ssdnm
środowiskowe
studia doktoranckie
z nauk matematycznych

Michał Kukieła

Uniwersytet M. Kopernika w Toruniu

Rescalings and the coreduction homology algorithm for cubical complexes

Praca semestralna nr 3

(semestr letni 2011/12)

Opiekun pracy: prof. dr hab. Marian Mrozek

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Rescalings and the coreduction homology algorithm for cubical complexes

Michał Kukieła

Abstract

A detailed proof is given of a result sketched in a paper of M. Mrozek and B. Batko concerning the behaviour of the coreduction homology algorithm on rescalings of 2-dimensional cubical complexes. We also discuss a conjecture concerning the size of the output of the algorithm.

1 Introduction

The coreduction algorithm discussed in the present paper is an algorithm for preprocessing homology computations of large geometric complexes (we shall restrict our attention to cubical complexes, though the algorithm is applicable in a much wider setting). It was proposed by Marian Mrozek and Bogdan Batko in [6] and is a member of a family of algorithms [7, 4, 3], whose underlying idea is to reduce the considered complex before the actual computation of homology takes place.

In applications this is often necessary, since the classical algorithms for homology computation, based on Smith diagonalization, are relatively slow and cannot be effectively applied to complexes containing, say, millions of facets, which appear in practice (cf. the discussion in [6]). In contrast, the reduction methods very often allow to significantly decrease the size of the input with little computational cost (for example, the coreduction algorithm is linear in size of the input). Implementations of the algorithms could be found at [1, 2].

One source of applications of computational homology are some algorithms in rigorous numerics of dynamical systems [5]. In such applications we encounter large cubical complexes with relatively simple topology, whose homology needs to be verified. In particular, big parts of these complexes resemble cubulations of the euclidean space. The reason for this is that algorithms producing these complexes apply numerous subdivisions. One would like to have some ideas on efficiency of the reduction algorithms applied to such complexes – do they 'cut off' the large uninteresting parts of the complexes?

In [6] the authors stated a result that gives strong evidence that the answer to the above question is positive. They sketched the proof in the case of 2-dimensional cubical sets and announced a proof of the result for complexes of arbitrary dimension, which however was never published. In this paper we prove their theorem for complexes of dimension 2, giving all the necessary details. It is the first step towards providing the 'technically complicated' proof of the general case.

The paper [6] also contains a conjecture concerning the size of the output of the coreduction algorithm in a wider setting. We shall give a simple example that shows the conjecture is not true in its full generality and propose a restriction of the conjecture.

2 Preliminaries

We assume the reader is familiar with basic notions of homology theory, and in particular the homology theory of cubical sets [3]. In this section we shall fix some notation and introduce the coreduction homology algorithm together with some of its crucial properties.

An *elementary interval* in $\mathbb{R}$ is an interval $[x, x+\delta]$, where $x \in \mathbb{Z}$ and $\delta \in \{0,1\}$. If $\delta = 0$, the interval is called *degenerate*. An *elementary cube of dimension* d in $\mathbb{R}^n$ is the product of n elementary intervals, d of which are nondegenerate. Elementary cubes of dimensions $0, 1, 2$ shall be called, respectively, *vertices, edges* and *squares*.

The union of a finite, nonempty family of elementary cubes in $\mathbb{R}^n$ is called a *cubical set*. Let X be a cubical set in $\mathbb{R}^n$ and let $\mathcal{K}^d(X)$ be the set of elementary cubes $Q \subseteq \mathbb{R}^n$ of dimension d such that $Q \subseteq X$. Let $\mathcal{K}^{-1}(X) = \{\emptyset\}$. *Dimension* of X is the number $\dim(X) = \max\{d \in \mathbb{N} : \mathcal{K}^d(X) \neq \emptyset\}$. The cubical set X is *full* if $X = \bigcup \mathcal{K}^{\dim(X)}$.

For an elementary d-dimensional cube $Q = I_1 \times \cdots \times I_n \subseteq \mathbb{R}^n$, where I_j are elementary intervals, let

$$\nu(Q,j) = \begin{cases} 0 & \text{if } I_j \text{ is degenerate,} \\ \#\{i < j : I_i \text{ is nondegenerate}\} & \text{otherwise.} \end{cases}$$

If $I = [x, x+1]$ is a nondegenerate elementary interval, define the degenerate elementary intervals $I^+ = [x+1, x+1]$ and $I^- = [x, x]$. Now, for an elementary cube P set

$$\kappa_\Box(Q,P) = \begin{cases} (-1)^{\nu(Q,j)} & \text{if } \dim(P) = \dim(Q) - 1 \text{ and} \\ & \quad P = I_1 \times \cdots \times I_j^- \times \cdots \times I_n, \\ (-1)^{\nu(Q,j)+1} & \text{if } \dim(P) = \dim(Q) - 1 \text{ and} \\ & \quad P = I_1 \times \cdots \times I_j^+ \times \cdots \times I_n, \\ 0 & \text{otherwise.} \end{cases}$$

Let

$$\kappa_\Box(Q,\emptyset) = \begin{cases} 1 & \text{if } \dim(Q) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

A *S-complex* consists of a set S of *generators* with gradation $S = \coprod_{i=0}^{\infty} S_i$ and a graded map $\kappa : S \times S \to \mathbb{Z}$, called the *coincidence index*, satisfying

$$\kappa(s,t) \neq 0 \implies \dim(s) = \dim(t) + 1$$

, such that $(\mathbb{Z}(S), d)$ is a chain complex, where $d(s) = \sum_{t \in S} \kappa(s,t)t$. Here, $\dim(s)$ denotes the unique number such that $s \in S_{\dim(p)}$. Homology of the chain complex $(\mathbb{Z}(S), d)$ is called the *homology of S* and denoted by $H_*(S)$.

If X is a cubical set, then the above defined pair $(\mathcal{K}(X), \kappa_\square)$, where $\mathcal{K}(X) = \bigcup_{p=-1}^{\infty} \mathcal{K}^p(X)$ is an S-complex. The homology of this chain complex coincides with the usual reduced singular homology of X.

In an S-complex S define for $s \in S$ the sets:

$$\mathrm{bd}_S(s) = \{t \in S : \kappa(s,t) \neq 0\},$$
$$\mathrm{cbd}_S(s) = \{t \in S : \kappa(t,s) \neq 0\}.$$

An element of $\mathrm{bd}(s)$ is called a *face* of s; elements of $\mathrm{cbd}(s)$ are called *cofaces* of s. Suppose $s \in S$ has a unique coface t. Let $S' = S \setminus \{s,t\}$, $\kappa' = \kappa|_{S' \times S'}$. One may show [6] that (S', κ') is an S-complex, and moreover $H_*(S) = H_*(S')$. This is a basis for Algorithm 1.

Data: S-complex S, generator $s \in S$
Result: reduced S-complex

Q := empty set of generators;
enqueue(Q,s);
while $Q \neq \emptyset$ **do**
  s := dequeue(Q);
  if $\mathrm{bd}_S(s)$ *contains exactly one element* t **then**
    S := $S \setminus \{s\}$;
    for $u \in \mathrm{cbd}_S(t)$ **do**
      if $u \notin Q$ **then** enqueue(Q,u);
    end
    S := $S \setminus \{t\}$;
  end
  else if $\mathrm{bd}_S(s) = \emptyset$ **then**
    for $u \in \mathrm{cbd}_S(s)$ **do**
      if $u \notin Q$ **then** enqueue(Q,u);
    end
  end
end
return S;

Algorithm 1: Coreduction algorithm

Algorithm 1 returns a subset S' of the input S-complex S such that S' is an S-complex and $H_*(S) = H_*(S')$, cf. [6, Theorem 6.2]. Therefore, it may in particular be used to simplify a cubical set before algorithms based on Smith normal form are applied to compute its reduced homology. Moreover, the algorithm may be implemented so that for a cubical set S in $\mathbb{R}^d$ it runs in $O(2d^2n)$ time, where n is the cardinality of S ([6, Corollary 6.3]).

Note that Algorithm 1 should be applied to connected complexes. However, one may apply it to each connected component separately, and it may be easily extended so that the "**while**" loop detects connected components.

3 Theorem

In this section we shall state precisely the theorem we are going to prove and explain some technical assumptions that we make to allow the analysis in the next section to take place.

First, we shall apply Algorithm 1 to cubical sets only and moreover assume that all considered cubical sets are full. Cubical sets are the class of S-complexes that occurs most often in applications. Converting a non-full cubical set to a homotopically equivalent full cubical set is a straighforward operation, so this restriction is not very severe. Moreover, in the proof we consider connected cubical complexes only – for disconnected complexes the proof proceeds by applying the same argument to each connected component.

To simplify analysis of Algorithm 1 we introduce the following order on elementary cubes in $\mathbb{R}^k$, called the *lexicographical order*. For an elementary cube $Q = [x_1, x_1 + \delta_1] \times \cdots \times [x_n, x_n + \delta_n]$ let $b_Q = (x_1 + \frac{\delta_1}{2}, \ldots, x_n + \frac{\delta_n}{2})$ be its barycenter. We say that $Q < Q'$ in the lexicographical order if $b_Q < b_{Q'}$ in the standard lexicographical order on $\mathbb{R}^k$. (The lexicographical order on elementary cubes can also be naturally introduced by treating them as pixels in a bitmap.)

We shall assume that the queue Q in Algorithm 1 is kept sorted with respect to lexicographical order all of the time. This modification makes the proof a lot easier, but on the other hand it increases the complexity of the algorithm, from $O(n)$ to $O(n \log n)$. However, for the result presented below concerning the overall complexity of calculating homology this minor change is irrelevant.

If the algorithm is applied to a cubical set X whose lowest vertex (in the lexicographical order) is s, we should take s as the generator that is the second argument of the algorithm. We will now introduce one more technical modification in the algorithm. Namely, before the first iteration of the "**while**" loop we shall replace the line "enqueue(Q,s);" with the following lines:

for $u \in \operatorname{cbd}_S(s)$ **do** enqueue(Q,u);
S := $S \setminus \{s, \emptyset\}$;

This does not have a great impact on the behaviour of the algorithm – in fact, if all that is concerned is the order of removal of elementary cubes, it really only changes the first few steps. However, this minor change allows us to treat the whole analysis in an unified way and in the case of cubical complexes (as opposed to general S-complexes) it seems a natural thing to do.

We also need to have a notion of subdivision of a cubical set. In our model it is most conveniently introduced by rescaling the complex. For $n \in \mathbb{N}_{>0}$ consider the map $R_n : \mathbb{R}^k \to \mathbb{R}^k$ given by multiplication by n: $R_n(x) = nx$. The image $R_n(X)$ of a cubical set $X \subseteq \mathbb{R}^k$ is a cubical set called the *n-rescaling* of X. Note that under this transformation the image of an elementary cube Q of dimension d contains n^d elementary cubes of dimension d. Therefore, rescaling may be treated as a way of expressing the subdivision process.

For a cubical set X define $\gamma(X) = \frac{|X'|}{|X|}$, where X' is the S-complex returned by Algorithm 1 applied to X. Now we can state the following theorem.

Theorem 3.1 ([6, Theorem 8.3]). *Consider the class of all n-rescalings of a fixed d-dimensional full cubical set X_0. Then γ restricted to this class satisfies $\gamma(X) = O(|X|^{-\frac{1}{d}})$.*

Before proceeding to the proof (in the case $d = 2$) we note the following corollary, which follows immediately from the more general Proposition 8.1 of [6].

Corollary 3.2. *The complexity of finding homology in the class of n-rescalings of a fixed cubical set X by means of Algorithm 1 followed by another homology algorithm applied to X' is $O(|X|^{\alpha(1-\frac{1}{d})})$, if the complexity of the other algorithm is $O(|X|^{\alpha})$.*

4 Proof of Theorem 3.1

Let X be a cubical set that is an n-rescaling of X_0 for some $n \in \mathbb{N}$. By X' we denote the S-complex returned by Algorithm 1 applied to X. For an S-complex Y denote by $|Y|_2$ the cardinality of the set of 2-dimensional elements of Y. We only need to show that

$$|X'|_2 \leq 4n|X_0|_2. \tag{1}$$

Indeed, (1) implies that:

$$|X'| \leq 9|X'|_2 \leq 36n|X_0|_2 = 36|X_0|_2^{\frac{1}{2}}|X|^{\frac{1}{2}} = O(|X|^{\frac{1}{2}}),$$

so in conclusion $\gamma(X) = O(\frac{1}{|X|^{\frac{1}{2}}})$.

Given a square Q in X we shall call the lowest (in the lexicographical order) edge in Q the *bottom edge of Q* and the highest edge the *top edge of Q*. An elementary cube E in X is called *internal* if $R_n(v) \cap E = \emptyset$ for all vertices $v \in X_0$.

We will now need a number of lemmas.

Lemma 4.1. *No square Q in X is removed by the algorithm if it contains a vertex that was not removed earlier by the algorithm.*

Proof. Suppose Q is the first square that the algorithm removes in pair with some edge A, but which at the time of removal contains a vertex v that was not removed earlier.

Therefore, there is an edge $C \neq A$ in X contained in Q that was already removed by the algorithm, but which contains v. This means the edge C was not removed in a vertex-edge coreduction and must have been removed in an edge-square coreduction with some square W. But W contains v and was removed before Q, which is a contradiction. □

Lemma 4.2. *Every vertex in X is removed in a vertex-edge coreduction, apart from the lowest one in the lexicographical order, which is removed in pair with the empty set.*

Proof. The statement concerning the vertex removed in pair with the empty set is trivial. Next, if a vertex v in X is removed, each of the edges in $\mathrm{cbd}_X(v)$ either gets enqueued, or has already been removed. In the second case, it has necessarily been removed in a vertex-edge coreduction, which follows immediately from Lemma 4.1. In the first case, either the edge gets removed at some point in a vertex-edge coreduction, or the unique vertex other than v that it contains is removed in some other vertex-edge coreduction. In any case, all vertices that are contained in any edge containg v get removed. Since we assumed that X is connected, this ends the proof. □

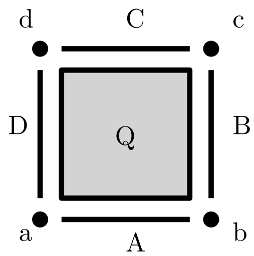

Figure 1: A square (the lexicographical order here is: $a, A, b, D, Q, B, d, C, c$).

Lemma 4.3. *In the situation of Figure 1 the edge C cannot be removed in pair with any of the vertices c or d.*

Proof. If C is removed with c, then d was removed earlier. Thus, since the queue in the algorithm is sorted with respect to the lexicographical order, either D was alredy enqueued and removed with a or a had been removed earlier. By the same reasoning b was removed before C (either in pair with A or with some other edge), and thus B was enqueued and removed with c (B was not removed in an edge-square coreduction by Lemma 4.1). This is a contradiction.

Similarily, if C is removed with d, then c was removed earlier, thus b and also a were removed, so D was removed with d, which is a contradiction. □

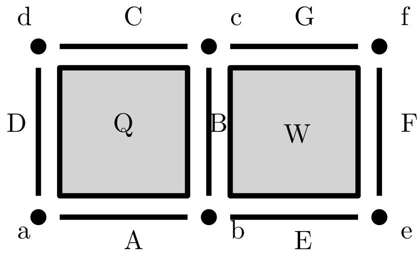

Figure 2: Two squares

Lemma 4.4. *In the situation of Figure 2, if B is an internal edge, then B cannot be removed in pair with b.*

Proof. Suppose B is the first internal edge removed by the algorithm as in the statement of the lemma. Therefore, the internal vertex c must have been removed earlier. By Lemma 4.3 it was not removed with C or G. By assumption it was not removed with any internal edge greater than c in the lexicographical order.

Thus, it was removed with an internal edge lower than c in the lexicographical order, say H (cf. Figure 3). But this means h was removed earlier, and therefore i and d, and a were removed. Therefore, b was removed before B, which is a contradiction.

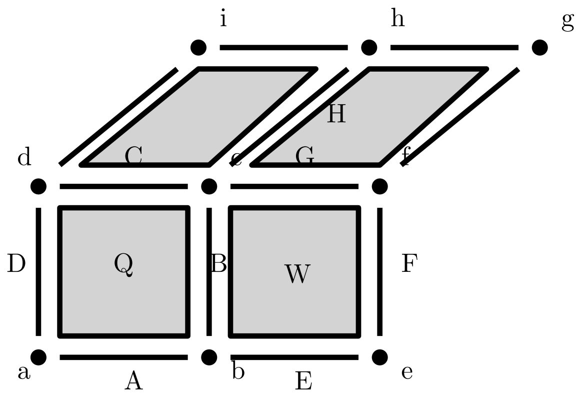

Figure 3: Four squares

□

Lemma 4.5. *If A is an internal bottom edge in X and A is not contained in any square in X smaller than A in the lexicographical order, then A is removed in a vertex-edge coreduction.*

Proof. We may adapt the notation of Figure 1. By Lemma 4.2 one of the vertices a, b is at some point removed by the algorithm and A gets enqueued. If b is removed first, then observe that since b is internal and A is contained in no square lower than Q, b is contained in no edge lower than A and A becomes the first element of the queue. Consequenty, A is removed with a. If a is removed first, then A is lower in the lexicographical order than any edge containing b and thus it is the edge A that is at some point later on removed with the vertex b. □

Lemma 4.6. *Let C be an internal top edge in X and let W be the minimal (in the lexicographical order) square in X_0 such that $C \subseteq R_n(W)$. Thus C is the top of a unique square Q in X such that $Q \subseteq R_n(W)$. If in the notation of Figure 1 the edge A and the vertices a, b are removed by the algorithm, then the edge C is removed by the algorithm in pair with the square Q.*

Proof. The vertices d, c are contained in no edges lower in the lexicographical order than D, B. If at the time of removal of a and b the vertices d and c had not been already removed, then they are removed with D and B, and consequently C is removed with Q.

Suppose that d was removed before a. Then D got enqueued and therefore, by minimality of W, D was removed with a in the next step. But this is impossible by Lemma 4.4, since D is internal. The same reasoning applies to the situation of removal of c before b is removed. □

The proof of the theorem now proceeds by induction. Introduce the following partial order on internal squares in X. Set $P \prec Q$ if the bottom edge of Q is the top edge of P. Bottoms of minimal elements of $\prec$ are removed by the algorithm by Lemma 4.5. Therefore, by Lemma 4.6 minimal elements of $\prec$ are removed together with their top edges. Now, let Q be an internal square in X and suppose that the top edges of all internal squares $P \prec Q$ were removed by the algorithm. By Lemma 4.6 the top of the square Q is removed with some square Q'. Moreover, if Q is contained in the interior of $R_n(W)$ for some square W in X_0, then $Q = Q'$.

The above reasoning shows that all squares in X that are contained in $\mathrm{Int}(R_n(W))$ for some square W in X_0 are removed by the algorithm. This proves inequality (1) and thus finishes the proof.

5 Conjecture

In this section we shall present a simple example that shows that the following conjecture stated in [6, Conjecture 8.2] is false.

Conjecture 1. *Consider a class of d-dimensional cubical sets such that any two of them are homeomorphic under a certain class of homeomorphisms. Then γ restricted to this set satisfies $\gamma(X) = O(|X|^{-\frac{1}{d}})$.*

Let X_0 be the cubical complex depicted in Figure 4. For $n \in \mathbb{N}_{>0}$ consider the homeomorphism $L_n : \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$L_n(x, y) = \begin{cases} (x, ny) & \text{if } y \geq 0, \\ (x, y) & \text{otherwise.} \end{cases}$$

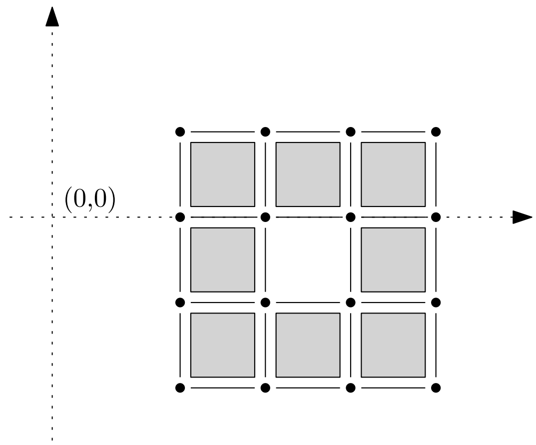

Figure 4: The cubical set X_0.

The cubical set $X_n = L_n(X_0)$ may be seen on the left of Figure 5. The S-complex X'_n obtained by applying Algorithm 1 (with the modifications described in Section 3) to X is presented on the right of Figure 5.

Note that $|X'_n| = 2n + 1$, while $|X_n| = 14n + 34$, so it follows that

$$\gamma(X_n) = \frac{2n+1}{14n+34} \neq O(\frac{1}{\sqrt{|X_n|}}).$$

Therefore, the class $\{X_n\}_{n\in\mathbb{N}}$ provides a counterexample to Conjecture 1. One can also construct larger counterexamples based on the same idea, e.g. by gluing several copies of X_n together. Such examples show, in particular, that the behaviour of Algorithm 1 in parts of the processed cubical sets resembling the euclidean space depends not only on local properties of the cubical set, but also on its global features.

One should be aware that our example applies to the version of Algorithm 1 with sorted queue. The author at the moment does not know a similar example for the algorithm without sorting, although he suspects that it exists. It is also worth noting that the algorithm with sorting is sensitive to changes in the ordering – the algorith applied to the set X_n rotated by ninety degrees gives a vastly different (and smaller) output.

A possible remedy for similar counterexamples would be to restrict the allowed classes of homeomorphisms. Perhaps the conjecture could be true for homeomorphisms that either keep the size bounded, or change the size of the complex ‘with the same rate in all directions’, though this idea needs to be made precise.

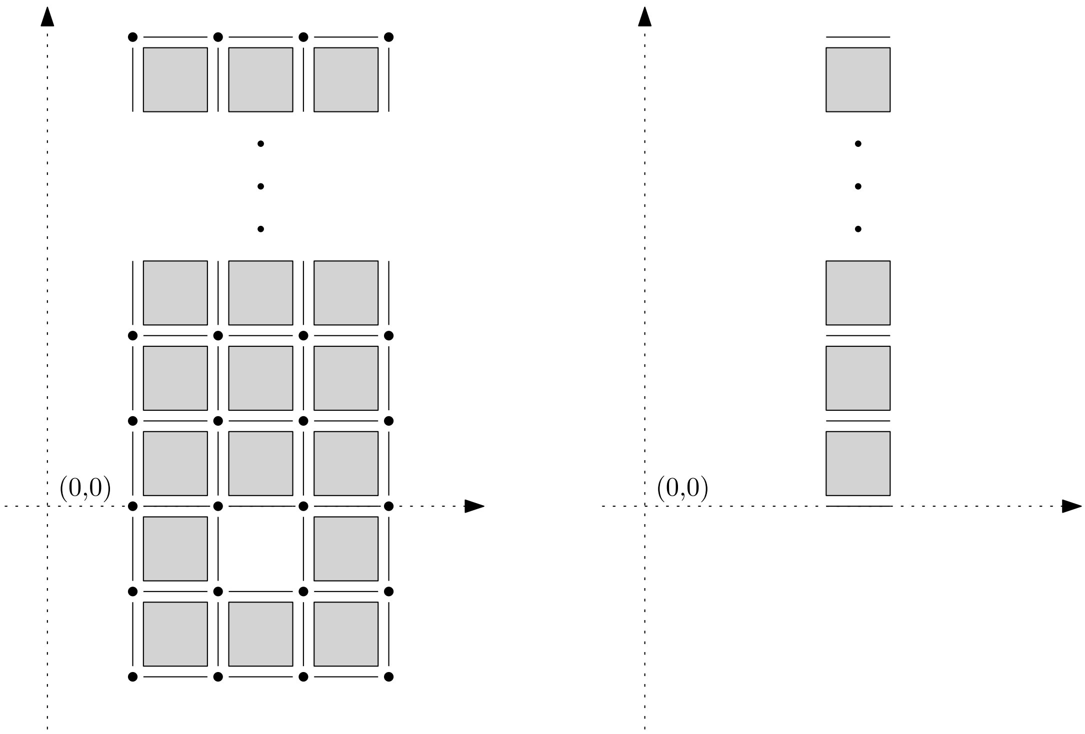

Figure 5: The cubical set X_n and the S-complex X'_n.

6 Final remarks

The author believes the general case of Theorem 3.1 can be proved by methods similar to those used in Section 4, though the proof will certainly be much more elaborated. This should be a topic of a future publication.

The structure of the output of Algorithm 1 is not well understood, and its understanding needs – as the authors of [6] have written – more experiments. Because of the high applicability and efficiency of the coreduction algorithm, this is certainly an interesting topic for research that could possibly result in improvements in the algorithm.

References

[1] http://chomp.rutgers.edu/

[2] http://redhom.ii.uj.edu.pl/

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

[4] T. Kaczynski M. Mrozek and M. Slusarek, Homology Computation by Reduction of Chain Complexes. Computers and Mathematics with Applications, 35, 59–70, 1998.

[5] M. Mrozek, Index Pairs Algorithms. Foundations of Computational Mathematics 6, 457–493, 2006.

[6] M. Mrozek, B. Batko: Coreduction homology algorithm. Discrete and Computational Geometry 47, 96–188, 2009.

[7] M. Mrozek, P. Pilarczk, N. Żelazna: Homology algorithm based on acyclic subspace. Computers and Mathematics with Applications 55, 2395–2412, 2008.