	+	−	−
expandA (Alg. 4.5)	+	+	−	+	+	−	−
expandF (Alg. 4.7)	+	+	−	−	+	−	−
reducemap (Alg. 4.12)	+	+	+	+	−	+	−
collapse (Alg. 4.9)	+	−	+	+	+	+	−
computation time (min)	0.36	0.64	1.8	0.94	0.95	2.1	224(!)
memory used (MB)	9.18	20.6	35.8	27.3	99.1	36.7	540(!)

Table 3: Computation times and memory usage with some geometric reductions turned off

All the combinatorial multivalued maps used for benchmarks mentioned in this section were obtained with the software available at [2] as cubical enclosures of translation maps in various ODEs, except for the 6-dimensional example listed in Table 1, which was provided to us by S. Day and O. Junge (see [27] for details).

Although we don't prove it in this paper, the worst-case complexity of all the algorithms for the geometric reduction introduced in the paper is linear in the number of [elementary] cubes, provided the space dimension is fixed. Unfortunately, this might not be the case with the algebraic homology computations used in the software (see [24]). However, due to the simplicity of that algorithm, as well as the specific data that arises from the geometric complexes, the algorithm [24] proves to be efficient in practice.

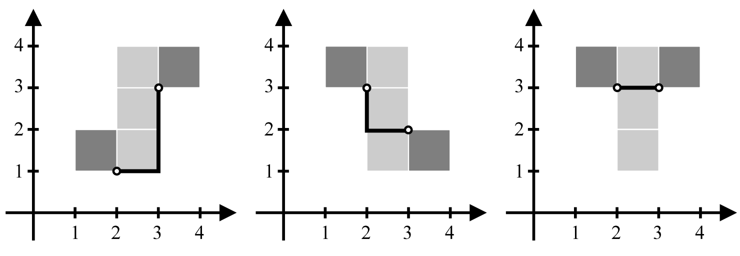

Figure 7: Three maps $\mathcal{F}$ which differ only on $\mathcal{A}$ but induce different homomorphisms in homology

Notice that in order to compute the homomorphism induced by a suitable combinatorial multivalued map $\mathcal{F}\colon (\mathcal{X},\mathcal{A}) \rightrightarrows (\mathcal{Y},\mathcal{B})$ one needs to know the map

$\mathcal{F}$ on $\mathcal{X} \setminus \mathcal{A}$ and on only these cubes in $\mathcal{A}$ which have at least one neighbor in $\mathcal{X} \setminus \mathcal{A}$. This is a valuable observation, but one can go even one step further. The idea of relative homology of (X, A) is that the subset A of X is, from the topological point of view, collapsed to a single point which is mapped to what B is collapsed to. Therefore, the homomorphism induced in relative homology by the map on (X, A) does not require the knowledge of the map on A at all. However, this observation does not carry over to the cubical setting. The three maps illustrated in Figure 7 prove that the knowledge of $\mathcal{F}$ only on $\mathcal{X} \setminus \mathcal{A}$ may not be sufficient to determine the homomorphism induced in homology by $\mathcal{F}$. In these examples, $X = Y = [1, 4]$, $A = B = [1, 2] \cup [3, 4]$, Γ_G is indicated in dark grey, $\Gamma_F \setminus \Gamma_G$ is indicated in bright grey, and $\Gamma_{\widetilde{F}} \setminus \Gamma_G$ for some $\widetilde{F}$ is sketched in black. The homomorphism induced in homology is either the identity, or minus identity, or zero.

7 Proofs for Section 4

In this section we prove all the results introduced in Section 4. Since in most cases the fact that a specific algorithm stops after a finite number of steps is fairly obvious, we skip this issue and focus on more important features. We begin with the following lemma which was proved implicitly in [25] but for the sake of completeness we provide a proof.

Lemma 7.1 *Let $Q \in \mathcal{D} \subset \mathcal{K}^n_n$. If $Q \cap |\mathcal{D} \setminus \{Q\}|$ is acyclic, then the inclusion $|\mathcal{D} \setminus \{Q\}| \hookrightarrow |\mathcal{D}|$ induces an isomorphism in homology.*

Proof: To simplify the notation, set $D' := |\mathcal{D} \setminus \{Q\}|$ and $D := |\mathcal{D}| = Q \cup D'$. Consider the following portion of the Mayer-Vietoris sequence for Q and D':

$$H_k(Q \cap D') \xrightarrow{i_k} H_k(Q) \oplus H_k(D') \xrightarrow{j_k} H_k(D) \xrightarrow{\partial_k} H_{k-1}(Q \cap D').$$

Since Q and $Q \cap D'$ are acyclic, for $k > 1$ the first and the last entries in this sequence are trivial. By the exactness of the sequence, the homomorphism in the middle, which is the homomorphism induced by the inclusion of interest (because $H_k(Q) \cong 0$), is an isomorphism for each $k > 1$.

Now consider the following part of the Mayer-Vietoris sequence:

$$\underbrace{H_1(Q \cap D')}_{0} \xrightarrow{i_1} \underbrace{H_1(Q)}_{0} \oplus H_1(D') \xrightarrow{j_1} H_1(D) \xrightarrow{\partial_1}$$

$$\xrightarrow{\partial_1} \underbrace{H_0(Q \cap D')}_{\cong \mathbf{Z}} \xrightarrow{i_0} \underbrace{H_0(Q)}_{\cong \mathbf{Z}} \oplus H_0(D') \xrightarrow{j_0} H_0(D) \xrightarrow{\partial_0} 0$$

Since i_0 acts as $z \mapsto (z, -z)$, one can see from the form of the domain and codomain of i_0 that i_0 is a monomorphism. Therfore, $\partial_1 \equiv 0$ and j_1 is an epimorphism. Since i_1 is the zero map, j_1 is in fact an isomorphism, and this is the isomorphism induced by the inclusion we are interested in, because $H_1(Q) \cong 0$.

The fact that $\partial_0 \equiv 0$ implies that j_0 is an epimorphism. Since j_0 acts as $(x, y) \mapsto x + y$, one can use the information on i_0 to see that j_0 restricted to $H_0(D')$ is an isomorphism. ■

Lemma 7.2 *Let $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$. If $Q \in \mathcal{X}$ satisfies at least one of the following conditions:*

(i) $Q \notin \mathcal{A}$ and $Q \cap |\mathcal{X} \setminus \{Q\}|$ is acyclic,

(ii) $Q \in \mathcal{A}$ and both $Q \cap |\mathcal{A} \setminus \{Q\}|$ and $Q \cap |\mathcal{X} \setminus \{Q\}|$ are acyclic,

(iii) $Q \in \mathcal{A}$ and $Q \cap |\mathcal{X} \setminus \mathcal{A}| = \emptyset$,

then the inclusion $(|\mathcal{X} \setminus \{Q\}|, |\mathcal{A} \setminus \{Q\}|) \hookrightarrow (|\mathcal{X}|, |\mathcal{A}|)$ induces an isomorphism in homology.

Proof: To simplify the notation, define $X' := |\mathcal{X} \setminus \{Q\}|$ and $A' := |\mathcal{A} \setminus \{Q\}|$. For (i) and (ii) consider the following commutative diagram

$$
\begin{array}{ccccccccc}
H_k(A') & \longrightarrow & H_k(X') & \longrightarrow & H_k(X', A') & \longrightarrow & H_{k-1}(A') & \longrightarrow & H_{k-1}(X') \\
\downarrow \cong & & \downarrow \cong & & \downarrow & & \downarrow \cong & & \downarrow \cong \\
H_k(A) & \longrightarrow & H_k(X) & \longrightarrow & H_k(X, A) & \longrightarrow & H_{k-1}(A) & \longrightarrow & H_{k-1}(X)
\end{array}
$$

where the rows are fragments of the exact sequences for the pairs (X', A') and (X, A), respectively, and the maps indicated by the vertical arrows are the homomorphisms induced by the corresponding inclusion maps. Note that in both cases (i) and (ii) the inclusions $A' \hookrightarrow A$ and $X' \hookrightarrow X$ induce isomorphisms in homology by Lemma 7.1 (however, in the case (i) the inclusion $A' \hookrightarrow A$ is just the identity map). The five lemma implies that also the inclusion $(X', A') \hookrightarrow (X, A)$ induces an isomorphism in homology.

For the case (iii) notice that since $X \setminus X' = A \setminus A'$, the inclusion map $(X', A') \hookrightarrow (X, A)$ is an excision map and therefore it induces an isomorphism in homology. ■

Proof of Proposition 4.2: The isomorphism part follows directly from Lemma 7.2. The inclusion $\mathcal{S} \subset \widetilde{\mathcal{X}}$ follows from the fact that in Algorithm 4.1 only cubes from $\mathcal{X} \setminus \mathcal{S}$ are analyzed and therefore no cube which belongs to $\mathcal{S}$ is removed from $\mathcal{X}$. ■

Proof of Proposition 4.4: We only need to prove that if $\mathcal{F}$ and $\mathcal{F}|_{\mathcal{A}}$ are acyclic then so are $\mathcal{F}|_{\widetilde{\mathcal{X}}}$ and $\mathcal{F}|_{\widetilde{\mathcal{A}}}$, because the rest follows directly from Proposition 4.2. Note that in each step of the algorithm, $\lceil \mathcal{F}|_{\mathcal{X}} \rceil|_{|\mathcal{X} \setminus \{Q\}|}$ differs from $\lceil \mathcal{F}|_{\mathcal{X} \setminus \{Q\}} \rceil$ only on the proper faces of Q, and the acyclicity of these images is verified in the condition for the removal of Q. ■

Lemma 7.3 *Let $\mathcal{A} \subset \mathcal{X} \subset \mathcal{K}_n^n$. Let $Q \in \mathcal{X}$. If $Q \cap |\mathcal{A}|$ is acyclic, then the inclusion $(|\mathcal{X}|, |\mathcal{A}|) \hookrightarrow (|\mathcal{X}|, |\mathcal{A} \cup \{Q\}|)$ induces an isomorphism in homology.*

Proof: If $Q \in \mathcal{A}$, then this is trivial. Otherwise, we use Lemma 7.1 and the five lemma in the following way.

To simplify the notation, let $\bar{A} := |\mathcal{A} \cup \{Q\}|$. Consider the following commutative diagram:

$$
\begin{array}{ccccccccc}
H_k(A) & \longrightarrow & H_k(X) & \longrightarrow & H_k(X, A) & \longrightarrow & H_{k-1}(A) & \longrightarrow & H_{k-1}(X) \\
\downarrow \cong & & \downarrow \cong & & \downarrow & & \downarrow \cong & & \downarrow \cong \\
H_k(\bar{A}) & \longrightarrow & H_k(X) & \longrightarrow & H_k(X, \bar{A}) & \longrightarrow & H_{k-1}(\bar{A}) & \longrightarrow & H_{k-1}(X)
\end{array}
$$

where the rows are fragments of the exact sequences for the pairs (X, A) and $(X, \bar{A})$, respectively, and the maps indicated by the vertical arrows are the homomorphisms induced by the corresponding inclusion maps. By Lemma 7.1, the inclusion $A \hookrightarrow \bar{A}$ induces an isomorphism in homology (we apply this lemma to the inclusion $|(\mathcal{A} \cup \{Q\}) \setminus \{Q\}| \hookrightarrow \mathcal{A} \cup \{Q\}$). The inclusion $X \hookrightarrow X$ induces the identity isomorphism. By the five lemma, also the inclusion $(X, A) \hookrightarrow (X, \bar{A})$ induces an isomorphism in homology. ■

Proof of Proposition 4.6: This follows directly from Lemma 7.3. ■

Proof of Proposition 4.8: The fact that the inclusion i induces an isomorphism in homology follows directly from Lemma 7.2, case (i). For the inclusion j, note that the condition "*reduce* $(\mathcal{B} \cup \mathcal{F}(Q), \emptyset, \mathcal{B}) = \mathcal{B}$" implies that the inclusion $|\mathcal{B}| \hookrightarrow |\mathcal{B} \cup \mathcal{F}(Q)|$ induces an isomorphism in homology, and so does the inclusion $(Y, |\mathcal{B}|) \hookrightarrow (Y, |\mathcal{B} \cup \mathcal{F}(Q)|)$ (see the proof of Lemma 7.3 for details).

The inclusion $\mathcal{F}(\widetilde{\mathcal{A}}) \subset \widetilde{\mathcal{B}}$ follows from the fact that whenever Q is added to $\mathcal{A}$, its image is added to $\mathcal{B}$.

The acyclicity of $\mathcal{F}$ on $\widetilde{\mathcal{A}}$ follows from the same argument as used in the proof of Proposition 4.4. ■

Lemma 7.4 *Let $A \subset X \subset \mathbf{R}^n$ be cubical sets. Let $Q \subset X$, $Q \not\subset A$, be a free face in X. Let $P \subset X$ be the elementary cube such that $Q \subset P$ and $\dim P - \dim Q = 1$. Then the inclusion $(X \setminus (\mathring{Q} \cup \mathring{P}), A) \hookrightarrow (X, A)$ induces an isomorphism in homology.*

Proof: In [11] such a modification of (X, A) is called a free face collapse. A minor modification of the proof therein shows that the inclusion in question induces an isomorphism in homology. ■

Proof of Proposition 4.10: The isomorphism part follows directly from Lemma 7.4. The inclusion $K \subset \widetilde{X}$ follows from the fact that whenever $Q \subset K$, the neither $\mathring{Q}$ nor $\mathring{P}$ is removed from X (note that if $P \subset K$, then also $Q \subset K$). ■

Proof of Proposition 4.13: The fact that $\widetilde{F}$ is an upper semicontinuous cubical multivalued map follows directly from the way $\widetilde{F}$ is constructed. Since for every $x \in \widetilde{X}$ its image $\widetilde{F}(x)$ is constructed from $F(x)$ with Algorithm 4.9, the inclusion $\widetilde{F}(x) \subset F(x)$ is obvious. Moreover, $\widetilde{G} := G|_{\widetilde{A}}$ is a submap of $\widetilde{F}$, because whenever $\mathring{Q} \subset \widetilde{A}$, its image by G is added to K so that $\widetilde{F}(\mathring{Q})$ contains it. The acyclicity of $\widetilde{F}$ and $\widetilde{G}$ follows from Proposition 4.10, because each $\widetilde{F}(\mathring{Q})$ is obtained from an acyclic set $F(\mathring{Q})$ with Algorithm 4.9. ■

References

[1] M. Allili, T. Kaczynski, *An algorithmic approach to the construction of homomorphisms induced by maps in homology*, Trans. Amer. Math. Soc. **352** (2000), no. 5, 2261–2281.

[2] Computer Assisted Proofs in Dynamics Group, `http://capd.wsb-nlu.edu.pl`.

[3] C. Conley, *Isolated invariant sets and the Morse index*, CBMS Regional Conference Series in Math., no. 38, Amer. Math. Soc., Providence, RI, 1978.

[4] S. Day, *A Rigorous Numerical Method in Infinite Dimensions*, PhD thesis, Georgia Institute of Technology, 2003.

[5] S. Day, O. Junge, and K. Mischaikow, *A Rigorous Numerical Method for the Global Analysis of an Infinite Dimensional Discrete Dynamical System,* SIAM Journal on Applied Dynamical Systems **3** (2004) 117–160.

[6] S. Day, O. Junge, and K. Mischaikow, Towards automated chaos verification, *Proceedings Equadiff03.*

[7] Lech Górniewicz, *Homological methods in fixed-point theory for multivalued maps*, Diss. Math. CXXIX, 1976.

[8] L. Górniewicz, Topological Degree of Morphisms and its Applications to Differential Inclusions, *Raccolta di Seminari del Dipartimento di Matematica dell'Universita degli Studi della Calabria*, No. **5**, 1983.

[9] A. Granas and L. Górniewicz, Some general theorems in coincidence theory, J. Math. Pure Appl. 60 (1981), 661–373.

[10] T. Kaczyński, M. Mrozek, M. Ślusarek, *Homology computation by reduction of chain complexes*, Computers and Mathematics, **35** (1998), no. 4, 59–70.

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

[12] R.J. Lohner, *Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems*, in: *Computational Ordinary Differential Equations*, J.R. Cash, I. Gladwell Eds., Clarendon Press, Oxford 1992, 425–435.

[13] M. Mazur, J. Szybowski, *The implementation of the Allili-Kaczyński algorithm for the construction of a chain homomorphism induced by a multivalued map*, Proceedings of the International Conference on Differential Equations (Berlin, 1999), 225–227, World Sci. Publishing, River Edge, NJ, 2000.

[14] K. Mischaikow, Topological techniques for efficient rigorous computations in dynamics, *Acta Numerica 2002* Cambridge University Press 2001.

[15] K. Mischaikow and M. Mrozek , 'Chaos in Lorenz equations: A computer assisted proof', *Bull. Amer. Math. Soc. (N.S.)* **33**, (1995) 66–72.

[16] K. Mischaikow and M. Mrozek, 'Chaos in Lorenz equations: A computer assisted proof, Part II: Details', *Mathematics of Computation*, **67**, (1998)1023–1046.

[17] K. Mischaikow and M. Mrozek, Conley index, In *Handbook of dynamical systems, Vol 2.* pages 393–460. North-Holland, Amsterdam, 2002.

[18] K. Mischaikow, M. Mrozek, J. Reiss and A. Szymczak, Construction of symbolic dynamics from experimental time series, *Phys. Rev. Let.* **82**(6)(1999), 1144–1147.

[19] K. Mischaikow, M. Mrozek and A. Szymczak, 'Chaos in Lorenz equations: A computer assisted proof, Part III: The classic case'. J. Diff. Eqns., **169**(1) (2001), Part 3, 17–56.

[20] R.E. MOORE, *Interval analysis*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966.

[21] M. Mrozek, 'Topological invariants, multivalued maps and computer assisted proofs in dynamics', *Computers & Mathematics*, **32**, (1996) 83–104.

[22] M. MROZEK, P. PILARCZYK, *The Conley index and rigorous numerics for attracting periodic orbits*, Proceedings of the Conference on Variational and Topological Methods in the Study of Nonlinear Phenomena (Pisa, 2000), 65–74, Progr. Nonlinear Differential Equations Appl., **49**, Birkhauser Boston, Boston, MA, 2002.

[23] M. MROZEK, P. ZGLICZYŃSKI, *Set arithmetic and the enclosing problem in dynamics*, Ann. Polon. Math. **74** (2000), 237–259.

[24] P. PILARCZYK, *Algorithmic homology computation over a Euclidean domain*, in preparation.

[25] P. PILARCZYK, *Computer assisted method for proving existence of periodic orbits*, Topol. Methods Nonlinear Anal. **13** (1999), no. 2, 365–377.

[26] P. PILARCZYK, *Computer assisted proof of the existence of a periodic orbit in the Rössler equations*, Proceedings of the International Conference on Differential Equations (Berlin, 1999), 228–230, World Sci. Publishing, River Edge, NJ, 2000.

[27] P. PILARCZYK, *Homology Software* [in:] Computational Homology Program, `http://www.math.gatech.edu/~chom`.

[28] P. PILARCZYK, *The Conley index and rigorous numerics for hyperbolic periodic trajectories in ODEs*, in preparation.

[29] P. PILARCZYK, *Topological-numerical approach to the existence of periodic trajectories in ODEs*, Discrete and Continuous Dynamical Systems 2003, A Supplement Volume: Dynamical Systems and Differential Equations, 701–708.

[30] E.H. SPANIER, *Algebraic topology*, McGraw-Hill 1966.

[31] A. SZYMCZAK, *A combinatorial procedure for finding isolating neighbourhoods and index pairs*, Proc. Royal Soc. Edinburgh Sect. A **127** no. 5 (1997), 1075–1088.

[32] A. SZYMCZAK, *The Conley index for discrete semidynamical systems*, Topology Appl. **66** (1995), 215–240.

[33] P. ZGLICZYŃSKI, C^1*-Lohner algorithm*, Foundations of Computational Mathematics, 2(2002), 429–465.