Computational Topology in simpcomp and GAP

Topology and Computer, Osaka, October 22, 2017

Jonathan Spreer, Freie Universität Berlin

The GAP system

- GAP - Groups, Algorithms, Programming – a computer algebra system, developed at RWTH Aachen, Germany ($198x$, $6 \leq x \leq 9$)
- Open source, obtainable online for Linux, Windows and MacOS from `http://www.gap-system.org`
- Particular emphasis on Computational Group Theory
- Provides a (high-level) programming language, library of thousands of functions implementing algebraic algorithms, as well as large data libraries of groups
- Supports peer-reviewed extensions (so-called GAP packages) written in C or GAP scripting language

The GAP package simpcomp

- Joint work with Felix Effenberger
- Purpose: working with (abstract) simplicial complexes in the context of combinatorial topology
- Goal: easy to use in an interactive way
- Uses the convenient GAP environment including command completion, inline help system and a comprehensive amount of already implemented functionalities
- Comes with every full installation of GAP

What can `simpcomp` do?

- Homology et al,
- bistellar moves,
- discrete Morse theory,
- simplicial blowups,
- automorphism groups,
- 300+ helper functions, etc.

Easy to extend (mostly written in GAP scripting language)

There exists a comprehensive manual available from command line or as a pdf (200+ pages including at least one example for each function)

Saves time

Getting started

Standard

- On ubuntu `sudo apt-get install gap`
- Available within `Sage`
- Windows installer from `www.gap-system.org`

For the latest version source code available from

- `www.gap-system.org`
- `https://github.com/simpcomp-team/simpcomp`

After installation

- Start GAP
- To use simpcomp:
  - `ret := LoadPackage("simpcomp");`
  - `if ret = fail then`

    `talk to me:` **jonathan.spreer@fu-berlin.de**

    `fi;`

Example 1: out of the box

The minimal 16-vertex triangulation of the K3 surface due to Casella and Kühnel[1] is presented in the literature by its automorphism group $G \cong AGL(1, \mathbb{F}_{16})$ with permutation representation

$$G = \left\langle \begin{array}{c} (1,2)(3,4)(5,6)(7,8)(9,10)(11,12)(13,14)(15,16), \\ (1,3)(2,4)(5,7)(6,8)(9,11)(10,12)(13,15)(14,16), \\ (1,5)(2,6)(3,7)(4,8)(9,13)(10,14)(11,15)(12,16), \\ (1,9)(2,10)(3,11)(4,12)(5,13)(6,14)(7,15)(8,16), \\ (2,13,15,11,14,3,5,8,16,7,4,9,10,6,12) \end{array} \right\rangle,$$

acting on two generating simplices

- $\Delta_1 = \langle 2,3,4,5,9 \rangle$ and
- $\Delta_2 = \langle 2,5,7,10,11 \rangle$.

$G \cdot \{\Delta_1, \Delta_2\}$ is a pure 4-dimensional simplicial complex with 288 facets.

Question: Is this really a triangulation of the K3 surface?

[1] Casella, Kühnel: A triangulated $K3$ surface with the minimum number of vertices, *Topology* **40** (2001), 753–772.

Example 1: out of the box

Let us check:

```
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),
> (1,5)(2,6)(3,7)(4,8)(9,13)(10,14)(11,15)(12,16),
> (1,9)(2,10)(3,11)(4,12)(5,13)(6,14)(7,15)(8,16),
> (2,13,15,11,14,3,5,8,16,7,4,9,10,6,12));;

gap> K3:=SCFromGenerators(G,[[2,3,4,5,9],[2,5,7,10,11]]);
[SimplicialComplex

 Properties known: AutomorphismGroup, AutomorphismGroupSize,
 AutomorphismGroupStructure, AutomorphismGroupTransitivity,
 Dim, Facets, Generators, Name, VertexLabels.

 Name="complex from generators under group ((C2 x C2 x C2 x C2) : C5) : C3"
 Dim=4
 AutomorphismGroupSize=240
 AutomorphismGroupStructure="((C2 x C2 x C2 x C2) : C5) : C3"
 AutomorphismGroupTransitivity=2

/SimplicialComplex]
```

Example 1: out of the box

First compute the *f*-vector, the Euler characteristic and the homology groups of K3:

```
gap> K3.F;
[ 16, 120, 560, 720, 288 ]
gap> K3.Chi;
24
gap> K3.Homology;
[ [ 0, [  ] ], [ 0, [  ] ], [ 22, [  ] ], [ 0, [  ] ], [ 1, [  ] ] ]
```

Now verify that K3 is a combinatorial manifold using a heuristic algorithm based on bistellar moves:

```
gap> K3.IsManifold;
true
```

Example 1: out of the box

K3 is simply connected (compute fundamental group or check that the complex is 3-neighborly):

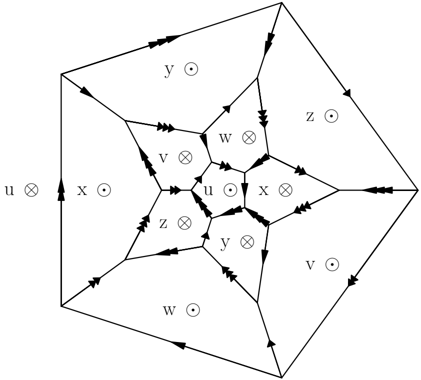
```
gap> K3.FundamentalGroup;
<fp group with 105 generators>
gap> Size(last);
1
gap> K3.Neighborliness;
3
```

Compute the parity and the signature of the intersection form of K3:

```
gap> K3.IntersectionFormParity;
0
gap> K3.IntersectionFormSignature;
[ 22, 3, 19 ]
```

This means that the intersection form of K3 is even, has dimension 22 and signature $19 - 3 = 16$.

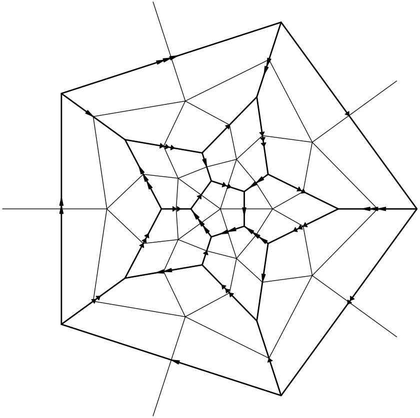
- Closed hyperbolic 3-manifold obtained by gluing opposite faces of the dodecahedron with a $3/5\pi$-twist
- Known to be non-Haken[2]
- Here: looking for interesting Haken covers



[2] Burton, Rubinstein, Tillmann: The Weber-Seifert dodecahedral space is non-Haken, *TAMS* **364** (2012), 911–932.

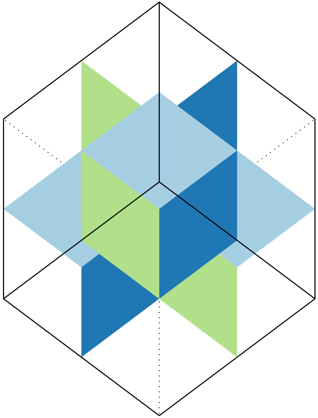
Example 2: The Weber-Seifert dodecahedral space

- Dodecahedron can be decomposed into 20 cubes
- Cubulation non-positively curved



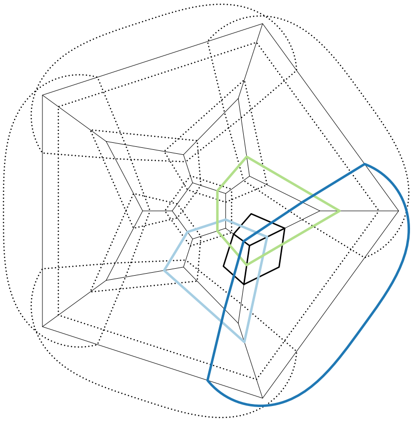
Example 2: The Weber-Seifert dodecahedral space

- Mid-squares define a canonical immersed surface...
- ...in any finite cover of the Weber-Seifert dodecahedral space
- Interested in covers where the canonical immersed surface splits into embedded connected components



Example 2: The Weber-Seifert dodecahedral space

- Mid-squares define a canonical immersed surface...
- ...in any finite cover of the Weber-Seifert dodecahedral space
- Interested in covers where the canonical immersed surface splits into embedded connected components



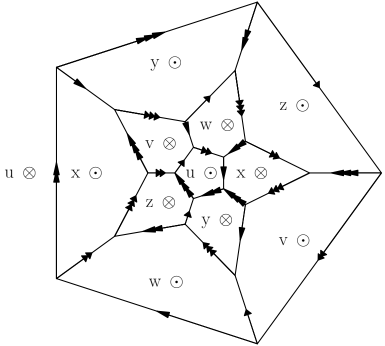
Example 2: The Weber-Seifert dodecahedral space

```
G:=FreeGroup(["u","v","w","x","y","z"]);
gens:=GeneratorsOfGroup(G);

u:=gens[1];
v:=gens[2];
w:=gens[3];
x:=gens[4];
y:=gens[5];
z:=gens[6];

rels:=[
x*u*w*v^-1*y^-1,
y*u*x*w^-1*z^-1,
z*u*y*x^-1*v^-1,
v*u*z*y^-1*w^-1,
y^-1*v^-1*x^-1*z^-1*w^-1,
w*u*v*z^-1*x^-1];

pi1:=G/rels;
<fp group on the generators [ u, v, w, x, y, z ]>
```



Example 2: The Weber-Seifert dodecahedral space


```
G:=FreeGroup(["u","v","w","x","y","z"]);
gens:=GeneratorsOfGroup(G);

u:=gens[1];
v:=gens[2];
w:=gens[3];
x:=gens[4];
y:=gens[5];
z:=gens[6];

rels:=[
x*u*w*v^-1*y^-1,
y*u*x*w^-1*z^-1,
z*u*y*x^-1*v^-1,
v*u*z*y^-1*w^-1,
y^-1*v^-1*x^-1*z^-1*w^-1,
w*u*v*z^-1*x^-1];

pi1:=G/rels;
<fp group on the generators [ u, v, w, x, y, z ]>
```

- Structure of fundamental group (and its low index subgroups) respects cell structure of Weber-seifert dodecahedral space (and its low degree covers)

Example 2: The Weber-Seifert dodecahedral space

```
list:=LowIndexSubgroupsFpGroup(pi1,6);;
for g in list do
  d:=Index(pi1,g);
  orbits:=GeneratorsOfGroup(Range(FactorCosetAction(pi1,g)));
  # 'orbits' is a set of six permutations
  # encoding how the 'd' copies of the
  # dodecahedron (and its cubulation)
  # are glued together
  ...
  # Build up cell structure of cubulation
  ...
  # Track components of canonical immersed surface
  ... etc ...
od;
```

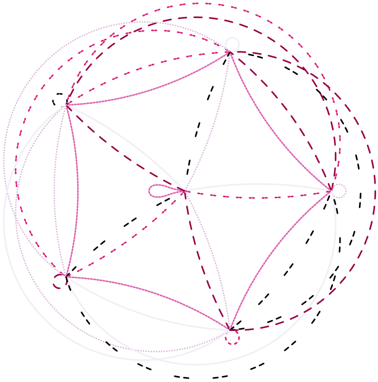
GAP-script available from

https://arxiv.org/src/1702.08080v2/anc

- Makes heavy us of GAP's ability to handle finitely presented groups and permutation groups

Example 2: The Weber-Seifert dodecahedral space

face	orbit
u	$(2,5,3,6,4)$
v	$(1,2,6,4,3)$
w	$(1,3,2,5,4)$
x	$(1,4,3,6,5)$
y	$(1,5,4,2,6)$
z	$(1,6,5,3,2)$



- Examined close to one million covers
- Found 6-fold cover where canonical immersed surface splits into six embedded components
- Found special cover of degree 60 (via normal core of smaller index subgroup)[3]

[3] Spreer, Tillmann: Unravelling the Dodec. Spaces, *2016 Matrix annals*, Springer.

Example 3: Configuration spaces

Ongoing joint work with Haruko Nishi

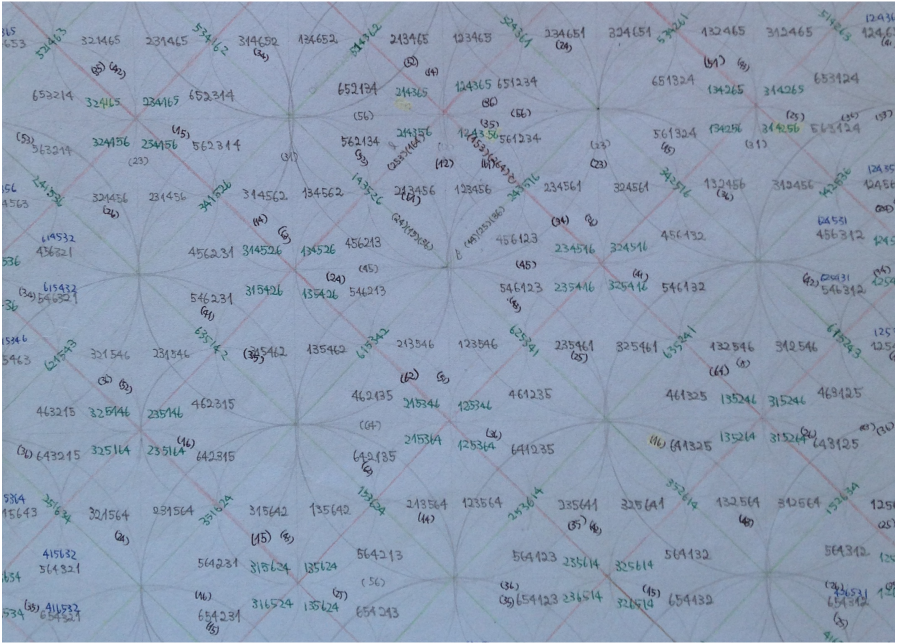
- Configuration space of n marked (weighted) points on the circle ($\mathbb{R}P^1$)

- After factoring out projective transformations: $(n-3)$-dim. space ($\mathrm{PGL}(2,\mathbb{R})$ or $\mathrm{PSL}(2,\mathbb{R})$)

- Each choice of weights defines a Polyhedral structure:
  - facets correspond to generic configurations with a given cyclic ordering (up to / with orientation)
  - lower dimensional faces occur where points collide
  - cusps occur where collisions are “illegal”

Goal: use `GAP` and `simpcomp` to analyse configuration spaces

Example 3: Configuration spaces



Example 3: Configuration spaces

Program takes as input weights $(a_1, a_2, \ldots, a_n)$, $\sum a_i = 2$, and computes

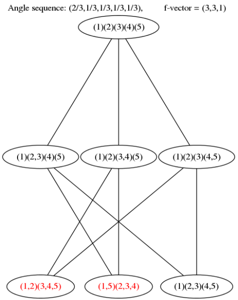
- f-vector, χ, fundamental group
- list combinatorial types of facets, draw their Hasse diagrams
- compute entire complex (up to / with orientation)
- triangulate (compactified complex)

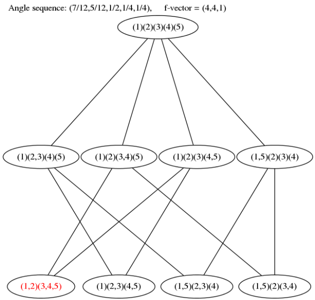
The triangulation can then be passed to

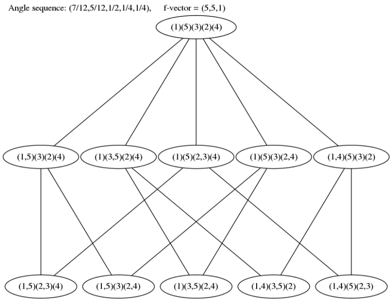
- `simpcomp` for further topological analysis / simplification, etc.
- `Regina`[4] for volume computations (in dimension 3 $\Rightarrow$ $n = 6$), analysis of cusped version, etc.

[4] Burton, Budney, Pettersson et al.: Regina: normal surface and 3-manifold topology software, `https://regina-normal.github.io/` (1999–2017)

Example 3: Configuration spaces

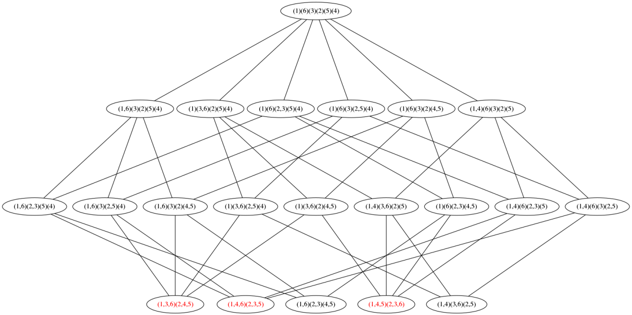




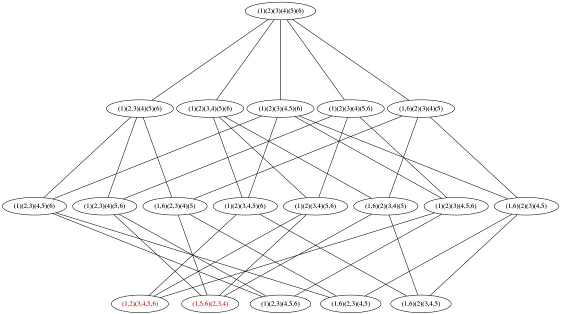


Example 3: Configuration spaces

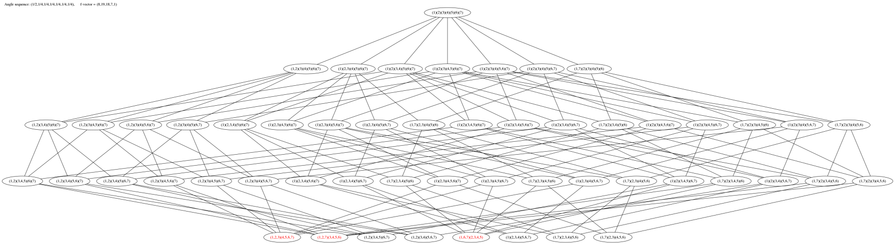
Angle sequence: (1/2,1/2,1/4,1/4,1/4,1/4), f-vector = (5,9,6,1)



Angle sequence: (1/2,1/2,1/4,1/4,1/4,1/4), f-vector = (5,8,5,1)



Example 3: Configuration spaces



How to cite

Effenberger, Spreer, simpcomp - *A GAP package for simplicial complexes*, Version 2.1.7, 2009–2017, https://github.com/simpcomp-team/simpcomp.

GAP – *Groups, Algorithms, and Programming*, Version 4.8.8, Aug 2017, http://www.gap-system.org