simpcomp – A GAP toolbox for simplicial complexes

Felix Effenberger and Jonathan Spreer

April 9, 2010

1 The GAP package simpcomp

simpcomp [ES09] is an extension (a so called package) to GAP [GAP07], the well known system for computational discrete algebra. The package enables the user to compute numerous properties of (abstract) simplicial complexes, provides functions to construct new complexes from existing ones and an extensive library of triangulations of manifolds. For an introduction to the field of PL topology see the books [RS72] and [Küh95].

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.

2 What is new

simpcomp is a package for working with simplicial complexes in the GAP system. In contrast to the package homology [DHSW04] which focuses on simplicial homology computation, simpcomp claims to provide the user with a broader spectrum of functionality regarding simplicial constructions.

simpcomp allows the user to interactively construct complexes and to compute their properties in the GAP shell. Furthermore, it makes use of GAP’s expertise in groups and group operations. For example, automorphism groups and fundamental groups of complexes can be computed and examined further within the GAP system. Apart from supplying a facet list, the user can as well construct simplicial complexes from a set of generators and 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, simpcomp has an extensive library of known triangulations of manifolds. This is the first time that they are easily accessible without having to look them up in the literature [KL99], [CK01], or online [Lut]. This allows the user to work with many different known triangulations without having to construct them first.

3 simpcomp benefits

simpcomp is written entirely in the GAP scripting language, thus giving the user the possibility to see behind the scenes and to customize or alter simpcomp functions if needed.

The main benefit when working with simpcomp over implementing the needed functions from scratch is that simpcomp encapsulates all methods and properties of a simplicial complex in a new GAP object type (as an abstract data type). This way, among other things, simpcomp can
transparently cache properties already calculated, thus preventing unnecessary double calculations. It also takes care of the error-prone vertex labeling of a complex.

**simpcomp** provides the user with functions to save and load the simplicial complexes to and from files and to import and export a complex in various formats (e.g. from and to polymake/TOPAZ [GJ00], Macaulay2 [GS], \LaTeX, etc.).

In contrast to the software package polymake [GJ00] providing the most efficient algorithms for each task in form of a heterogeneous package (where algorithms are implemented in various languages), the primary goal when developing **simpcomp** was not efficiency (this is already limited by the GAP scripting language), but rather ease of use and ease of extensibility by the user in the GAP language with all its mathematical and algebraic capabilities.

The package includes an extensive manual (see [ES09]) in which all functionality of **simpcomp** is documented and also makes use of GAP’s built in help system such that all the documentation is available directly from the GAP prompt in an interactive way.

### 4 Some operations and constructions that **simpcomp** supports

**simpcomp** implements many standard and often needed functions for working with simplicial complexes. These functions can be roughly divided into three groups: (i) functions generating simplicial complexes (ii) functions to construct new complexes from old and (iii) functions calculating properties of complexes – for a full list of supported features see the documentation [ES09].

**simpcomp** furthermore implements a variety of functions connected to bistellar moves (also known as Pachner moves [Pac87]) on simplicial complexes. For example, **simpcomp** can be used to construct randomized spheres or randomize a given complex. Another prominent application of bistellar moves implemented in **simpcomp** is a heuristical algorithm that determines whether a simplicial complex is a combinatorial manifold (i.e. that each link is PL homeomorphic to the boundary of the simplex). This algorithm was first presented by F.H. Lutz and A. Björner [BL00]. It uses a simulated annealing type strategy in order to minimize vertex numbers of triangulations while leaving the PL homeomorphism type invariant.

The package also supports slicings of 3-manifolds (known as normal surfaces, see [Kne29], [Hak61], [Spr10]) and related constructions.

The current version of **simpcomp** is 1.1.43 (April 7th, 2010). On the roadmap for the next version 1.2.x which should appear still in 2010 are the following features:

- Support for simplicial blowups, i.e. the resolutions of ordinary double points in combinatorial 4-pseudomanifolds. This functionality is to the authors’ knowledge not provided by any other software package so far.

- Construction of simplicial complexes of prescribed dimension, vertex number and transitive automorphism group as described in [Lut99], [CK01].

- Extension of the complex library by pseudo manifolds with vertex transitive automorphism groups.

- Closer interaction with the software system Macaulay2.
5 An example

This section contains a small demonstration of the capabilities of simpcomp in form of an example construction.

M. Casella and W. Kühnel constructed a triangulated K3 surface with minimum number of 16 vertices in [CK01]. They presented it in terms of the complex obtained by the automorphism group $G \cong AGL(1,F_{16})$ given by the five generators

$$G = \left\{ (12)(34)(56)(78)(910)(1112)(1314)(1516), \\
(13)(24)(57)(68)(911)(1012)(1315)(1416), \\
(15)(26)(37)(48)(913)(1014)(1115)(1216), \\
(2131511143581674910612) \right\} .$$

acting on the two generating simplices $\Delta_1 = \langle 2, 3, 4, 5, 9 \rangle$ and $\Delta_2 = \langle 2, 5, 7, 10, 11 \rangle$. It turned out to be a non-trivial problem to show (i) that the complex obtained is a combinatorial 4-manifold and (ii) to show that it is homeomorphic to a K3 surface as topological 4-manifold.

This turns out to be a rather easy task using simpcomp, as will be shown below. We will fire up GAP, load simpcomp and then construct the complex from its representation given above:

```
gap

IsomorphismFamily( G, \langle 2, 3, 4, 5, 9 \rangle );
IsomorphismFamily( G, \langle 2, 5, 7, 10, 11 \rangle );
\end{verbatim}
```

Information at: \protect\url{http://www.gap-system.org}

Try '?help' for help. See also '?copyright' and '?authors'.

Loading the library. Please be patient, this may take a while.

GAP4, Version: 4.4.6 of 02-Sep-2005, i686-pc-linux-gnu-gcc
Components: small 2.1, small2 2.0, small3 2.0, small4 1.0, small5 1.0,
small6 1.0, small7 1.0, small8 1.0, small9 1.0, small10 0.2,
id3 2.0, id3 2.1, id4 1.0, id5 1.0, id6 1.0, id9 1.0, id10 0.1,
trans 1.0, prim 2.1 loaded.

Packages: AClib 1.1, Polycyclic 1.1, Alnuth 2.1.3, CrystCat 1.1.2,
Cryst 4.1.4, AutPGrp 1.2, CRISP 1.2.1, CTblLib 1.1.3,
TomLib 1.1.2, FactInt 1.4.10, FGA 1.1.0.1, GAPDoc 0.9999,
Homology 1.4.2, IRREDsol 1.0.9, LAGUNA 3.3.1, Sophus 1.21,
Polenta 1.2.1, ResClasses 2.1.1 loaded.

```
gap> LoadPackage("simpcomp");
#load the package
```

Loading simpcomp 1.1.21
by F.Effenberger and J.Spreer

```
gap> SCInfoLevel(0);
#suppress simpcomp info messages
```

```
gap> G:=Group((1,2)(3,4)(5,6)(7,8)(9,10)(11,12)(13,14)(15,16),
> (1,3)(2,4)(5,7)(6,8)(9,11)(10,12)(13,15)(14,16),
```

3
We first compute the $f$-vector, the Euler characteristic and the homology groups of $K_3$:

\begin{verbatim}
gap> K3.F; [ 16, 120, 560, 720, 288 ]
gap> K3.Chi; 24
\end{verbatim}

Now we verify that the complex $K_3$ is a combinatorial manifold using the heuristic algorithm based on bistellar moves described above:

\begin{verbatim}
gap> K3.IsManifold; true
\end{verbatim}

In a next step we compute the parity and the signature of the intersection form of the complex $K_3$:

\begin{verbatim}
gap> K3.IntersectionFormParity; 0
gap> K3.IntersectionFormSignature; [ 22, 3, 19 ]
\end{verbatim}

This means that the intersection form of the complex $K_3$ is even. It has dimension 22 and signature $19 - 3 = 16$. Furthermore, $K_3$ is simply connected as can either be verified by showing that the fundamental group is trivial or by checking that the complex is 3-neighborly:

\begin{verbatim}
gap> K3.FundamentalGroup;
<fp group with 105 generators>
gap> Size(last); 1
\end{verbatim}

It now follows from a theorem of M. Freedman [Fre82] that the complex is in fact homeomorphic to a $K_3$ surface because it has the same (even) intersection form.

References


In Polytopes—combinatorics and computation (Oberwolfach, 1997), volume 29 of DMV Sem., 

[GS] Daniel R. Grayson and Michael E. Stillman. Macaulay2, a software system for research in 
algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.


[Küh95] Wolfgang Kühnel. Tight polyhedral submanifolds and tight triangulations, volume 1612 of Lecture 


[Pac87] Udo Pachner. Konstruktionsmethoden und das kombinatorische Homöomorphieproblem für 

[RS72] Colin P. Rourke and Brian J. Sanderson. Introduction to piecewise-linear topology. Springer- 