

Homology Map Computation with the CHomP Software

Paweł Pilarczyk

August 9, 2007

Abstract

This note provides some information on how to compute the homomorphism induced in homology by a continuous map between two cubical sets using combinatorial cubical multivalued maps and the programs available in the CHomP software package.

1 Introduction

The homomorphism induced in homology by a continuous map $f: (X, A) \rightarrow (Y, B)$, where X, Y, A, B are full cubical sets, is computed by the CHomP software with the use of a new idea developed especially for cubical sets [1]. This idea does not work for simplicial complexes, and there are no efficient algorithms for the computation of the homomorphism induced in homology by a continuous map between simplicial complexes. This shows an advantage of using cubical sets for scientific computations.

2 Combinatorial Cubical Multivalued Maps

A continuous map $f: X \rightarrow Y$ is represented by means of a multivalued map $\mathcal{F}: \mathcal{X} \multimap \mathcal{Y}$, where \mathcal{X} is a set of full cubes such that $X = |\mathcal{X}|$, \mathcal{Y} is a set of full cubes such that $Y = |\mathcal{Y}|$, and $\mathcal{F}(Q) \subset \mathcal{Y}$ for every $Q \in \mathcal{X}$. Such a map is called a *combinatorial cubical multivalued map*, or a *cubical map* for short. We say that the map \mathcal{F} is a *representation* of f if $f(Q) \subset |\mathcal{F}(Q)|$ for every $Q \in \mathcal{X}$.

The CHomP software accepts two different formats of text files for defining a cubical map. In the first format, each assignment $Q \mapsto \{Q_1, \dots, Q_n\}$ is defined in a separate line by listing all the image cubes in braces; for example, $(1, 2) \rightarrow \{(4, 1) (4, 2) (5, 1) (5, 2) (6, 1) (6, 2)\}$ means that $\mathcal{F}([1, 2] \times [2, 3]) = \{[4, 5] \times [1, 2], [4, 5] \times [2, 3], [5, 6] \times [1, 2], [5, 6] \times [2, 3], [6, 7] \times [1, 2], [6, 7] \times [2, 3]\}$, as illustrated in Figure 1. Another example is the following map which is defined on some rough cubical approximation of a circle.

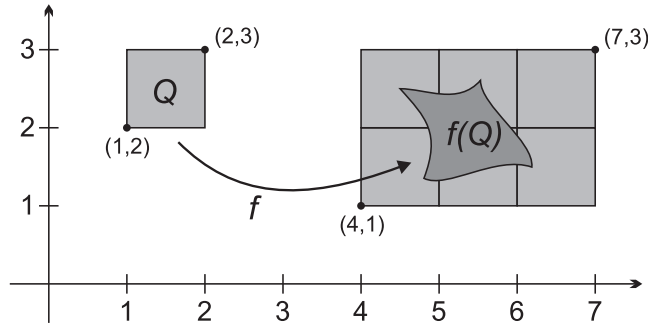


Figure 1: The image of a cube Q by a multivalued map \mathcal{F} that represents f must cover $f(Q)$.

Example 1 *A combinatorial cubical multivalued map*

- $(0,0) \rightarrow \{(0,0) (0,1)\}$
- $(0,1) \rightarrow \{(0,1) (0,2) (0,3)\}$
- $(0,2) \rightarrow \{(0,4) (1,4)\}$
- $(0,3) \rightarrow \{(2,4) (3,4)\}$
- $(0,4) \rightarrow \{(4,3) (4,4)\}$
- $(1,4) \rightarrow \{(4,1) (4,2) (4,3)\}$
- $(2,4) \rightarrow \{(3,0) (4,0)\}$
- $(3,4) \rightarrow \{(1,0) (2,0)\}$
- $(4,4) \rightarrow \{(0,0) (0,1)\}$
- $(4,3) \rightarrow \{(0,1) (0,2) (0,3)\}$
- $(4,2) \rightarrow \{(0,4) (1,4)\}$
- $(4,1) \rightarrow \{(2,4) (3,4)\}$
- $(4,0) \rightarrow \{(4,3) (4,4)\}$
- $(3,0) \rightarrow \{(4,1) (4,2) (4,3)\}$
- $(2,0) \rightarrow \{(3,0) (4,0)\}$
- $(1,0) \rightarrow \{(1,0) (2,0)\}$

The second format for defining a cubical map is suitable for the case in which the image of each cube is convex, which is equivalent to the fact that it fills an entire rectangular box. The image of a cube is then defined by specifying the two opposite vertices of this box (the one with minimal and maximal coordinates). The cube from the domain is also defined in the same way. In this format, the assignment discussed above is written as $[(1,2) (2,3)] [(4,1) (7,3)]$. As one can see, using this format saves a lot of space if the map has large convex images. At the beginning of the file that defines the map in this format some additional information must be provided: the space dimension, the number of assignments, and the type of the map. The following example shows the actual format of this information.

Example 2 *A combinatorial cubical multivalued map with convex images.*

Space Dimension: 2

Number Of Primitive Arguments: 16

Map: AlmostPerfect

| Primitive Argument | Value |
|--------------------|---------------|
| [(0,0) (1,1)] | [(0,0) (1,2)] |
| [(0,1) (1,2)] | [(0,1) (1,4)] |
| [(0,2) (1,3)] | [(0,4) (2,5)] |
| [(0,3) (1,4)] | [(2,4) (4,5)] |
| [(0,4) (1,5)] | [(4,3) (5,5)] |
| [(1,4) (2,5)] | [(4,1) (5,4)] |
| [(2,4) (3,5)] | [(3,0) (5,1)] |
| [(3,4) (4,5)] | [(1,0) (3,1)] |
| [(4,4) (5,5)] | [(0,0) (1,2)] |
| [(4,3) (5,4)] | [(0,1) (1,4)] |
| [(4,2) (5,3)] | [(0,4) (2,5)] |
| [(4,1) (5,2)] | [(2,4) (4,5)] |
| [(4,0) (5,1)] | [(4,3) (5,5)] |
| [(3,0) (4,1)] | [(4,1) (5,4)] |
| [(2,0) (3,1)] | [(3,0) (5,1)] |
| [(1,0) (2,1)] | [(1,0) (3,1)] |

END

3 Homology Computation via Graph Projections

The idea of computing the homomorphism induced in homology by a continuous map $f: X \rightarrow Y$, where X and Y are full cubical sets in \mathbf{R}^n , is to use the graph of the map. If G_f is a graph of the map f , then the projection $p: G_f \rightarrow X$ is a homeomorphism, so it induces an isomorphism in homology. Thus, we only need to compute the homomorphism induced by the projection $q: G_f \rightarrow Y$ in order to obtain the homomorphism induced in homology by f (see Figure 2).

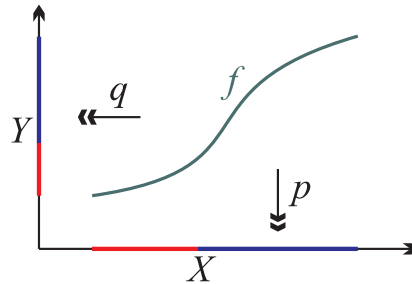


Figure 2: The projections p and q from the graph of f to its domain and codomain, respectively.

In the cubical approach to the computation of the homomorphism induced in homology by a continuous map $f: X \rightarrow Y$ between full cubical sets, the first step is to cover the graph of the map by full cubes, and in this way to obtain the full cubical set $G \subset X \times Y \subset \mathbf{R}^{2n}$. The cubical set G can also be understood as a graph of a combinatorial cubical multivalued map \mathcal{F} , where the image of a full cube $Q \subset X$ by \mathcal{F} is the union of all the full cubes in G contained in $Q \times Y$.

There are two simple programs in the CHomP software package which make the conversion between the two data structures in question. The program `cub2map` transforms a cubical set which is a graph of some cubical map (as explained above) into the cubical map in the form illustrated in Example 1. The program `map2cub` creates a graph of a cubical map, so it does the opposite conversion. These programs might be useful if one obtains the data from some scientific computations or experiment in the form of a graph of a cubical map and needs to compute the corresponding homomorphism induced in homology.

In fact, the procedure for the computation of the homomorphism induced in homology by a continuous map is much more sophisticated than described above. While creating the graph, it reduces it in a geometrical way so that it is much more efficient than the simple approach based on creating the graph and computing the projections (see Figure 3 for an idea, and [1] for details).

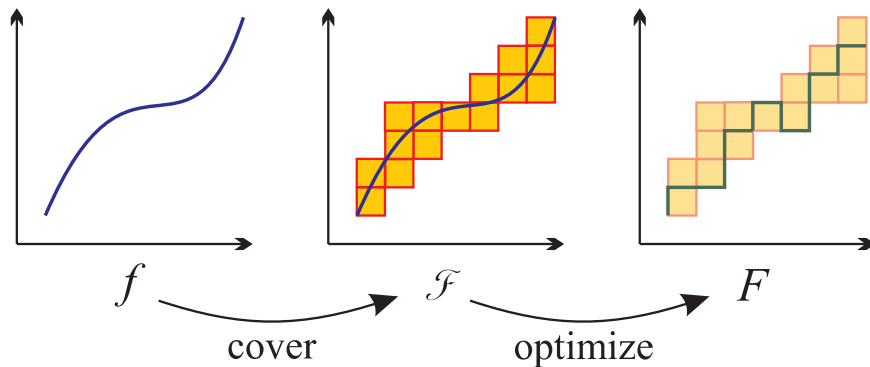


Figure 3: The idea of representing the graph of a continuous map with a cubical set for the purpose of homology computation.

4 Assumptions on the Cubical Map

The combinatorial cubical multivalued map $\mathcal{F}: (\mathcal{X}, \mathcal{A}) \multimap (\mathcal{Y}, \mathcal{B})$ must satisfy certain assumptions in order for the homology computation to be reliable. Apart from the obvious assumptions, like $\mathcal{F}(Q) \neq \emptyset$ or $\mathcal{F}(\mathcal{A}) \subset \mathcal{B}$, there is one crucial assumption which might not be easy to verify. It is the *acyclicity* assumption. It can be formulated as follows.

Definition 3 We say that a cubical map $\mathcal{F}: \mathcal{X} \multimap \mathcal{Y}$ is acyclic if for every set of cubes $\mathcal{R} \subset \mathcal{X}$ such that $\bigcap \mathcal{R} \neq \emptyset$, the set $|\mathcal{F}(\mathcal{R})|$ is acyclic.

The verification of this condition is rather lengthy, because all the subsets \mathcal{R} of the domain of the map \mathcal{F} which have a non-empty intersection must be analyzed, and the homology of their images must be computed and verified that it is the same as for a single point. In case of relative homology computation, both the map $\mathcal{F}: \mathcal{X} \multimap \mathcal{Y}$, as well as its restriction to \mathcal{A} , that is $\mathcal{F}_{\mathcal{A}}: \mathcal{A} \multimap \mathcal{B}$, must be acyclic.

The good news is that this assumption is automatically satisfied if the images of cubes by \mathcal{F} are convex, like in Example 2. This feature suggests that for the purpose of homology computation it is beneficial to work with this kind of maps.

There is a program `chkmvmap` (the name of this program is an abbreviation of “check multivalued map”) which verifies whether a given map satisfies all the necessary assumptions for the homology computation, including the acyclicity. This program should be called with the same arguments as the program `homcubes`, for example,

```
chkmvmap F.map X.cub A.cub Y.cub B.cub
```

5 Computations with the Program `homcubes`

The program `homcubes` contained in the CHomP software package is capable of computing the homomorphism induced in homology by a combinatorial cubical multivalued map satisfying the assumptions described above. If relative homology must be computed, that is, we consider the map $\mathcal{F}: (\mathcal{X}, \mathcal{A}) \multimap (\mathcal{Y}, \mathcal{B})$, then the program `homcubes` must be called with five arguments, the names of files that contain the text definitions of \mathcal{F} , \mathcal{X} , \mathcal{A} , \mathcal{Y} , and \mathcal{B} , in this order. For plain homology, that is, if $\mathcal{A} = \mathcal{B} = \emptyset$, the program `homcubes` must be called with three arguments, namely, \mathcal{F} , \mathcal{X} , and \mathcal{Y} .

Just like in the case of homology computation of cubical sets, the program `homcubes` is also capable of saving homology generators to files. This is achieved by adding the `-g` switch to the program call, followed by the name of a file to which the generators are to be saved. This argument can be given up to three times, for instance,

```
homcubes F.map X.cub A.cub Y.cub B.cub -g XA.gen -g YB.gen -g F.gen
```

In this example, the homology generators of (X, A) are saved to the file `XA.gen`, the homology generators of (Y, B) are saved to `YB.gen`, and the homology generators of the graph of the map (which might be useful in some cases) are saved to `F.gen`.

6 The Conley Index Map

While computing the homomorphism induced in homology by a continuous map $f: (X, A) \rightarrow (Y, B)$ using its combinatorial cubical multivalued representation

$\mathcal{F}: (\mathcal{X}, \mathcal{A}) \multimap (\mathcal{Y}, \mathcal{B})$, in some cases there is the additional property that the inclusion $i: (X, A) \rightarrow (Y, B)$ induces an isomorphism in homology. This essentially means that the pair (Y, B) is some kind of an extension of the pair (X, A) . This happens, for example, during the computation of the Conley index. In that case the homology generators of (X, A) can also serve as the homology generators of (Y, B) , so we would like to compute the composition of the homomorphism induced by f with the inverse of the homomorphism induced by i :

$$(i_*)^{-1} \circ f_*: H_*(X, A) \rightarrow H_*(X, A).$$

This useful operation is incorporated in the program `homcubes`, and can be used by specifying the `-i` flag on the command line. In this case, the program `homcubes` computes the two homomorphisms f_* and i_* , then inverts i_* (it stops with a message if it fails), and composes the two homomorphisms.

The program `chkmvmap` for the verification of assumptions which a cubical map must satisfy can also be given the flag `-i` to verify a few additional obvious properties of the map, e.g. if $\mathcal{A} \subset \mathcal{B}$. However, this program does not check if the inclusion induces an isomorphism in homology. This is only checked by the program `homcubes` during the homology computation.

7 Computing the Inclusion Map

In order to compute the homomorphism induced by an inclusion map in homology for full cubical sets, one can use the program `cub2map` to create a representation of the inclusion map, and then compute the homomorphism in homology with the program `homcubes`. Here is an example for $X \subset Y$:

```
cub2map -i X.cub incl.map
homcubes incl.map X.cub Y.cub
```

References

- [1] K. MISCHAIKOW, M. MROZEK, P. PILARCZYK, *Graph approach to the computation of the homology of continuous maps*, Foundations of Computational Mathematics 5 (2005) 199–229.