Simplicial blowups and discrete normal surfaces in the GAP package simpcomp

Felix Effenberger and Jonathan Spreer
Institut für Geometrie und Topologie
Universität Stuttgart, Germany
effenberger@mathematik.uni-stuttgart.de
spreer@mathematik.uni-stuttgart.de

May 27, 2011

Abstract

simpcomp is an extension to GAP, the well known system for computational discrete algebra. It allows the user to work with simplicial complexes. In the latest version, support for simplicial blowups and discrete normal surfaces was added, both features unique to simpcomp. Furthermore, new functions for constructing certain infinite series of triangulations have been implemented and interfaces to other software packages have been improved to previous versions.

1 Introduction

simpcomp [10, 9] is a package for working with simplicial complexes. Its aim is to provide the user with a broad spectrum of functionality regarding simplicial constructions and the calculation of properties of simplicial complexes.

Important goals during the development of simpcomp were interactivity, ease of use, completeness of documentation and ease of extensibility. The software allows the user to interactively construct simplicial complexes and to compute their properties in the GAP [12] or SAGE [25] shell. It is sought of as a tool for researchers to verify or disprove a conjecture one might have and to quickly do simplicial constructions using the computer. Furthermore, it makes use of GAP’s expertise in groups and group operations. For example, automorphism groups (cf. [2]) and fundamental groups of simplicial complexes can be computed and examined further within the GAP system.
With its development being started in March 2009, simpcomp still is a rather young project, but now already contains roughly 270 functions and its manual [9] contains about 180 pages of documentation. At the ISSAC 2010 in Munich, simpcomp won the Best Software Presentation Award by the Fachgruppe Computeralgebra. The package is currently under review by the GAP Council and is subject to acceptance as a shared package of GAP that will be included in all future standard GAP installations.

Furthermore, the software system SAGE can be used to work with simpcomp. See [9] for an interactive web demo of simpcomp based on SAGE notebooks. Connecting simpcomp more tightly with SAGE is planned for future releases.

The upcoming version 1.5 scheduled for May 2010 will be fully compliant to the standard GAP object mechanism and will have more advanced interfaces to other software packages.

2 Why simpcomp

simpcomp encapsulates all methods and properties of a simplicial complex in a new GAP object type (as an abstract data type). This way simpcomp can transparently cache properties already calculated, thus preventing unnecessary double calculations. This is mainly done by using the GAP native caching mechanism [6]. It also takes care of the error-prone vertex labeling of a complex.

simpcomp is written entirely in the GAP scripting language. This has two implications:

(1) On the one hand, this limits the efficiency of the implementation as the scripted GAP code can never be as fast as native code.

(2) On the other hand, this gives the user the possibility to see behind the scenes and to customize or alter simpcomp functions in an interactive way, profiting of all the mathematical and algebraic capabilities of the GAP scripting language.

In the author’s view, the advantages of (2) outweigh the drawbacks of (1). This was a major point when deciding on simpcomp’s design principles and also sets the software in contrast to other software packages like polymake [14] that are heterogeneous, i.e. in which algorithms are implemented in various languages.
simpcomp's fundamental functions can be roughly divided into four groups: (i) functions constructing simplicial complexes, (ii) functions calculating properties of simplicial complexes, (iii) functions dealing with bistellar moves and (iv) functions concerning the library and the communication with other software packages – for a full list of supported features see the documentation [9] or use GAP’s built in interactive help system (all of simpcomp’s functions start with the prefix SC).

Concerning (i), complexes can be constructed by supplying a facet list or a set of generators together with a prescribed automorphism group – the latter form being the common in which a complex is presented in a publication. This feature is to our knowledge unique to simpcomp. Furthermore, standard triangulations can be generated from scratch (simplex, cross polytope, cyclic polytope, stacked polytopes, etc.) and simplicial Cartesian products, connected sums, handle additions, etc. can be formed, enabling the user to obtain a wide variety of complexes with different properties. In (ii), basic properties and invariants of a simplicial complex like its dimension, the f-, h- or g-vector, Euler characteristic, (co-)homology groups, intersection form, Betti numbers, fundamental group, orientation, etc. can be computed. Concerning (iii): bistellar moves [23] allow to modify a given triangulation while leaving its PL homeomorphism type invariant (for an introduction to PL topology see [24]). The concept has proven a powerful tool in combinatorial topology and can for example be used to reduce the vertex number of a given triangulation, to check if a simplicial complex is a manifold, to establish PL homeomorphisms between pairs of manifolds, to randomize complexes, to check whether a complex lies in a certain class of triangulations [8], and so on. Concerning (iv), there exist functions to save and load simplicial complexes to and from files (in an XML format) and to import and export complexes in various formats (e.g. from and to polymake/TOPAZ [14], Macaulay2 [15], \LaTeX, etc.). In addition, the internal library, currently containing more than 7,000 triangulations, can be searched either using the name of a complex or a condition on the properties that it has to fulfill. The software also supports user libraries which can be used to organize own collections of triangulations produced with simpcomp.
4 New features in version 1.4

- Support for simplicial blowups: in algebraic geometry, blowups provide a useful way to study singularities of algebraic varieties [13]. The idea is to replace a point by all lines passing through that point. This concept is now available for combinatorial 4-manifolds (cf. [27]) and integrated into simpcomp. This functionality is to the authors’ knowledge not provided by any other software package so far.

- Support for discrete normal surfaces [17, 16, 28] and slicings: slicings of combinatorial $d$-manifolds are (non-singular) $(d - 1)$-dimensional level sets of polyhedral Morse functions. In dimension 3, slicings are discrete normal surfaces. simpcomp supports discrete normal surfaces as a new object type and enables the user to generate and analyze slicings together with the corresponding Morse functions.

- New infinite series of highly symmetric triangulations: many highly symmetric triangulations occur as members of infinite series. Some of these series are well known and have been integrated into simpcomp already (simplex, cross polytope, cyclic polytope, etc.). Others were just recently found by the second author. simpcomp contains the first computer implementation of these series presented in [29, Chapter 4].

- homalg interface: simpcomp now can use the GAP package homalg [4] for its homology computations. This allows the computation of (co-)homology groups of simplicial complexes over arbitrary rings and fields, as well as the usage of all the functionality related to homological algebra that homalg provides.

5 Roadmap: version 1.5 and beyond

The current version of simpcomp is 1.4. On the roadmap for the upcoming versions are the following points.

- Faster bistellar moves: currently, the algorithms to perform bistellar moves are implemented in the GAP scripting language. Since the performance of some of the algorithms implemented in simpcomp are mainly dependent on the running time of bistellar moves we plan to implement these functions in C. This should vastly speed up all calculations using bistellar moves. However, sticking to our design principles, the
higher-level steering code related to bistellar moves will remain on the GAP side.

- We want to investigate how simpcomp can more closely interact with other software packages in the field (both GAP and non-GAP, e.g. Macaulay2 and SAGE).

- There exists a combinatorial formula for calculating the Stiefel-Whitney class of combinatorial manifolds [3] due to Banchoff. We plan on including this, and possibly further invariants, into simpcomp.

- As a long term goal, we would like to provide the functionality to perform surgery on combinatorial 3- and 4-manifolds. This would be a step forward to constructing candidates for combinatorial manifolds with exotic PL structures as already done in the smooth setting by Akbulut [1].

6 Examples

This section contains a small demonstration of the capabilities of simpcomp in form of transcripts of the GAP shell for some example constructions. Most of the features presented below have been newly introduced in version 1.4.

6.1 Normal surfaces in cyclic 4-polytopes

For $n \geq 3$, consider the cyclic 4-polytope $C_4(2n)$ on $2n$ vertices with vertex labels 1 to $n$. By Gale’s evenness condition, neither the span of all odd nor the span of all even vertices in $C_4(2n)$ contains a triangle of $C_4(2n)$. Thus, given the combinatorial 3-sphere $S = \partial C_4(2n)$, a level set of a Morse function on $S$ separating the even from the odd vertices gives rise to a handle body decomposition of $S$ — this is a discrete normal surface in the sense of [28].

This construction can be done in simpcomp as follows. Note that we arbitrarily chose $n = 5$ for demonstration purposes below.

\begin{verbatim}
gap> LoadPackage("simpcomp");; #load the package
Loading simpcomp 1.4.0 by F.Effenberger and J.Spreer
\protect\vrule width0pt\protect\href{http://www.igt.uni-stuttgart.de/LstDiffgeo/simpcomp}{http://www.igt.uni-stuttgart.de/LstDiffgeo/simpcomp}
gap> c_4_10:=SCBdCyclicPolytope(4,10);
[SimplicialComplex
 Properties known: Chi, Dim, ... , TopologicalType, VertexLabels.
\end{verbatim}
Above, we constructed the boundary of the cyclic 4-polytope $\partial C_4(10)$ on 10 vertices. Note the properties of $c_{4\cdot10}$ already computed by simpcomp. We now look at the level set of a Morse function on $c_{4\cdot10}$ which separates even and odd vertices:

```gap
gap> sl:=SCSlicing(c,[[1,3,5,7,9],[2,4,6,8,10]]);

Properties known: Chi, ConnectedComponents, ..., VertexLabels, Vertices.

Name="slicing [ [ 1, 3, 5, 7, 9 ], [ 2, 4, 6, 8, 10 ] ] of Bd(C_4(10))"
Dim=2
Chi=-10
F= [ 25, 70, 0, 35 ]
IsConnected=true
TopologicalType="#8(T^2)"

/NormalSurface
```

The resulting polytopal complex on 25 vertices is a discrete normal surface without triangles and with 35 quadrilaterals. Topologically, it is the orientable surface with Euler characteristic $-10$, and thus homeomorphic to $(T^2)^6$. A triangulated version of this complex can be easily obtained as follows.

```gap
gap> trig:=SCNSTriangulation(sl);
```

6.2 Combinatorial blowups of the Kummer variety $K^4$

The 4-dimensional abstract Kummer variety $K^4$ with 16 nodes leads to the $K3$ surface by resolving the 16 singularities [26]. Using simpcomp, this process could be carried out in a combinatorial setting for the first time, cf. [27]. The first step of this so-called dilatation or blowup process can be done as follows.

We first load the singular 16-vertex triangulation of $K^4$ due to Kühnel [18] from the library.

```gap
gap> SCLib.SearchByName("Kummer");
[ [ 7493, "4-dimensional Kummer variety (VT)" ] ]
gap> k4:=SCLib.Load(7493);
```
We now verify that the link of vertex 1 in $K^4$ topologically is a real projective 3-space. The ranks of its integral homology groups and its fundamental group are the following:

```
gap> lk1:=k4.Link(1);;
gap> lk1.Homology;
[ [ 0, [ ] ], [ 0, [ 2 ] ], [ 0, [ ] ], [ 1, [ ] ] ]
```

We now verify that, as suspected, the complex is PL homeomorphic to the minimal 11-vertex triangulation of $\mathbb{RP}^3$ from the library. This is done using a heuristic algorithm based on bistellar moves.

```
gap> SCLib.SearchByName("RP^3");
[ [ 45, "RP^3" ], [ 113, "RP^3=L(2,1) (VT)" ], ... ]
gap> minRP3:=SCLib.Load(45);;
gap> SCEquivalent(lk1,minRP3);
#I SCReduceComplexEx: complexes are bistellarly equivalent.
true
```

Finally, we resolve the singularity of $K^4$ at vertex 1 by a simplicial blowup.

```
#I SCBlowup: checking if singularity is a combinatorial manifold...
#I SCBlowup: ...true
#I SCBlowup: checking type of singularity...
```
Indeed, the second Betti number increased by 1, again as expected.

\[
gap>k4.Homology;  
[[0, []], [0, []], [6, [2, 2, 2, 2]], [0, []], [1, []]]  
gap;c.Homology;  
[[0, []], [0, []], [7, [2, 2, 2]], [0, []], [1, []]]  
\]

The resulting complex now only has 15 singularities. By iterating this process 15 more times, we obtain a combinatorial triangulation of the \(K3\) surface with standard PL structure.

7 Acknowledgments

The authors acknowledge support by the DFG: simpcomp has partly been developed within the DFG projects Ku 1203/5-2 and Ku 1203/5-3.

References


