CAPD::RedHom v2 - Homology Software Based on Reduction Algorithms

Mateusz Juda and Marian Mrozek*

Institute of Computer Science and Computational Mathematics,
Jagiellonian University, Poland
`mateusz.juda@uj.edu.pl`
`http://redhom.ii.uj.edu.pl/`

Abstract. We present an efficient software package for computing homology of sets, maps and filtrations represented as cubical, simplicial and regular CW complexes. The core homology computation is based on classical Smith diagonalization, but the efficiency of our approach comes from applying several geometric and algebraic reduction techniques combined with smart implementation.

Keywords: Homology software, homology algorithms, Betti numbers, homology groups, homology generators, homology maps, persistent homology, cubical sets, simplicial complexes, CW complexes.

1 Introduction

In 1995 M. Mrozek and K. Mischaikow presented a computer assisted proof of the existence of chaotic dynamics in Lorenz equations [10,11]. The computer programs needed for the proof became the seed of the software package developed by members of the CAPD (Computer Assisted Proofs in Dynamics) group [21]. Throughout the years the package became a reach collection of software libraries and tools for rigorous numerics of dynamical systems (see [9] for the description of the mainstream CAPD package).

An important ingredient of the mentioned proof is Conley index, a homological invariant of dynamical systems. The computer assisted proofs based on Conley index brought interest in cubical homology theory [7] and stimulated the development of the homology package for the needs of computer assisted proofs. Since 2005 the homology software for CAPD has been developed jointly with the Computational Homology Project (CHomP) [23].

After having implemented the classical algorithm based on Smith diagonalization it became clear that it is much too slow for the needs of computer assisted proofs. This originated the search for faster homology algorithms.

CAPD::RedHom is a software package for efficient homology computations of cubical and simplicial complexes as well as some special cases of regular CW

---

* This research is partially supported by the Polish National Science Center under grant 2012/05/N/ST6/03621 and by the TOPOSYS project FP7-ICT-318493-STREP.

H. Hong and C. Yap (Eds.): ICMS 2014, LNCS 8592, pp. 160–166, 2014.

complexes. Originally, the software was designed for applications in rigorous numerics of Topological Dynamics. Such applications, based on interval arithmetic, lead in a natural way to cubical sets. They may be represented very efficiently as bitmaps. The cubical sets arising from the algorithms in dynamics usually are strongly inflated in the sense the sets which much smaller representation have the same topology or homotopy type. Such small representations may be found in linear time by various geometric reduction techniques. The algebraic invariants of topology, in particular homology, are then computed for the small representation. This leads to a very significant speed up. In particular, the expensive, linear algebra computations, such as Smith diagonalization, are performed on small data.

The package was developed by: P. Brendel, P. Dlotko, G. Jablonski, M. Juda, A. Krajniak, M. Mrozek, P. Pilarczyk, H. Wagner, N. Zelazna.

2 Functionality

The CAPD::RedHom software package, which is currently under intensive development, constitutes a redesign of the CAPD homology software. It is based on the already mentioned as well as the very recent reduction ideas proposed in [1,4,5,6,14,17]. It is designed to meet the needs of various areas of applications, to apply to cubical and simplicial sets as well as CW complexes and at the same time to maintain the efficiency of the original CAPD software for cubical sets. This is achieved by applying the techniques of static polymorphism based on C++ templates so that the reduction algorithms may be applied to various representations of sets without any overhead run-time costs. An unwanted side effect is that this makes the code very hard to use as a library or a plug-in. For this reason recently we put a lot of effort to make the efficient C++ code accessible in external, commonly used libraries. Presently, the code is available as a plug-in for *GAP* [24], *Python*, and *Sage* [25].

The package is intended both for users who are interested in stand-alone programs as well as programmers who want to use the library in their programs. The ultimate goal is that the package will provide Betti numbers, torsion coefficients, homology generators and matrices of maps induced in homology. Moreover, for filtered sets the package will provide persistence intervals [3].

3 Applications

The original CAPD homology software was written for applications in rigorous numerics of dynamical systems. However, the range of applicability of homology software encompasses several other areas: electromagnetism, image analysis, visualization, data mining, sensor networks, robotics and many others. Although the general goal is the same, these areas differ in details of input and output. The cubical representation of sets is convenient in dynamics, because interval arithmetic used in computer assisted proofs in the theory of dynamical systems leads in a natural way to such sets. It is also convenient in the analysis of raster images. However, in many situations the simplicial representation is more natural. In electromagnetics and in all cases when sets exhibit fractal structure a general CW complex representation is most convenient.

Apart from the original applications in rigorous numerics of dynamical systems [13], so far the package has found applications in image analysis [18], material science [17,1], electromagnetism [2], and group representation theory [20].

4 Underlying Theory

One way of avoiding the supercubical cost of the classical homology algorithm is decreasing the size of the input to Smith algorithm without changing the homology. Such an approach was first proposed in [8] by means of a linear time reduction of chain complexes. The reduction process considered in that paper is purely algebraic and may be viewed as a method of limiting the fill-in process in the Smith diagonalization.

However, reductions may be performed directly on the level of the topological space. At first, this may look like acting against the fundamentals of algebraic topology. Algebraic topology solves problems in topology by translating them to the ground of much simpler algebra. But, experiments indicate that in many applications doing geometric reductions directly at the topological level instead of algebraic reductions after translating the problem from topology to algebra may significantly speed up the computations. Also, such an approach often uses significantly less memory.

The first implemented algorithm of this type is based on the observation that for a cube $Q \subset X$, if $Q \cap \mathrm{cl}(X \setminus Q)$ is acyclic then X can be replaced by $\mathrm{cl}(X \setminus Q)$ without affecting the homology (see [19]). This fact was used in the reduction techniques proposed in [12] and motivated the Acyclic Subspace Homology Algorithm (see [15]), based on the construction of a possibly large acyclic subspace A of the topological space X. The computation of the homology groups $H(X)$ reduces then to the computation of $H(X \setminus A)$ in the sense of one space homology theory (see [16]). The method is particularly useful for cubical subsets of $\mathbb{R}^n$ with $n \in \{2, 3\}$, because in these dimensions the acyclic subspace may be constructed extremely fast due to the possibility of storing all possible neighborhood configurations and using them as look-up tables for testing the acyclicity.

The simplest example of reductions on the topological level are free face collapses proposed in [7]. Unfortunately, in many situations free faces are quickly exhausted and the remaining set is still large. Significantly deeper reductions in low dimensions may be achieved by means of the dual concept of free cofaces. This idea leads to the Coreduction Homology Algorithm [16].

The Acyclic Subspace Homology Algorithm and the Coreduction Homology Algorithm together with Discrete Morse Theory [4] seem to be the fastest homology algorithms for inflated cubical and simplicial sets available so far. In particular, they outperform algebraic homology algorithms just because they run in a fraction of time needed to translate the problem to algebra.

5 Technical Contribution

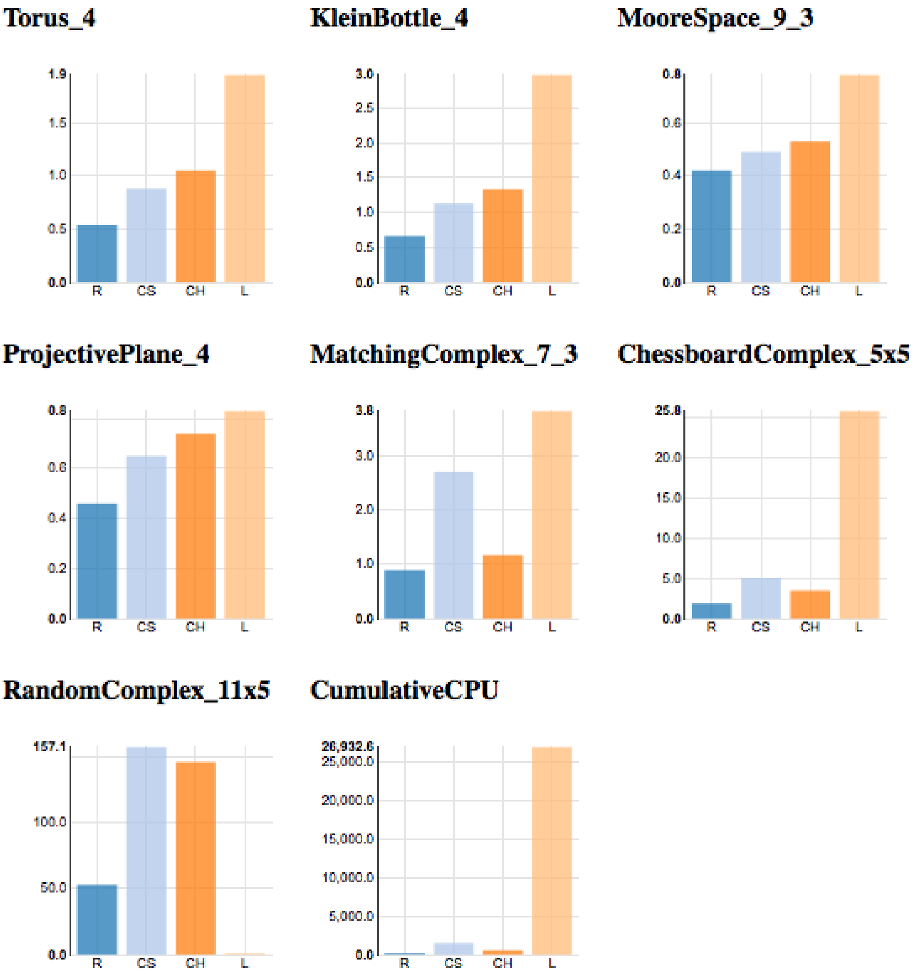
Algorithms implemented in the CAPD::RedHom package behave incredibly well for inflated data sets. We see such sets especially in applications, where a continuous problem is translated into a combinatorial problem. Among many examples, there is a common pattern: to achieve sufficient theoretical conditions for the discretization, we need to subdivide our space. That operation do not change homology, but increase data size.

We compared CAPD::RedHom with latest CHomP [23] (programs *homsimpl* and *chomp-simplicial*) and Linbox [26] (program *homology_gap* 1.4.3 used in GAP [24] - we cannot use latest version, GPC compiler removed from Ubuntu/Debian in 2011). For the comparison we generated simplicial complexes using Sage [25]. For classical examples available in module `sage.simplicial_complexes` we generate their subdivisions with `subdivide()` routine. Using various parameters we generated 380 input files. We will present detailed list of examples in the full paper. For the purpose of this article, on the Figure 1 we show CPU usage for following complexes:

- `Torus()` with 4 subdivisions, 18144 2-dimensional simplices on input;
- `KleinBottle()` with 4 subdivisions, 20736 2-dimensional simplices on input;
- `MooreSpace(9)` with 3 subdivisions, 13176 2-dimensional simplices on input;
- `ProjectivePlane()` with 4 subdivisions, 12960 2-dimensional simplices on input;
- `MatchingComplex(7)` with 3 subdivisions, 22680 2-dimensional simplices on input;
- `ChessboardComplex(5,5)` with 1 subdivision, 14400 4-dimensional simplices on input;
- `RandomComplex(11,5)` with 1 subdivision, 172680 5-dimensional simplices on input;

The number of subdivisions in each case is big enough to force non-instant computations. The case `RandomComplex(11,5)` emphasize benefits from our approach: CAPD::RedHom is almost three times faster than CHomP and Linbox cannot finish computations in one hour. On the chart *CumulativeCPU* we presents total CPU usage by each program in the experiment.

During development of the CAPD::RedHom package we faced a lot of interesting technical and theoretical problems. The most important challenge in our applications is in data set size. The biggest set computed so far contains 10^9 simplices in dimension $0 - 3$ ($600 \cdot 10^6$ facets) [20]. The set required 3 days of computations on a machine with 512 GB of RAM. Big data sets in applications convinced us to start implementations of our algorithms for parallel and distributed computations. This is a big challenge in the area of computational homology. In the full paper we will show our progress in this subject.


Torus_4

KleinBottle_4

MooreSpace_9_3

ProjectivePlane_4

MatchingComplex_7_3

ChessboardComplex_5x5

RandomComplex_11x5

CumulativeCPU

Fig. 1. CPU usage charts. On each picture from left: R (CAPD::RedHom), CS (CHomP - chomp-simplicial), CH (CHomP - homsimpl), L (Linbox - homology_gap).

References

1. Dłotko, P., Kaczynski, T., Mrozek, M., Wanner, T.: Coreduction Homology Algorithm for Regular CW-Complexes. Discrete and Computational Geometry 46, 361–388 (2011), doi:10.1007/s00454-010-9303-y
2. Dłotko, P., Specogna, R., Trevisan, F.: Automatic generation of cuts on large-sized meshes for the T-Omega geometric eddy-current formulation. Computer Methods in Applied Mechanics and Engineering 198, 3765–3781 (2009)
3. Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological Persistence and Simplification. Discrete and Computational Geometry 28, 511–533 (2002)
4. Harker, S., Mischaikow, K., Mrozek, M., Nanda, V., Wagner, H., Juda, M., Dłotko, P.: The Efficiency of a Homology Algorithm based on Discrete Morse Theory and Coreductions. In: Diaz, R.G., Jurado, P.R. (eds.) Proceedings of the 3rd International Workshop on Computational Topology in Image Context, Chipiona, Spain, Image A, vol. 1, pp. 41–47 (November 2010) ISSN: 1885-4508
5. Juda, M., Mrozek, M.: Z_2-Homology of weak $(p-2)$-faceless p-manifolds may be computed in $O(n)$ time. Topological Methods in Nonlinear Analysis 40, 137–159 (2012)
6. Kaczynski, T., Dłotko, P., Mrozek, M.: Computing the Cubical Cohomology Ring. In: Diaz, R.G., Jurado, P.R. (eds.) Proceedings of the 3rd International Workshop on Computational Topology in Image Context, Chipiona, Spain, Image A, vol. 3, pp. 137–142 (2010) ISSN: 1885-4508
7. Kaczynski, T., Mischaikow, M., Mrozek, M.: Computational Homology. Applied Mathematical Sciences, vol. 157. Springer-Verlag (2004)
8. Kaczynski, T., Mrozek, M., Ślusarek, M.: Homology computation by reduction of chain complexes. Computers and Math. Appl. 35, 59–70 (1998)
9. Kapela, T., Mrozek, M., Pilarczyk, P., Wilczak, D., Zgliczyński, P.: CAPD - a Rigorous Toolbox for Computer Assisted Proofs in Dynamics, technical report, Jagiellonian University (2010)
10. Mischaikow, K., Mrozek, M.: Chaos in Lorenz equations: a computer assisted proof. Bull. AMS (N.S.) 33, 66–72 (1995)
11. Mischaikow, K., Mrozek, M.: Chaos in the Lorenz equations: a computer assisted proof. Part II: details. Mathematics of Computation 67, 1023–1046 (1998)
12. Mischaikow, K., Mrozek, M., Pilarczyk, P.: Graph approach to the computation of the homology of continuous maps. Found. Comp. Mathematics 5, 199–229 (2005)
13. Mrozek, M.: Index Pairs Algorithms. Found. Comp. Mathematics 6, 457–493 (2006)
14. Mrozek, M.: Čech Type Approach to Computing Homology of Maps. Discrete and Computational Geometry 44(3), 546–576 (2010), doi:10.1007/s00454-010-9255-2.
15. Mrozek, M., Pilarczyk, P., Żelazna, N.: Homology algorithm based on acyclic subspace. Computers and Mathematics with Applications 55, 2395–2412 (2008)
16. Mrozek, M., Batko, B.: Coreduction homology algorithm. Discrete and Computational Geometry 41, 96–118 (2009)
17. Mrozek, M., Wanner, T.: Coreduction homology algorithm for inclusions and persistent homology. Computers and Mathematics with Applications 60(10), 2812–2833 (2010), doi:10.1016/j.camwa.2010.09.036
18. Mrozek, M., Żelawski, M., Gryglewski, A., Han, S., Krajniak, A.: Homological methods for extraction and analysis of linear features in multidimensional images. Pattern Recognition 45, 285–298 (2012)
19. Pilarczyk, P.: Computer assisted method for proving existence of periodic orbits. Topological Methods in Nonlinear Analysis 13, 365–377 (1999)

20. Koonin, J.: Topology of eigenspace posets for imprimitive reflection groups, `http://arxiv.org/abs/1208.4435`
21. Computer Assisted Proofs in Dynamics, `http://capd.ii.uj.edu.pl/`
22. Reduction Homology Algorithms, `http://redhom.ii.uj.edu.pl/`
23. Computational Homology Project, `http://chomp.rutgers.edu/`
24. GAP System, `http://www.gap-system.org/`
25. Sage, `http://www.sagemath.org/`
26. Linbox, `http://www.eecis.udel.edu/~dumas/Homology/`