Computing the Cohomology Ring on Simplicial Complexes*

Rocío González–Díaz, Pedro Real

Dept. of Applied Math., University of Seville (Spain),
e–mails: {rogodi, real}@us.es.

Abstract

There exist several algorithms for computing the homology groups of finite simplicial complexes (for instance, [Mun84], [DE95, ELZ00] and [DG98]), but concerning the algorithmic treatment of the cohomology ring or cohomology operations, very little is known. In this paper, we establish a version of the incremental algorithm given in [ELZ00] for computing the homology, which saves additional algebraic information, allowing us the computation of the cup product and primary and secondary cohomology operations on the cohomology of a finite simplicial complex. Our algorithmic approach makes possible to consider the cup product as an efficient computational tool for distinguishing non–homotopy equivalent objects. The computational complexity of this process is also studied.

1 Introduction

A simplicial complex is a well–known discrete model of a geometric object, which consists of a collection of simplices that fit together in a natural way to form the object. In order to classify simplicial complexes from a topological point of view, a first algebraic invariant that can be used is the homology, which in some sense, counts the number of holes of a given simplicial complex.

Two relevant algorithms for computing the homology groups H_*K of a simplicial complex K must be cited: (1) the classical algorithm based on reducing certain matrices to their Smith normal form [Mun84]; (2) the incremental algorithm [DE95, ELZ00, EZ01], consisting of assembling the complex, simplex by simplex and at each step updates the Betti numbers of the current complex. Starting with the boundary of a negative simplex, this persistence process finds the cycle which is destroyed by this simplex through the search, computing in this way the geometric realization of a homology cycle. It runs in time at most $O(m^3)$, where m is the number of simplices of the complex. For simplicial complexes embedding in $\mathbf{R}^3$, this complexity is reduced to $O(m)$ in time and space [DE95]. The algorithm proposed in [DG98] is based on simulating a thickening of a given complex in $\mathbf{R}^3$ to a topological 3-manifold homotopy equivalent to it, and computing the homology groups of the last one using classical results. The time and space complexity is linear and this method also produces representations of generators of the homology groups.

In general, computing the homology is not enough for determining whether two simplicial complexes are homotopy equivalent or not. Finer algebraic invariants such as cohomology (an algebraic dual notion to homology), the cup product on cohomology or cohomology operations [Spa81], allow us to topologically distinguish two geometric objects having isomorphic homology groups. For example, a torus and the wedge product of a sphere and two circumferences have the same homology but the respective cohomology rings are "essentially" different [Mun84, c. 5]. Using a field as the coefficient group, the cohomology H^*K of a simplicial complex K gives us the same topological

*Partially supported by the PAICYT research project FQM–296 from Junta de Andalucía and the DGESIC research project PB98–1621–C02–02 from Education and Science Ministry (Spain). Web address: *http://www.us.es/gtocoma*

information as the homology of it. However, neither the additional ring structure on the cohomology determined by the cup product, nor cohomology operations can directly be produced from the algorithms previously mentioned for computing homology.

For addressing this question, the key idea is the construction of an explicit chain contraction (a special chain equivalence) connecting the chain complex C_*K, canonically associated to a simplicial complex K and its homology H_*K. In [GR01], a translation of the classical matrix algorithm (1) in terms of chain contractions is designed. We give here a version of the incremental method described in [ELZ00] in terms of chain contractions. The complexity of our method is also $O(m^3)$ where m is the number of simplices of K, but our algorithm saves information which allows us to compute not only the representations of (co)homology generators but also the following computations:

1. The (co)homology class of a (co)cycle in terms of (co)homology generators in $O(m^2)$.
2. The construction of a (co)boundary of a given (co)cycle in $O(m^2)$.
3. The induced homomorphism at (co)homology level of a simplicial map between two complexes.
4. The cohomology ring of K in $O(m^6)$.
5. Cohomology operations such as the Steenrod square operation $Sq^i\alpha_n$ in $O(i^{n-i+1}m)$ (see [GR99]); or the Adem secondary cohomology operation Ψ_2 on a cohomology class $\alpha_2 \in KerSq^2H^1K$ in $O(m^5)$ (see [GR02b]).

In this paper, we only deal with the problem of the computation of the cup product on cohomology; Roughly speaking, our algorithmic approach allows us to consider the cohomology ring as an efficient computational tool for distinguishing non-homotopy equivalent objects. Observe that the difficulty for doing this is not the computation of the multiplication tables of the rings, the problem is that one cannot, in general, by examining the multiplication tables, determine at once whether or not the rings are isomorphic. We partially solve this problem computing the first non–null homology group of the simplicial bar construction [McL75] of the cohomology ring of a given simplicial complex.

2 Homology and Chain Contractions

In this section, we design a version of the incremental algorithm given in [ELZ00] in terms of chain contractions. In this way, a chain contraction from the chain complex canonically associated to a simplicial complex K to its homology is constructed.

We first give a brief summary of concepts and notations. The terminology follows Munkres [Mun84]. For the sake of clarity and simplicity, we only define the concepts that are really essential in this paper.

In the sequel, $\mathbf{Z}_2$ is the ground ring. A *q–simplex* σ in $\mathbf{R}^n$ (where $q \leq n$) is the convex hull of $q+1$ affinely independent points $\{v_0, ..., v_q\}$. We denote $\sigma = \langle v_0, ..., v_q\rangle$. The *dimension of* σ is $|\sigma| = q$. For example, a 0–simplex is a vertex, a 1–simplex is an edge, a 2–simplex is a triangle, a 3–simplex is a tetrahedron, and so on. An *i–face* of $\sigma = \langle v_0, ..., v_q\rangle$ $(i < q)$ is an i–simplex whose vertices are in the set $\{v_0, ..., v_q\}$. The $(q-1)$–faces of σ are called the *facets* of σ. A simplex is *maximal* if it does not belong to any higher–dimensional simplex. A *simplicial complex* K is a collection of simplices such that:

- If τ is a face of $\sigma \in K$, then $\tau \in K$.
- If $\sigma', \sigma \in K$, then $\sigma' \cap \sigma \in K$ or $\sigma' \cap \sigma = \emptyset$.

Let us note that K can be given by the set of its maximal simplices. All the simplices here have finite dimension and all the simplicial complexes are finite collections. The *dimension of* K is

$dim K = max\{|\sigma| : \sigma \in K\}$. The set of all the q–simplices of K is denoted by $K^{(q)}$. Let K and L be two simplicial complexes. A map $f : K^{(0)} \to L^{(0)}$ such that whenever $\langle v_0, ..., v_q \rangle \in K$ then $f(v_0), ..., f(v_n)$ are vertices of a simplex of L, is called a *vertex map*. Let E be a subset of K. Define

$$\overline{E} = \{\sigma' \in K : \ \sigma' \leq \sigma \in E\},$$

then, the *star* of E, is the set $St\, E = \{\sigma \in K : \ \sigma \geq \sigma' \in E\}$ and the *link* of E is

$$Lk\, E = \overline{St\, E} - St\, \overline{E}\,,$$

where $\sigma' < \sigma$ means that σ' is a face of σ.

The *chain complex* C_*K associated to a simplicial complex K is a family $\{(C_qK, \partial_q)\}_{q\geq 0}$ defined in each dimension q as follows:

- C_qK is the free abelian group generated by the q–simplices of K. An element $a = \sigma_1 + \cdots + \sigma_m$ of C_qK ($\sigma_i \in K^{(q)}$) is called a *q–chain.*
- $\partial_q : C_qK \to C_{q-1}K$ called the *boundary operator* is given by

$$\partial_q \langle v_0, ..., v_q \rangle = \sum_{i=0}^{q} \langle v_0, ..., \hat{v}_i, ..., v_q \rangle$$

  where $\langle v_0, ..., v_q \rangle$ is a q–simplex of K and the hat means that v_i is omitted. By linearity, ∂_q can be extended to C_qK, where it is a homomorphism. Observe that $\partial_q \partial_{q+1} = 0$.

By abuse of notation, if L is simply a graded set, we define C_*L as the family $\{(C_qL, 0)\}_{q\geq 0}$ where C_qL is the free abelian group generated by the elements of L of degree (or dimension) q and $0 : C_qL \to C_{q+1}L$ is the null map (observe that, in this case, $H * L = C_*L$). A q–chain a is called a *q–cycle* if $\partial a = 0$. If $a = \partial b$ for some $b \in C_{q+1}K$ then a is called a *q–boundary.* We denote the groups of q–cycles and q–boundaries by Z_qK and B_qK respectively, and define $Z_0K = C_0K$. Since $B_qK \subseteq Z_qK$, define the qth *homology group* H_qK to be the quotient group Z_qK/B_qK. Given that elements of this group are cosets of the form $\alpha = a + B_qK$, where $a \in Z_qK$, the coset α is the *homology class* in H_qK determined by a or a is a *representative cycle* of α. Let K and L be two simplicial complexes. A *chain map* $f : C_*K \to C_*L$ is a family of homomorphisms

$$\{f_q : C_qK \to C_qL\}_{q\geq 0}$$

such that $\partial_q f_q = f_{q-1} \partial_q$ for all q. Observe that for every vertex map $f : K^{(0)} \to L^{(0)}$, we can obtain the corresponding chain map $f_\# : C_*K \to C_*L$ such that

$$f_\# \langle v_0, ..., v_q \rangle = \begin{cases} \langle f(v_0), ..., f(v_q) \rangle & \text{if } f(v_i) \text{ distinct} \\ 0 & \text{otherwise} \end{cases}$$

Let $h, k : C_*K \to C_*L$ be two chain maps. A *chain homotopy* from h to k is a family of homomorphisms

$$\{\phi_q : C_qK \to C_{q+1}L\}_{q\geq 0}$$

such that $\partial_{q+1}\phi_q + \phi_{q-1}\partial_q = h_q + k_q$. A *chain contraction* [EM52] from C_*K to C_*L consists of two chain maps, $f : C_*K \to C_*L$ and $g : C_*L \to C_*K$, and a chain equivalence $\phi : C_*K \to C_{*+1}K$ from gf to the identity map $1_{C_*K} : C_*K \to C_*K$; that is, ϕ satisfies that

$$1_{C_*K} + gf = \partial\phi + \phi\partial\,. \tag{1}$$

Moreover, it is required that

$$fg = 1_{C_*L}\,. \tag{2}$$

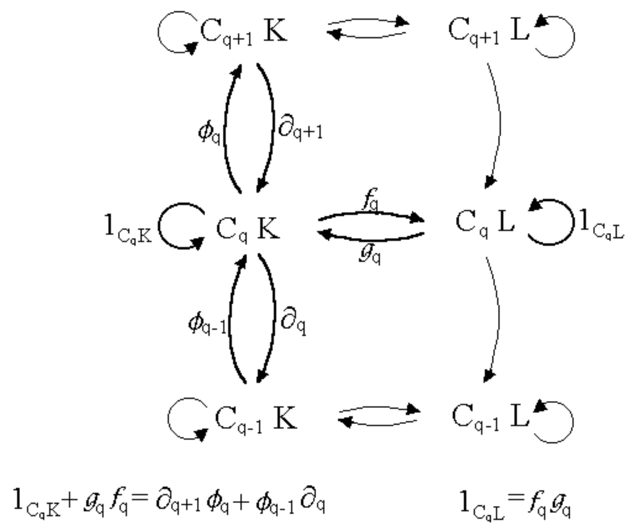

Figure 1: A chain contraction (ϕ, f, g) from C_*K to C_*L.

We denote such chain contraction as $(\phi, f, g) : C_*K \Rightarrow C_*L$. From (1), it is derived that H_*K is isomorphic to H_*L and, from (2), L has fewer or the same number of simplices than K. In this way, a chain contraction can be seen as an "homological thinning" of the starting object K. It is very intuitive to know what a contraction is when L is a subset of K. In this case, we can think of f as the way of partially destroying K to obtain L. On the other hand, the map ϕ can be seen as the way of reconstructing what we have destroyed. Note that a contraction can be defined between objects K and L that are not necessary simplicial complexes; for example, they can be simply graded sets. Moreover, in general, L is not a subset of K. We show two examples of contractions:

(a) Contraction to a Vertex.

Let $\sigma = \langle v_0, \dots, v_q \rangle$ be a simplex and let $K[\sigma]$ be the simplicial complex whose maximal simplex is σ. A contraction (ϕ, f, g) from $C_*K[\sigma]$ to $C_*\{\langle v_0 \rangle\}$ determining the acyclicity of the simplex σ, is defined as follows:

- $\phi : C_*K[\sigma] \to C_{*+1}K[\sigma]$ is given by $\phi\langle v_0, v_{j_1}, \dots, v_{j_n} \rangle = 0$ and $\phi\langle v_{j_1}, \dots, v_{j_n} \rangle = \langle v_0, v_{j_1}, \dots, v_{j_n} \rangle$ where $1 \leq j_1 < \cdots < j_n \leq q$.
- $f : C_*K[\sigma] \to C_*\{\langle v_0 \rangle\}$ is such that $f\langle v_i \rangle = \langle v_0 \rangle$ if $0 \leq i \leq q$ and $f\tau = 0$ otherwise.
- $g : C_*\{\langle v_0 \rangle\} \to C_*K[\sigma]$ is the inclusion.

(b) Edge Contractions.

Conditions under which edge contractions are homeomorphisms appear in [DEGN99]. Here, we show one property under which edge contractions become, at algebraic level, chain contractions.

Let K be a simplicial complex and let $\tau = \langle a, b \rangle$ be an edge in K. An *edge contraction* is given by the vertex map $f : K^{(0)} \to L^{(0)} = K^{(0)} - \{a, b\} \cup \{c\}$ where $f(a) = f(b) = c$, and $f(v) = v$ for all $v \neq a, b$.

If $Lk\, a \cap Lk\, b = Lk\, \tau$ then, a chain contraction $(\phi, f_\#, g)$ from C_*K to C_*L is defined as follows:

- $\phi : C_*K \to C_{*+1}K$ is given by $\phi\langle v_0, ..., v_q, b \rangle = \langle v_0, ..., v_q, a, b \rangle$ if $\langle v_0, ..., v_q \rangle \in Lk\, \tau$ and $\phi\tau = 0$ otherwise.

– $f_\#$ is the chain map induced by the vertex map f.
– $g : C_*L \to C_*K$ is such that

$$g\tau = \tau \qquad \forall \tau \notin St\, c,$$
$$g\langle c\rangle = \langle a\rangle,$$
$$g(\omega \cup \langle c\rangle) = \begin{cases} \omega \cup \langle a\rangle & \text{if } \omega \in Lk\, a, \\ \omega \cup \langle b\rangle + \bar{\omega} \cup \langle a, b\rangle & \text{if } \omega \in Lk\, b - Lk\, \tau \\ & \quad \text{and } \bar{\omega} \in Lk\, \tau \text{ is a facet of } \omega, \\ \omega \cup \langle b\rangle & \text{if } \omega \in Lk\, b - Lk\, \tau \\ & \quad \text{and no facet of } \omega \text{ belongs to } Lk\, \tau. \end{cases}$$

2.1 Incremental Homology Algorithm and Chain Contractions

Our algorithm for computing a chain contraction from the chain complex of a simplicial complex K to its homology is based on the incremental algorithm for computing the persistence of the Betti numbers developed in [ELZ00].

The input of the algorithm is the sorted set of all the simplices of K, $(\sigma_1, \ldots, \sigma_m)$, with the property that any subset of it, $\{\sigma_1, \ldots, \sigma_i\}$, $i \leq m$, is a simplicial complex itself. We call such an ordering a *filter*. Initially, we consider that $f\sigma_i = \phi\sigma_i = g\sigma_i = 0$ for $1 \leq i \leq m$ and $L = \emptyset$. Let $(\alpha_1, \ldots, \alpha_m)$ and $(\beta_1, \ldots, \beta_m)$ be two sorted sets of symbols. First, let us explain what this algorithm computes. The key idea of the algorithm is that any time we add a simplex σ_i, then a class of homology is created or destroyed. If $f\partial\sigma_i = 0$ then σ_i "creates" a new class of homology named α_i. Otherwise, $f\partial\sigma_i$ is a sum of classes of homology and, therefore, σ_i "destroys" one of the classes of homology involved in the expression $f\partial\sigma_i$. By convention, we say that σ_i destroys the youngest class of homology, that is, the class α_j in $f\partial\sigma_i$ with largest index j. The pseudocode of the algorithm is:

```
For i = 1 to i = m do
    if f∂σ_i = 0 then L = L ∪ {α_i},
                      φσ_i = β_i,
                      fσ_i = α_i,
                      gα_i = σ_i + φ∂σ_i;
    else let α_j be the youngest symbol in f∂σ_i then L = L − {α_j},
                                                      α_j = α_j + f∂σ_i,
                                                      β_j = σ_i + β_j + φ∂σ_i.
```

Now, we destroy the symbols $\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m$ in order to obtain a well–defined contraction from C_*K to C_*L:

```
For i = 1 to i = m do
    α_i = σ_i,
    β_i = 0.
```

Let us observe that L is not necessarily a simplicial complex. It is easy to check that (ϕ, f, g) is, in fact, a chain contraction from C_*K to C_*L, then K and L have isomorphic homology, so $H_*K \simeq C_*L$. Therefore, each simplex τ in L represents a class of homology of H_*K. Moreover, the representative cycle of the class τ is $a = g\tau = \tau + \phi\partial\tau$. Let us check that a is in fact a cycle:

$$\begin{aligned} \partial(\tau + \phi\partial\tau) &= \partial\tau + \partial\phi\partial\tau = \partial\tau + (gf + 1 + \phi\partial)\partial\tau = gf\partial\tau + \phi\partial\partial\tau \quad [\text{ since } \partial\partial\tau = 0, \text{ then }] \\ &= gf\partial\tau \quad [\text{ since, by construction, } f\partial\tau = 0, \text{ then }] = 0. \end{aligned}$$

Given a cycle b, if $fb = 0$ then b is also a boundary. In order to compute a chain b' such that $b = \partial b'$, we can use the relation

$$b + gfb = \phi\partial b + \partial\phi b\,.$$

Since $\partial b = 0$ and $fb = 0$, we have $b = \partial\phi b$ then, $b' = \phi b$.

An easy example of the computation of a chain contraction using the algorithm is the following. Let K be a simplicial complex whose set of maximal simplices is $\{\langle 1,3\rangle, \langle 1,4\rangle, \langle 2,3,4\rangle\}$. The output of the algorithm applied on a concrete filter of K is:

K	L	ϕ	f	g
$\langle 1\rangle$	$\langle 1\rangle$	0	$\langle 1\rangle$	$\langle 1\rangle$
$\langle 4\rangle$	$-$	$\langle 1,4\rangle$	$\langle 1\rangle$	$-$
$\langle 1,4\rangle$	$-$	0	0	$-$
$\langle 2\rangle$	$-$	$\langle 2,4\rangle + \langle 1,4\rangle$	$\langle 1\rangle$	$-$
$\langle 3\rangle$	$-$	$\langle 2,3\rangle + \langle 2,4\rangle + \langle 1,4\rangle$	$\langle 1\rangle$	$-$
$\langle 2,3\rangle$	$-$	0	0	$-$
$\langle 2,4\rangle$	$-$	0	0	$-$
$\langle 1,3\rangle$	$\langle 1,3\rangle$	0	$\langle 1,3\rangle$	$\langle 1,3\rangle + \langle 2,3\rangle + \langle 2,4\rangle + \langle 1,4\rangle$
$\langle 3,4\rangle$	$-$	$\langle 2,3,4\rangle$	0	$-$
$\langle 2,3,4\rangle$	$-$	0	0	$-$

Finally, let us analyze the complexity of the algorithm. Let $K = \{\sigma_1, \dots, \sigma_m\}$ and $d = dim\, K$. Suppose we are in the ith step of the algorithm. In the worst case, the number of simplices involved in $\partial\sigma_i$ is fewer or the same than the dimension of σ_i which is at most d and then, the number of elements involved in the formulas for $f\partial\sigma_i$ and $\phi\partial\sigma_i$ is $O(dm) = O(m)$. Since we have to replace the youngest class of homology in $f\partial\sigma_i$, the total cost of these operations is $O(m^2)$. Therefore, the total algorithm runs in time at most $O(m^3)$. Analogously, for obtaining the representative cycle a of any class of homology τ, we have to compute $\tau + \phi\partial\tau$. So, the cost of this is also $O(m^3)$.

3 The Cohomology Ring of a Simplicial Complex

One reason in order to use the cohomology instead of the homology, is that the cohomology has additional structures, such as the cup product and cohomology operations. If two spaces have isomorphic (co)homology groups but the behaviour of the ring structure or cohomology operations is different, then they are not homotopy equivalent. In this section we explain how to compute the cohomology ring of a simplicial complex K, starting from a chain contraction from C_*K to its homology. Using this information, we design a method for distinguishing non–homotopy equivalent simplicial complexes. We first need to define more concepts.

The *cochain complex* associated to K, denoted by C^*K, is the family

$$\{(C^qK, \delta^q)\}_{q\geq 0}\,,$$

defined in each dimension q by:

- The group $C^qK = Hom(C_qK; \mathbf{Z}_2)$=$\{c : C_qK \to \mathbf{Z}_2,\ c$ is a homomorphism$\}$.
- The homomorphism $\delta^q : C^qK \to C^{q+1}K$ called the *coboundary operator* given by

  $$\delta^q c\, a = c\, \partial_{q+1} a$$

  where $c \in C^qK$ and $a \in C_{q+1}K$.

The elements of C^qK are called *q–cochains*. Observe that a q–cochain can be defined on $K^{(q)}$ and it is naturally extended by linearity on C_qK. Z^qK and B^qK are the kernel of δ^q and the image of δ^{q-1}, respectively. The elements of Z^qK are called *q–cocycles* and those in B^qK are called *q–coboundaries*.

The coboundary operator satisfies that $\delta^q \delta^{q-1} = 0$. The qth *cohomology group* is defined for each integer q by

$$H^q K = Z^q K / B^q K \,.$$

Taking into account that the ground ring is a field, the homology and cohomology of K are isomorphic. Moreover, given a generator of homology, α, of dimension q, the corresponding generator of cohomology $\alpha^* : H_q K \to \mathbf{Z}_2$ can be defined:

$$\alpha^*(\alpha) = 1 \qquad \text{and} \qquad \alpha^*(\beta) = 0 \quad \text{for } \alpha \neq \beta \in H_q K \,.$$

Define the dual of chain maps and chain contractions, in the obvious way. The cohomology of K has an additional ring structure. Let $\alpha \in H^i K$ determined by an i–cocycle c and let $\beta \in H^j K$ determined by a j–cocycle c' then the *cup product* of α and β, $\alpha \smile \beta$, is the class of cohomology of dimension $i+j$ determined by the $(i+j)$–cocycle $c \smile c'$ defined by $(c \smile c')\sigma = (c\langle v_0, \dots, v_i\rangle) \cdot (c'\langle v_i, \dots, v_{i+j}\rangle)$, where $\sigma = \langle v_0, \dots, v_{i+j}\rangle \in K^{(i+j)}$ is such that $v_0 < \cdots < v_{i+j}$. In general, the cup product is commutative up to a sign. Since the group of coefficients is $\mathbf{Z}_2$, the cup product is, in fact, commutative.

Now, we show how to compute the cohomology ring of K using the chain contraction (ϕ, f, g) from C_*K to H_*K. Let us fix two integers I and J (we suppose $I \leq J$, since the cup product is commutative). Let $(\sigma_1, \dots, \sigma_p)$ be a basis for $H_I K$, $(\mu_1, \dots, \mu_q)$ a basis for $H_J K$ and $\mathcal{B} = (\gamma_1, \dots, \gamma_k)$ a basis for $H_{I+J} K$. For any two integers i and j (where $1 \leq i \leq p$ and $1 \leq j \leq q$), we have that

$$\sigma_i^* \smile \mu_j^* = \sum_{r=1}^{k} (((\sigma_i^* f) \smile (\mu_j^* f)) g \gamma_r)\, \gamma_r^* \,.$$

In the worst case, we have to do $k \cdot m \cdot p \cdot q$ elementary operations if K has m simplices, consequently, the cost of this function is $O(m^4)$. Notice that the resulting cohomology class $\sigma_i^* \smile \mu_j^*$ is determined by the cocycle $(\sigma_i^* f) \smile (\mu_j^* f)$. Having computed the multiplication tables of two cohomology rings, to determine whether or not the rings are isomorphic appears as an extremely difficult task. In order to avoid this problem, we consider the cup product as a linear operation instead of a bilinear one. More concretely, for fixed I and J, let V be the graded set of all the possible pairs (α_i^*, β_j^*) where $1 \leq i \leq p$ and $1 \leq j \leq q$. We impose that the degrees of the elements of V are $I + J$. Then, the cup product can be seen as a homomorphism $\smile : C_{I+J} V \to H^{I+J} K$. We express this homomorphism in a matrix form and the process of diagonalization of such matrix gives us detailed information about the kernel and image of this operation. Let us show the pseudocode of the algorithm for doing this. Let P be a set of integers and let $\gamma^* \in H^{I+J} K$. We first compute a function called changeBasis in order to obtain a simpler expression for the cup product.

```
changeBasis(γ, P, B)
 R = {r | γ_r ∈ B is a summand in γ};
 if R − R ∩ P is non empty then ℓ = min(R − R ∩ P),
                                γ_ℓ = γ,
                                P = P ∪ {ℓ}.
```

Now, for computing the cup product of any two classes of dimension I and J, we do:

```
cupProduct(I, J, K)
 P = { }, cup = { };
 if I = J then for i = 1 to i = p do
                    for j = i to j = p do
                         γ* = σ_i* ⌣ σ_j*,
                         changeBasis(γ, P, B)
                         if γ ≠ 0 then  cup = cup ∪ {((i, j), γ)};
```

```
else for i = 1 to i = p do
        for  j = 1 to j = q do
                γ* = σ*_i ⌣ μ*_j,
                changeBasis(γ, P, B)
                if γ ≠ 0 then  cup = cup ∪ {((i,j), γ)};
return (P, cup).
```

Since the cup product is commutative, in the case $I = J$, we only have to compute the products $\sigma_i^* \smile \sigma_j^*$ with $i \leq j$. Finally, it is easy to see that the complexity of this algorithm is $O(m^6)$ if K has m simplices.

Let us suppose that K and K' are two simplicial complexes of dimension d with isomorphic (co)homology groups. The goal now is to find out if K and K' are not homotopy equivalent. Let $(P_{(I,J,K)}, cup_{(I,J,K)})$ denote the output of cupProduct(I, J, K), then we simply have to study the number of elements of $P_{(I,J,K)}$ and $P_{(I,J,K')}$ for all the possible pairs (I, J):

```
testCup(I, J, K, K')
For I = 1 to I = d − 1 do
    for J = I to J = d − I do
        if |P_(I,J,K)| ≠ |P_(I,J,K')| then
            return K and K' are not homotopy equivalent.
```

An easy example applying of our method is showed. Consider the objects represented in Figure 2. We triangulate both objects (in order to obtain two simplicial complexes K and K' whose

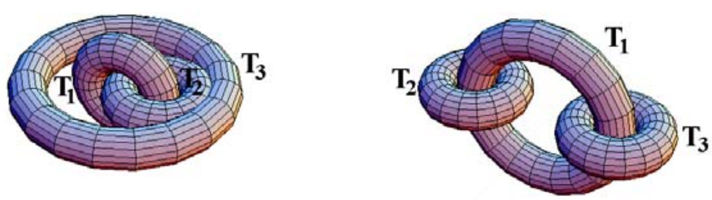

Figure 2: The object A and the object B.

geometric realizations are homomeomorphic to A and B, respectively), and apply our algorithm for computing a contraction to their homologies. We obtain that a basis for H_*K is $(\langle 1\rangle, \langle 2,19\rangle, \langle 4,19\rangle, \langle 12,29\rangle, \langle 14, 29\rangle, \langle 1,3,9\rangle, \langle 1,3,19\rangle, \langle 11,13,29\rangle)$ and a basis for H_*K' is $(\langle 21\rangle, \langle 2,9\rangle, \langle 9,23\rangle, \langle 12,19\rangle, \langle 21,27\rangle, \langle 1,3,9\rangle, \langle 9,21,23\rangle, \langle 11,13,19\rangle)$. We show in Figure 3 and Figure 4 the geometric realizations of K and K', respectively, and the cycles that represent the classes of homology of dimension 1 (the "tunnels" of A and B). Since both simplicial complexes have dimension 2, we only have to compute cupProduct $(1, 1, K)$ and cupProduct $(1, 1, K')$:

$$\begin{aligned}
\text{cupProduct } (1,1,K) &= (\{1,3\}, \{((1,2), \langle 1,3,9\rangle + \langle 1,3,19\rangle), ((3,4), \langle 11,13,29\rangle)\}) \\
\text{cupProduct } (1,1,K') &= (\{1,2,3\}, \{((1,4), \langle 1,3,9\rangle), ((2,4), \langle 9,2,23\rangle), ((3,4), \langle 11,13,19\rangle)\})
\end{aligned}$$

Since $|\{1,3\}| = 2 \neq 3 = |\{1,2,3\}|$, then A and B are not homotopy equivalent.

In order to obtain more powerful topological invariants derived from the cohomology ring structure, a basic tool in Algebraic Topology and Homological Algebra could be used: the bar construction of a ring [McL75]. More concretely, the method developed above can be seen, in fact, as the computation of the first non–null homology group of the simplicial bar construction of the cohomology ring of a simplicial complex K. Therefore, the design of algorithms computing the rest of the homology groups could allow us an adequate generalization of the method.

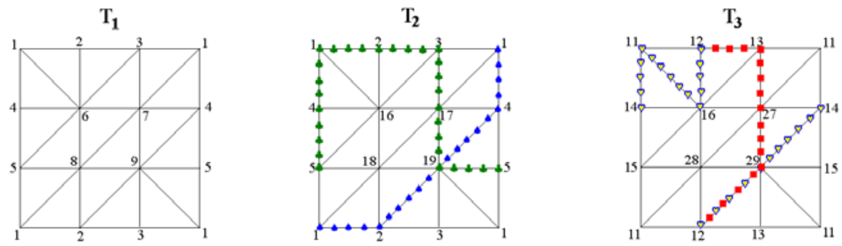

Figure 3: A triangulation of A and the cycles representing the tunnels of A.

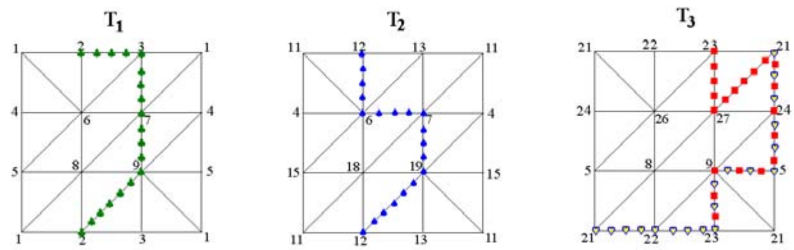

Figure 4: A triangulation of B and the cycles representing the tunnels of B.

4 Somme Comments

Let us note that in this paper the ground ring is $\mathbf{Z}_2$, but the processes described here work as well if the ground ring is any field.

Apart from allowing the computation of the cohomology ring, another argument for saving a chain contraction from the chain complex of a given simplicial complex K to its homology is the possibility of designing algorithms for calculating cohomology operations. A cohomology operation is a set map $\bar{\mathcal{O}} : H^m K \to H^n K$. The modus operandi for evaluating a cohomology operation $\bar{\mathcal{O}} : H^m K \to H^n K$ on a cohomology class $\alpha^* \in H^m K$ is the following:

1. Construct a chain contraction (ϕ, f, g) from $C_* K$ to $H_* K$ using our version of the incremental technique.

2. Evaluate $\bar{\mathcal{O}}$ on the cohomology class α^* using the diagram

$$\begin{array}{ccc} C^m K & \stackrel{g^*}{\leftarrow} & H^m K \\ \scriptstyle{\mathcal{O}}\downarrow & & \downarrow\scriptstyle{\bar{\mathcal{O}}} \\ C^n K & \stackrel{f^*}{\rightarrow} & H^n K\,, \end{array}$$

   where $\mathcal{O} : C^m K \to C^n K$ is a cochain operation associated to $\bar{\mathcal{O}}$ whose formulation is explicitly given in simplicial terms. Then, for obtaining $\bar{\mathcal{O}}\alpha^*$, compute $\mathcal{O}(\alpha^* f)g$.

For example, a combinatorial description $\mathcal{O}$ for $\bar{\mathcal{O}}$ being a Steenrod square, a Steenrod reduced power [GR99, GR02a] or some Adem secondary cohomology operations [GR01, GR02b] have already been done by the authors. More concretely, in [GR02b], an algorithm implemented in Mathematica for computing the Adem secondary cohomology operation $\Psi_2\alpha_2$ on any cohomology class $\alpha_2 \in KerSq^2H^1(K;\mathbf{Z_2})$ is explained.

From a practical standpoint, we emphasize that all the algorithms presented here have been implemented. For the particular case of objects embedded in $\mathbf{R}^3$, we have developed a software for visualizing the homology and the cohomology ring of 3D objects represented as simplicial complexes obtained from cubical decompositions (more details can be found in [BGLLR01]).

Finally, in order to get the action of any cohomology operation at cochain level on a representative cocycle, we have to compute it on a basis for C_*K on the desired dimension. In order to improve the efficiency of the algorithms, we could "topologically" thin the simplicial complex K. We obtain a "thinned" simplicial complex $M_{\text{top}}K$ such that there exists a chain contraction from C_*K to $C_*(M_{\text{top}}K)$. Two examples of thinning in this way are edge contractions (example (b) on page 4) and simplicial collapses [For99]. After doing this, we could apply our machinery to compute the cohomology ring or cohomology operations on $M_{\text{top}}K$ and the results could be easily interpreted in the "big" simplicial complex K via the chain contractions.

References

[BGLLR01] J.M. Berrio, R. González-Díaz, F. Leal, M.M López, P. Real. *Visualizing Cohomology Aspects of 3D Objects.* Proc. of the 6th Asian Tech. Conf. in Math. (2001), 459–468.

[DE95] C.J.A. Delfinado, H. Edelsbrunner. *An Incremental Alg. for Betti Numbers of Simplicial Complexes on the* 3*–Sphere.* Comput. Aided Geom. Design, v. 12 (1995), 771–784.

[DEGN99] T.K. Dey, H. Edelsbrunner, S. Guha, D.V. Nekhayev. *Topology Preserving Edge Contraction.* Publ. Inst. Math. (Beograd) (N.S.) v. 66 (1999), 23–45.

[DG98] T.K. Dey, S. Guha. *Computing homology groups of simplicial complexes in* R^3. Journal of ACM , vol. 45, No. 2, (1998), 266-287.

[ELZ00] H. Edelsbrunner, D. Letscher, A. Zomorodian. *Topological Persistence and Simplification.* Proc. 41st Ann. IEEE Sympos. Found. Comput. Sci (2000), 454–463.

[EM52] S. Eilenberg, S. MacLane. *On the Groups of* $H(\Pi, n)$*, I.* Annals of Mathematics, v. 58, n. 1 (1952), 55–107.

[EZ01] H. Edelsbrunner, A. Zomorodian. *Computing Linking Numbers in a Filtration.* Algorithms in Bioinformatics (LNCS 2149), Springer, Berlin (2001), 112–127.

[For99] R. Forman. *Combinatorial Differential Topology and Geometry.* New Perspective in Geometric Combinatorics. MSRI Publications, v. 8 (1999), 177–206.

[GR99] R. Gonzalez–Diaz, P. Real. *A Combinatorial Method for Computing Steenrod Squares.* J. of Pure and Applied Algebra, v. 139 (1999), 89–108.

[GR01] R. Gonzalez–Diaz, P. Real. *Computation of Cohomology Operations on Finite Simplicial Complexes.* Conf. on Alg. Top. Methods in Computer Sci. (Stanford, 2001). To appear in Homology, Homotopy and Applic.

[GR02a] R. González–Díaz, P. Real. *A Combinatorial Approach to Steenrod Reduced Powers.* 8th Confer. on Applic. of Computer Algebra (Volos, 2002). Sent to J. of Symbolic Comput.

[GR02b] R. González–Díaz, P. Real. *Geometric Objects and Cohomology Operations.* Proc. of the 5th Workshop on Computer Algebra in Scientific Computing (2002), 121-130.

[McL75] S. Mac Lane. *Homology.* Classics in Mathematics, Springer-Verlag, Berlin, 1995.

[Mun84] J.R. Munkres. *Elements of Algebraic Topology.* Addison–Wesley Co., 1984.

[Spa81] E.H. Spanier. *Algebraic Topology.* McGraw–Hill, 1966. Reprinted by Springer, 1981.