# A Kenzo Interface for Algebraic Topology Computations in SageMath*

Julián Cuevas-Rozo[1], Jose Divasón[1], Miguel Marco-Buzunáriz[2], and Ana Romero[1]

[1]Department of Mathematics and Computer Science, University of La Rioja
{jucuevas,jose.divason,ana.romero}@unirioja.es
[2]Department of Mathematics, University of Zaragoza
mmarco@unizar.es

**Abstract**

In this work we present an interface between SageMath and Kenzo, together with an optional Kenzo package. Our work makes it possible to communicate both computer algebra programs and enhances the SageMath system with new capabilities in algebraic topology, dealing in particular with simplicial objects of infinite nature.

## 1 Introduction

SageMath [6] is a general purpose computer algebra system. It was created by William Stein in 2005 with the mission statement of "Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab"; and since then, it has grown to a big international project, with hundreds of developers all around the world, and keeps growing at a fast pace: new versions are released every few months, with bugfixes and new functionalities.

It follows the philosophy of "building the car instead of reinventing the wheel", in the sense that it includes many other libraries and systems developed for specific purposes, and makes them work together under a common abstraction layer. Besides that, specific code is written directly in SageMath for many computations, when there is no suitable available system that can take care of that.

In particular, in the realm of algebraic topology, SageMath includes modules for simplicial complexes, $\Delta$-complexes, cubical complexes, and simplicial sets (plus other topological objects of different nature, such as knots and braids for example). It can compute invariants such as the homology of these objects, both by using custom written code and by leveraging an interface to CHomP [4]. Sadly, for the moment SageMath only supports objects of finite type. However, the computer algebra system Kenzo [1] can deal with more general simplicial sets, so it would be a perfect candidate to extend the SageMath functionalities in this aspect.

The work we present here is an interface between SageMath and Kenzo, together with an optional Kenzo package for SageMath. Its importance is twofold: on one hand, it allows to install and use Kenzo within SageMath (which allegedly can be easier than a standalone version of Kenzo), making it easy to use Kenzo as part of a wider workflow. On the other hand, this interface can be seen as a first step towards an extension of SageMath capabilities around simplicial sets. Further work is planned by the authors in this direction.

*J. Cuevas-Rozo, J. Divasón and A. Romero are partially supported by the Spanish Ministry of Science, Innovation and Universities, project MTM2017-88804-P. M. Marco-Buzunáriz is partially supported by MTM2016-76868-C2-2-P and Grupo "Investigación en Educación Matemática" of Gobierno de Aragón/Fondo Social Europeo.

A first version of the optional package and the interface for Kenzo (containing some of the basic functionalities of the system) were already submitted to the SageMath development process (which is peer reviewed) and have been accepted, so they are available from version 8.7 onwards.

## 2 The Kenzo system for algebraic topology

Kenzo is a symbolic computation system devoted to algebraic topology, allowing in particular the computation of homology and homotopy groups of complicated spaces, such as iterated loop spaces of a loop space modified by a cell attachment, components of complex Postnikov towers, etc., which were not known before. The main goal of Kenzo is to be able to deal with topological spaces of infinite nature; homology and homotopy groups of these spaces are then computed by means of the *effective homology* method, introduced in [5] and deeply explained in [3].

The effective homology method is based on the following idea: on the one hand, if a topological space $X$ has finite nature, then its homology groups can be determined by means of elementary operations on matrices. On the other hand, if $X$ is of infinite type, then Kenzo should be able to find a homotopy equivalence between a chain complex canonically associated with $X$, $C_*(X)$, and a chain complex $E_*$ of finite type, whose homology groups can be determined. This particular homotopy equivalence (which must satisfy some special properties, see [3] for details) makes it possible to determine the homology groups of $X$ by means of those of $E_*$, providing not only a formal description of the groups but also their generators.

The computation of homotopy groups of spaces is obtained in Kenzo by means of the Whitehead tower method [7], constructing again the effective homology of some spaces. Moreover, Kenzo has been enhanced by the first and fourth authors of this paper with several modules computing spectral sequences, discrete vector fields, homology of groups, persistent homology and invariants of finite topological spaces.

The implementation of spaces of infinite nature and the effective homology method are done in Kenzo by means of the Common Lisp programming language, making an intensive use of functional programming. The Kenzo system requires then the use of a Common Lisp IDE and the user should be familiarized with functional programming, which can be considered a tedious task. The integration of Kenzo into SageMath, allowing the use of Kenzo constructions and operations (in particular those of infinite nature) by means of SageMath syntax, makes it possible to use Kenzo in a friendlier way and the enhancement of SageMath with algebraic topology constructions which up-to-now were not available in the system.

## 3 The SageMath-Kenzo interface

The work we present here consists of two parts: an optional package to install Kenzo within SageMath, and an interface to communicate between SageMath and the Kenzo instance installed with the package.

We can install the Kenzo optional package in SageMath by running:

```
sage -i kenzo
```

This will download the source code of a fork of Kenzo that we maintain for this specific purpose, compile it inside the embedded Common Lisp interpreter that SageMath includes, and install it. Allegedly, this can be an easy way to install Kenzo even if we just plan to use it as a standalone program.

Once done that, we can also communicate with Kenzo from a SageMath session by using the interface. Notice that, since Kenzo is installed as a package inside the Embeddable Common Lisp library, the communication between the two systems is done via a C-library interface, which has a much lower overhead than other alternatives, such as pseudo-terminals or temporary files. The interface provides functions to create Kenzo objects, and wrappers around them that allow to create new objects (with the corresponding wrapper) and call functions on them. For example, if we want to compute the homology groups of the cartesian product of two Eilenberg–MacLane spaces, we can do the following:

```
from sage.interfaces.kenzo import EilenbergMacLaneSpace
sage: E1 = EilenbergMacLaneSpace(ZZ, 3)
sage: E2 = EilenbergMacLaneSpace(AdditiveAbelianGroup([2]),2)
sage: E = E1.cartesian_product(E2)
sage: [E.homology(i) for i in range(8)]
[Z, 0, C2, Z, C4, C2 x C2 x C2, C2, C2 x C2 x C3 x C4]
```

Let us observe that the space $E$ is of infinite nature, but Kenzo is able to compute its homology groups.

The Kenzo interface available in the current SageMath distribution includes for the moment only some basic functionalities. In particular, as of today, the available interface exposes functions to create the following spaces:

- spheres
- Eilenberg–MacLane spaces
- Moore spaces

And from these spaces, it allows the following constructions:

- cartesian products
- classifying spaces
- loop spaces
- suspensions
- tensor products

For each resulting space, the `homology` method can construct the corresponding homology group.

In a second version of the package and interface (whose development can be found in `https://trac.sagemath.org/ticket/27880`), we have included the Kenzo functions for computing homotopy groups of simplicial sets (by using the Whitehead tower method). In addition, we have developed methods for translating objects such as chain complexes and simplicial sets both from SageMath to Kenzo and vice versa. In this way, we can build simplicial objects in SageMath (by using a package by J. Palmieri), translate them into Kenzo and compute their homology and homotopy groups. It is worth noting that the package by J. Palmieri already contains a function to construct simplicial sets given a Kenzo output, but it is restricted to some particular cases and is based on external Kenzo files, i.e., there is no proper connection between both systems (see [2] for further details).

In the next lines we consider as an example the computation of homotopy groups of a wedge of the spheres $S^3$ and $S^4$.

```
sage: from sage.homology.simplicial_set import *
sage: from sage.interfaces.kenzo import *
sage: s3 = simplicial_sets.Sphere(3)
sage: s4 = simplicial_sets.Sphere(4)
sage: s3vs4 = s3.wedge(s4)
sage: K = KFiniteSimplicialSet(s3vs4)
sage: [K.homotopy_group(i) for i in range(2,6)]
[Trivial Abelian group,
```

```
 Multiplicative Abelian group isomorphic to Z,
 Multiplicative Abelian group isomorphic to Z x C2,
 Multiplicative Abelian group isomorphic to C2 x C2]
```

Let us note that both systems are collaborating for the task: the wedge product of the spheres is computed by means of a function that is available in SageMath but not in Kenzo, whereas the computation of the homotopy groups of that simplicial object is carried out via Kenzo, since the simplicial set module provided by SageMath is not able to deal with that computation.

## 4 Conclusions and further work

In this work we have presented an interface between SageMath and Kenzo, together with an optional Kenzo package. The first version includes some basic functionalities of algebraic topology computations and is already available from version 8.7 of SageMath. The computation of homotopy groups and the translation of chain complexes and simplicial sets from SageMath to Kenzo and vice versa have been considered in the second version, which will be submitted to the SageMath development process in the near future.

Other algebraic topology constructions are implemented as additional modules in Kenzo and are not available in SageMath, such as spectral sequences, discrete vector fields and finite topological spaces. The integration of these constructions in SageMath requires to adapt the Kenzo Common Lisp code to the Embeddable Common Lisp library in SageMath and to develop the corresponding wrappers and necessary functions in SageMath. This is an on-going work that has been started recently, in which the spectral sequence module can be viewed as a first step.

## References

[1] X. Dousson, J. Rubio, F. Sergeraert, and Y. Siret. The Kenzo program. Institut Fourier, Grenoble, 1999. `http://www-fourier.ujf-grenoble.fr/~sergerar/Kenzo/`.

[2] J. Palmieri. Examples of simplicial sets – Sage reference manual v8.7, 2019. `https://bit.ly/2LzgaBB`.

[3] J. Rubio and F. Sergeraert. Constructive Homological Algebra and Applications, Lecture Notes Summer School on Mathematics, Algorithms, and Proofs. University of Genova, 2006. `http://www-fourier.ujf-grenoble.fr/~sergerar/Papers/Genova-Lecture-Notes.pdf`.

[4] S. Harker et al. *The Computational HOMology Project*, 2009. `www.chomp.rutgers.edu`.

[5] F. Sergeraert. The computability problem in Algebraic Topology. *Advances in Mathematics*, 104(1):1–29, 1994.

[6] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 8.7.0)*, 2019. `https://www.sagemath.org`.

[7] G. Whitehead. Fiber spaces and the Eilenberg homology groups. *Proceedings of the National Academy of Science of the United States of America*, 38(5):426–430, 1952.