CAPD::RedHom - Reduction heuristics for homology 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/`

July 15, 2014

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, cubical sets, simplicial complexes, CW complexes.

1 Introduction

CAPD::RedHom is a software package for efficient homology computations of cubical and simplicial complexes as well as some special cases of regular CW 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: Piotr Brendel, Pawel Dlotko, Grzegorz Jablonski, Mateusz Juda, Andrzej Krajniak, Marian Mrozek, Pawel Pilarczyk, Hubert Wagner, Natalia Zelazna.

2 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 [14], so far the package has found applications in image analysis [19], sensor networks [1], material science [18, 2] and electromagnetism [3], representation theory [21].

---

[*] 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.

3 Implementation

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 [2, 5, 6, 7, 15, 18]. 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 System* [25], *Python* language module, and *Sage* [26].

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 [4].

4 Recent research

The software development is mostly driven by homolgy applications in challenging problems in theoretical and applied science. It is worth to mention here a few ongoing projects which forced us to improve the methods. The first one is about computations of knot invariants. For cubical representation of knots we provide an alorithm which computes fundamental group presentation of the knot complement.In the next project we provide an implementation for computing persistence of maps. It is a new theory which extends the concept of persistence homology to maps. The idea is to get a knowledge about an unknown map from sampled data. In the last project mentioned here we focus on big data sets. Often we are facing problems which cannot be solved on one machine or use only once CPU core. Applications conviced us to start implementations of our algorithms for parallel and distributed computations.

5 Experimental evaluation

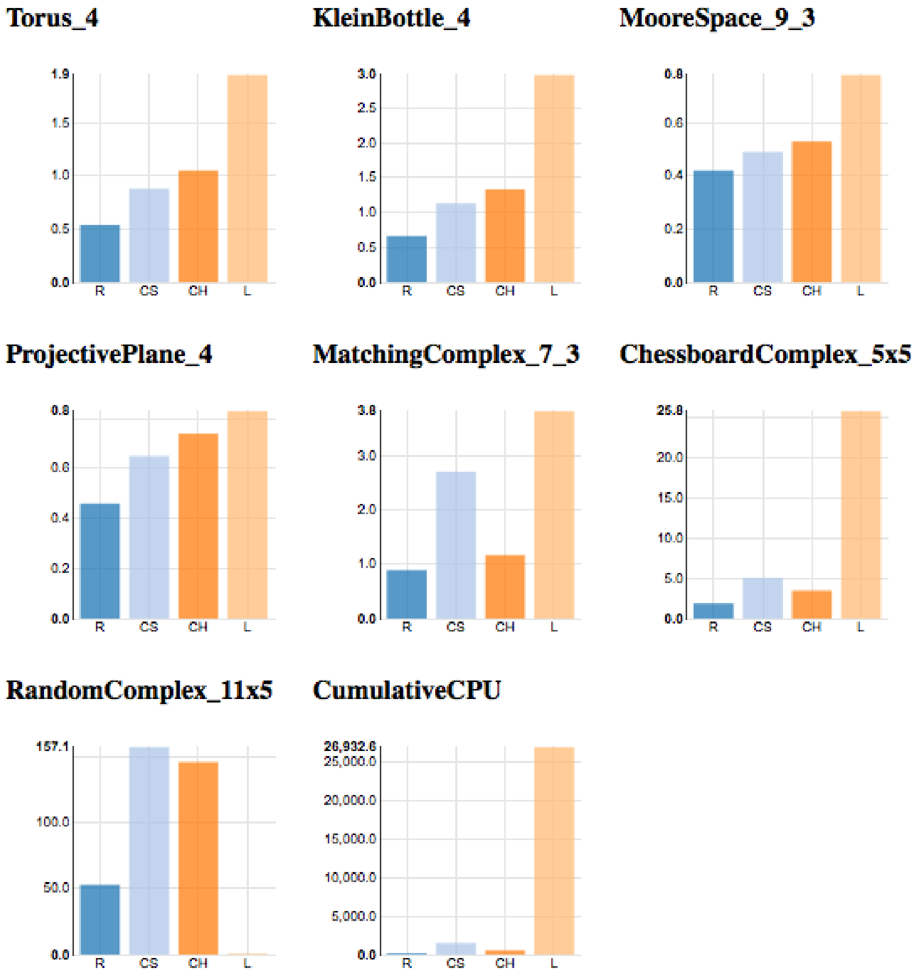
Algorithms implemented in CAPD::RedHom package behaves incredibly well for inflated data sets. We see such sets especially in applications, where a continuous problem is translated into a combinatorial problem. Among many examples, there is a common pattern: to achieve sufficient theoretical conditions 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 [24] (programs *homsimpl* and *chomp-simplicial*) and Linbox [27] (program *homology_gap* 1.4.3 used in GAP [25] - we were not able to compile latest version 1.4.5, because GPC compiler was removed from Ubuntu/Debian in 2011). For the comparison we generated simplcial complexes using Sage [26]. 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 below pictures 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 during the experiment.

Figure 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] P. Dłotko, R. Ghrist, M. Juda, M. Mrozek, Distributed computation of coverage in sensor networks by homological methods, *Applicable Algebra in Engineering, Communication and Computing*, **23**(2012), 29–58.

[2] P. Dłotko, T. Kaczynski, M. Mrozek, Th. Wanner, Coreduction Homology Algorithm for Regular CW-Complexes, *Discrete and Computational Geometry*, **46**(2011), 361–388, DOI: 10.1007/s00454-010-9303-y.

[3] P. Dłotko,R. Specogna, F. Trevisan, Automatic generation of cuts on large-sized meshes for the T-Omega geometric eddy-current formulation; *Computer Methods in Applied Mechanics and Engineering*, **198**(2009), 3765–3781.

[4] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological Persistence and Simplification, *Discrete and Computational Geometry* **28**(2002), 511–533.

[5] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, H. Wagner, M. Juda, P. Dłotko, The Efficiency of a Homology Algorithm based on Discrete Morse Theory and Coreductions, in: Proceedings of the 3rd International Workshop on Computational Topology in Image Context, Chipiona, Spain, November 2010 (Roco Gonzlez Daz Pedro Real Jurado (Eds.)), *Image A* Vol. 1(2010), 41–47 (ISSN: 1885-4508)

[6] M. Juda, M. Mrozek, Z_2-Homology of weak $(p-2)$-faceless p-manifolds may be computed in $O(n)$ time, *Topological Methods in Nonlinear Analysis*, **40**(2012), 137–159.

[7] T. Kaczynski, P. Dłotko, M. Mrozek, Computing the Cubical Cohomology Ring, in: Proceedings of the 3rd International Workshop on Computational Topology in Image Context, Chipiona, Spain, November 2010 (Roco Gonzlez Daz Pedro Real Jurado (Eds.)), *Image A* Vol. 3(2010), 137–142 (ISSN: 1885-4508)

[8] T. Kaczynski, M. Mischaikow, M. Mrozek, *Computational Homology*, Applied Mathematical Sciences 157, Springer-Verlag, 2004.

[9] T. Kaczynski, M. Mrozek, M. Ślusarek, Homology computation by reduction of chain complexes, *Computers and Math. Appl.* **35**(1998), 59–70.

[10] T. Kapela, M. Mrozek, P. Pilarczyk, D. Wilczak, P. Zgliczyński, CAPD - a Rigorous Toolbox for Computer Assisted Proofs in Dynamics, technical report, Jagiellonian University, 2010.

[11] K. Mischaikow, M. Mrozek, Chaos in Lorenz equations: a computer assisted proof, *Bull. AMS (N.S.)*, 33(1995), 66–72.

[12] K. Mischaikow, M. Mrozek, Chaos in the Lorenz equations: a computer assisted proof. Part II: details, *Mathematics of Computation*, **67**(1998), 1023–1046.

[13] K. Mischaikow, M. Mrozek, P. Pilarczyk, Graph approach to the computation of the homology of continuous maps, *Found. Comp. Mathematics* **5**(2005), 199–229.

[14] M. Mrozek, Index Pairs Algorithms, *Found. Comp. Mathematics*, **6**(2006), 457-493.

[15] M. Mrozek, Čech Type Approach to Computing Homology of Maps, *Discrete and Computational Geometry*, **44.3**(2010), 546–576, DOI: 10.1007/s00454-010-9255-2.

[16] M. Mrozek, P. Pilarczyk, N. Żelazna, Homology algorithm based on acyclic subspace, *Computers and Mathematics with Applications*, **55**(2008), 2395–2412.

[17] M. Mrozek, B. Batko, Coreduction homology algorithm, *Discrete and Computational Geometry*, **41**(2009), 96–118.

[18] M. Mrozek, Th. Wanner, Coreduction homology algorithm for inclusions and persistent homology, *Computers and Mathematics with Applications*, **60.10**(2010), 2812-2833. DOI: 10.1016/j.camwa.2010.09.036.

[19] M. Mrozek, M. Żelawski, A. Gryglewski, S. Han, A. Krajniak, Homological methods for extraction and analysis of linear features in multidimensional images, *Pattern Recognition*, **45**(2012), 285–298.

[20] P. Pilarczyk, Computer assisted method for proving existence of periodic orbits, *Topological Methods in Nonlinear Analysis*, **13**(1999), 365–377.

[21] J. Koonin, Topology of eigenspace posets for imprimitive reflection groups, *http://arxiv.org/abs/1208.4435*

[22] Computer Assisted Proofs in Dynamics, `http://capd.ii.uj.edu.pl/`

[23] Reduction Homology Algorithms, `http://redhom.ii.uj.edu.pl/`

[24] Computational Homology Project, `http://chomp.rutgers.edu/`

[25] GAP System, `http://www.gap-system.org/`

[26] Sage, `http://www.sagemath.org/`

[27] Linbox, `http://www.eecis.udel.edu/ dumas/Homology/`