A HAP tutorial

(See also an older tutorial or mini-course notes or related book) The HAP home page is here

Graham Ellis

Contents

1 Simplicial complexes & CW complexes **7**

1.1 The Klein bottle as a simplicial complex . . . 7
1.2 Other simplicial surfaces . . . 8
1.3 The Quillen complex . . . 8
1.4 The Quillen complex as a reduced CW-complex . . . 9
1.5 Simple homotopy equivalences . . . 9
1.6 Cellular simplifications preserving homeomorphism type . . . 10
1.7 Constructing a CW-structure on a knot complement . . . 10
1.8 Constructing a regular CW-complex by attaching cells . . . 11
1.9 Constructing a regular CW-complex from its face lattice . . . 12
1.10 Cup products . . . 13
1.11 Intersection forms of 4-manifolds . . . 18
1.12 Cohomology Rings . . . 19
1.13 Bockstein homomorphism . . . 20
1.14 Diagonal maps on associahedra and other polytopes . . . 21
1.15 CW maps and induced homomorphisms . . . 21
1.16 Constructing a simplicial complex from a regular CW-complex . . . 22
1.17 Some limitations to representing spaces as regular CW complexes . . . 23
1.18 Equivariant CW complexes . . . 24
1.19 Orbifolds and classifying spaces . . . 26

2 Cubical complexes & permutahedral complexes **31**

2.1 Cubical complexes . . . 31
2.2 Permutahedral complexes . . . 32
2.3 Constructing pure cubical and permutahedral complexes . . . 34
2.4 Computations in dynamical systems . . . 35

3 Covering spaces **36**

3.1 Cellular chains on the universal cover . . . 36
3.2 Spun knots and the Satoh tube map . . . 37
3.3 Cohomology with local coefficients . . . 39
3.4 Distinguishing between two non-homeomorphic homotopy equivalent spaces . . . 40
3.5 Second homotopy groups of spaces with finite fundamental group . . . 40
3.6 Third homotopy groups of simply connected spaces . . . 41
3.7 Computing the second homotopy group of a space with infinite fundamental group . . . 43

4 Three Manifolds **45**
4.1 Dehn Surgery . . . 45
4.2 Connected Sums . . . 46
4.3 Dijkgraaf-Witten Invariant . . . 46
4.4 Cohomology rings . . . 47
4.5 Linking Form . . . 48
4.6 Determining the homeomorphism type of a lens space . . . 49
4.7 Surgeries on distinct knots can yield homeomorphic manifolds . . . 51
4.8 Finite fundamental groups of 3-manifolds . . . 52
4.9 Poincare's cube manifolds . . . 53
4.10 There are at least 25 distinct cube manifolds . . . 54
4.11 There are at most 41 distinct cube manifolds . . . 60
4.12 There are precisely 18 orientable cube manifolds, of which 9 are spherical and 5 are euclidean . . . 62
4.13 Cube manifolds with boundary . . . 64
4.14 Octahedral manifolds . . . 65
4.15 Dodecahedral manifolds . . . 65
4.16 Prism manifolds . . . 66
4.17 Bipyramid manifolds . . . 67
4.18 Flat manifolds, orbifolds and crystallographic groups . . . 67

5 Topological data analysis **69**
5.1 Persistent homology . . . 69
5.2 Mapper clustering . . . 70
5.3 Some tools for handling pure complexes . . . 71
5.4 Digital image analysis and persistent homology . . . 72
5.5 A second example of digital image segmentation . . . 73
5.6 A third example of digital image segmentation . . . 74
5.7 Naive example of digital image contour extraction . . . 75
5.8 Alternative approaches to computing persistent homology . . . 76
5.9 Knotted proteins . . . 78
5.10 Random simplicial complexes . . . 79
5.11 Computing homology of a clique complex (Vietoris-Rips complex) . . . 81

6 Group theoretic computations **83**
6.1 Third homotopy group of a supsension of an Eilenberg-MacLane space . . . 83
6.2 Representations of knot quandles . . . 83
6.3 Identifying knots . . . 84
6.4 Aspherical 2-complexes . . . 84
6.5 Group presentations and homotopical syzygies . . . 84
6.6 Bogomolov multiplier . . . 86
6.7 Second group cohomology and group extensions . . . 86
6.8 Cocyclic groups: a convenient way of representing certain groups . . . 89
6.9 Effective group presentations . . . 90
6.10 Second group cohomology and cocyclic Hadamard matrices . . . 92
6.11 Third group cohomology and homotopy 2-types . . . 92

7 Cohomology of groups (and Lie Algebras) 95
7.1 Finite groups . . . 95
7.2 Nilpotent groups . . . 99
7.3 Crystallographic and Almost Crystallographic groups . . . 99
7.4 Arithmetic groups . . . 99
7.5 Artin groups . . . 105
7.6 Graphs of groups . . . 105
7.7 Lie algebra homology and free nilpotent groups . . . 107
7.8 Cohomology with coefficients in a module . . . 108
7.9 Cohomology as a functor of the first variable . . . 111
7.10 Cohomology as a functor of the second variable and the long exact coefficient sequence 111
7.11 Transfer Homomorphism . . . 113
7.12 Cohomology rings of finite fundamental groups of 3-manifolds . . . 113
7.13 Explicit cocycles . . . 115
7.14 Quillen's complex and the p-part of homology . . . 118
7.15 Homology of a Lie algebra . . . 122
7.16 Covers of Lie algebras . . . 122

8 Cohomology rings and Steenrod operations for groups 124
8.1 Mod-p cohomology rings of finite groups . . . 124
8.2 Poincare Series for Mod-p cohomology . . . 126
8.3 Functorial ring homomorphisms in Mod-p cohomology . . . 127
8.4 Steenrod operations for finite 2-groups . . . 130
8.5 Steenrod operations on the classifying space of a finite p-group . . . 131
8.6 Mod-p cohomology rings of crystallographic groups . . . 131

9 Bredon homology 135
9.1 Davis complex . . . 135
9.2 Arithmetic groups . . . 135
9.3 Crystallographic groups . . . 136

10 Chain Complexes 137
10.1 Chain complex of a simplicial complex and simplicial pair . . . 137
10.2 Chain complex of a cubical complex and cubical pair . . . 138
10.3 Chain complex of a regular CW-complex . . . 139
10.4 Chain Maps of simplicial and regular CW maps . . . 140
10.5 Constructions for chain complexes . . . 140
10.6 Filtered chain complexes . . . 141
10.7 Sparse chain complexes . . . 142

11 Resolutions 144
11.1 Resolutions for small finite groups . . . 144
11.2 Resolutions for very small finite groups . . . 144
11.3 Resolutions for finite groups acting on orbit polytopes . . . 146
11.4 Minimal resolutions for finite p-groups over $\mathbb{F}_p$ . . . 147
11.5 Resolutions for abelian groups . . . 147
11.6 Resolutions for nilpotent groups . . . 148

11.7 Resolutions for groups with subnormal series . . . 149
11.8 Resolutions for groups with normal series . . . 149
11.9 Resolutions for polycyclic (almost) crystallographic groups . . . 149
11.10 Resolutions for Bieberbach groups . . . 150
11.11 Resolutions for arbitrary crystallographic groups . . . 151
11.12 Resolutions for crystallographic groups admitting cubical fundamental domain . . . 151
11.13 Resolutions for Coxeter groups . . . 152
11.14 Resolutions for Artin groups . . . 152
11.15 Resolutions for $G = SL_2(\mathbb{Z}[1/m])$ . . . 153
11.16 Resolutions for selected groups $G = SL_2(\mathcal{O}(\mathbb{Q}(\sqrt{d}))$ . . . 153
11.17 Resolutions for selected groups $G = PSL_2(\mathcal{O}(\mathbb{Q}(\sqrt{d}))$ . . . 153
11.18 Resolutions for a few higher-dimensional arithmetic groups . . . 154
11.19 Resolutions for finite-index subgroups . . . 154
11.20 Simplifying resolutions . . . 155
11.21 Resolutions for graphs of groups and for groups with aspherical presentations . . . 155
11.22 Resolutions for $\mathbb{F}G$-modules . . . 156

12 Simplicial groups **157**
12.1 Crossed modules . . . 157
12.2 Eilenberg-MacLane spaces as simplicial groups (not recommended) . . . 158
12.3 Eilenberg-MacLane spaces as simplicial free abelian groups (recommended) . . . 158
12.4 Elementary theoretical information on $H^*(K(\pi,n),\mathbb{Z})$ . . . 160
12.5 The first three non-trivial homotopy groups of spheres . . . 161
12.6 The first two non-trivial homotopy groups of the suspension and double suspension of a $K(G,1)$ . . . 162
12.7 Postnikov towers and $\pi_5(S^3)$ . . . 162
12.8 Towards $\pi_4(\Sigma K(G,1))$ . . . 164
12.9 Enumerating homotopy 2-types . . . 165
12.10 Identifying cat^1-groups of low order . . . 166
12.11 Identifying crossed modules of low order . . . 167

13 Congruence Subgroups, Cuspidal Cohomology and Hecke Operators **169**
13.1 Eichler-Shimura isomorphism . . . 169
13.2 Generators for $SL_2(\mathbb{Z})$ and the cubic tree . . . 170
13.3 One-dimensional fundamental domains and generators for congruence subgroups . . . 171
13.4 Independent generators . . . 172
13.5 Genus . . . 173
13.6 Cohomology of congruence subgroups . . . 174
13.7 Cuspidal cohomology . . . 176
13.8 Hecke operators on forms of weight 2 . . . 178
13.9 Hecke operators on forms of weight ≥ 2 . . . 179
13.10 Reconstructing modular forms from cohomology computations . . . 179
13.11 The Picard group . . . 181
13.12 Bianchi groups . . . 183
13.13 (Co)homology of Bianchi groups and $SL_2(\mathcal{O}_{-d})$ . . . 185
13.14 Some other infinite matrix groups . . . 189
13.15 Ideals and finite quotient groups . . . 191

13.16 Congruence subgroups for ideals . . . 193
13.17 First homology . . . 194
13.18 Generators and relators for $SL_3(\mathbb{Z})$ . . . 196

14 Fundamental domains for Bianchi groups **200**
14.1 Bianchi groups . . . 200
14.2 Swan's description of a fundamental domain . . . 200
14.3 Computing a fundamental domain . . . 201
14.4 Examples . . . 201
14.5 Establishing correctness of a fundamental domain . . . 203
14.6 Computing quotient 2-complexes . . . 203
14.7 Computing a free resolution for $SL_2(\mathcal{O}_{-d})$ . . . 206
14.8 Some sanity checks . . . 207
14.9 Group presentations . . . 209
14.10 Finite index subgroups . . . 210
14.11 Totally real quadratic fields / Hilbert modular group . . . 211
14.12 Calling Magma's Voronoi algorithm directly . . . 211

15 Parallel computation **213**
15.1 An embarassingly parallel computation . . . 213
15.2 A non-embarassingly parallel computation . . . 213
15.3 Parallel persistent homology . . . 215
15.4 Parallelism not a substitute for careful serial implementation . . . 215

16 Regular CW-structure on knots (written by Kelvin Killeen) **218**
16.1 Knot complements in the 3-ball . . . 218
16.2 Tubular neighbourhoods . . . 219
16.3 Knotted surface complements in the 4-ball . . . 222

References **232**

Chapter 1

Simplicial complexes & CW complexes

1.1 The Klein bottle as a simplicial complex

The following example constructs the Klein bottle as a simplicial complex K on 9 vertices, and then constructs the cellular chain complex $C_* = C_*(K)$ from which the integral homology groups $H_1(K,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}$, $H_2(K,\mathbb{Z}) = 0$ are computed. The chain complex $D_* = C_* \otimes_{\mathbb{Z}} \mathbb{Z}_2$ is also constructed and used to compute the mod-2 homology vector spaces $H_1(K,\mathbb{Z}_2) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$, $H_2(K,\mathbb{Z}) = \mathbb{Z}_2$. Finally, a presentation $\pi_1(K) = \langle x, y : yxy^{-1}x \rangle$ is computed for the fundamental group of K.

Example

```
gap> 2simplices:=
> [[1,2,5], [2,5,8], [2,3,8], [3,8,9], [1,3,9], [1,4,9],
>  [4,5,8], [4,6,8], [6,8,9], [6,7,9], [4,7,9], [4,5,7],
>  [1,4,6], [1,2,6], [2,6,7], [2,3,7], [3,5,7], [1,3,5]];;
gap> K:=SimplicialComplex(2simplices);
Simplicial complex of dimension 2.

gap> C:=ChainComplex(K);
Chain complex of length 2 in characteristic 0 .

gap> Homology(C,1);
[ 2, 0 ]
gap> Homology(C,2);
[  ]

gap> D:=TensorWithIntegersModP(C,2);
Chain complex of length 2 in characteristic 2 .

gap> Homology(D,1);
2
gap> Homology(D,2);
1

gap> G:=FundamentalGroup(K);
<fp group of size infinity on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(G);
[ f2*f1*f2^-1*f1 ]
```

1.2 Other simplicial surfaces

The following example constructs the real projective plane P, the Klein bottle K and the torus T as simplicial complexes, using the surface genus g as input in the oriented case and $-g$ as input in the unoriented cases. It then confirms that the connected sums $M = K\#P$ and $N = T\#P$ have the same integral homology.

Example

```
gap> P:=ClosedSurface(-1);
Simplicial complex of dimension 2.

gap> K:=ClosedSurface(-2);
Simplicial complex of dimension 2.

gap> T:=ClosedSurface(1);
Simplicial complex of dimension 2.

gap> M:=ConnectedSum(K,P);
Simplicial complex of dimension 2.

gap> N:=ConnectedSum(T,P);
Simplicial complex of dimension 2.

gap> Homology(M,0);
[ 0 ]
gap> Homology(N,0);
[ 0 ]
gap> Homology(M,1);
[ 2, 0, 0 ]
gap> Homology(N,1);
[ 2, 0, 0 ]
gap> Homology(M,2);
[  ]
gap> Homology(N,2);
[  ]
```

1.3 The Quillen complex

Given a group G one can consider the partially ordered set $\mathscr{A}_p(G)$ of all non-trivial elementary abelian p-subgroups of G, the partial order being set inclusion. The order complex $\Delta\mathscr{A}_p(G)$ is a simplicial complex which is called the *Quillen complex* .

The following example constructs the Quillen complex $\Delta\mathscr{A}_2(S_7)$ for the symmetric group of degree 7 and $p = 2$. This simplicial complex involves 11291 simplices, of which 4410 are 2-simplices..

Example

```
gap> K:=QuillenComplex(SymmetricGroup(7),2);
Simplicial complex of dimension 2.

gap> Size(K);
11291
```

```
gap> K!.nrSimplices(2);
4410
```

1.4 The Quillen complex as a reduced CW-complex

Any simplicial complex K can be regarded as a regular CW complex. Different datatypes are used in HAP for these two notions. The following continuation of the above Quillen complex example constructs a regular CW complex Y isomorphic to (i.e. with the same face lattice as) $K = \Delta\mathscr{A}_2(S_7)$. An advantage to working in the category of CW complexes is that it may be possible to find a CW complex X homotopy equivalent to Y but with fewer cells than Y. The cellular chain complex $C_*(X)$ of such a CW complex X is computed by the following commands. From the number of free generators of $C_*(X)$, which correspond to the cells of X, we see that there is a single 0-cell and 160 2-cells. Thus the Quillen complex $$\Delta{\cal A}_2(S_7) \simeq \bigvee_{1\le i\le 160} S^2$$ has the homotopy type of a wedge of 160 2-spheres. This homotopy equivalence is given in [Kso00, (15.1)] where it was obtained by purely theoretical methods.

Example

```
gap> Y:=RegularCWComplex(K);
Regular CW-complex of dimension 2

gap> C:=ChainComplex(Y);
Chain complex of length 2 in characteristic 0 .

gap> C!.dimension(0);
1
gap> C!.dimension(1);
0
gap> C!.dimension(2);
160
```

1.5 Simple homotopy equivalences

For any regular CW complex Y one can look for a sequence of simple homotopy collapses $Y \searrow Y_1 \searrow Y_2 \searrow \ldots \searrow Y_N = X$ with X a smaller, and typically non-regular, CW complex. Such a sequence of collapses can be recorded using what is now known as a *discrete vector field* on Y. The sequence can, for example, be used to produce a chain homotopy equivalence $f\colon C_*Y \to C_*X$ and its chain homotopy inverse $g\colon C_*X \to C_*Y$. The function `ChainComplex(Y)` returns the cellular chain complex $C_*(X)$, wheras the function `ChainComplexOfRegularCWComplex(Y)` returns the chain complex $C_*(Y)$.

For the above Quillen complex $Y = \Delta\mathscr{A}_2(S_7)$ the following commands produce the chain homotopy equivalence $f\colon C_*Y \to C_*X$ and $g\colon C_*X \to C_*Y$. The number of generators of C_*Y equals the number of cells of Y in each degree, and this number is listed for each degree.

Example

```
gap> K:=QuillenComplex(SymmetricGroup(7),2);;
gap> Y:=RegularCWComplex(K);;
gap> CY:=ChainComplexOfRegularCWComplex(Y);
Chain complex of length 2 in characteristic 0 .
```

```
gap> CX:=ChainComplex(Y);
Chain complex of length 2 in characteristic 0 .

gap> equiv:=ChainComplexEquivalenceOfRegularCWComplex(Y);;
gap> f:=equiv[1];
Chain Map between complexes of length 2 .

gap> g:=equiv[2];
Chain Map between complexes of length 2 .


gap> CY!.dimension(0);
1316
gap> CY!.dimension(1);
5565
gap> CY!.dimension(2);
4410
```

1.6 Cellular simplifications preserving homeomorphism type

For some purposes one might need to simplify the cell structure on a regular CW-complex Y so as to obtained a homeomorphic CW-complex W with fewer cells.

The following commands load a 4-dimensional simplicial complex Y representing the K3 complex surface. Its simplicial structure is taken from [SK11] and involves 1704 cells of various dimensions. The commands then convert the cell structure into that of a homeomorphic regular CW-complex W involving 774 cells.

Example

```
gap> Y:=RegularCWComplex(SimplicialK3Surface());
Regular CW-complex of dimension 4

gap> Size(Y);
1704
gap> W:=SimplifiedComplex(Y);
Regular CW-complex of dimension 4

gap> Size(W);
774
```

1.7 Constructing a CW-structure on a knot complement

The following commands construct the complement $M = S^3 \setminus K$ of the trefoil knot K. This complement is returned as a 3-manifold M with regular CW-structure involving four 3-cells.

Example

```
gap> arc:=ArcPresentation(PureCubicalKnot(3,1));
[ [ 2, 5 ], [ 1, 3 ], [ 2, 4 ], [ 3, 5 ], [ 1, 4 ] ]
gap> S:=SphericalKnotComplement(arc);
```

```
Regular CW-complex of dimension 3

gap> S!.nrCells(3);
4
```

The following additional commands then show that M is homotopy equivalent to a reduced CW-complex Y of dimension 2 involving one 0-cell, two 1-cells and one 2-cell. The fundamental group of Y is computed and used to calculate the Alexander polynomial of the trefoil knot.

Example

```
gap> Y:=ContractedComplex(S);
Regular CW-complex of dimension 2

gap> CriticalCells(Y);
[ [ 2, 1 ], [ 1, 9 ], [ 1, 11 ], [ 0, 22 ] ]
gap> G:=FundamentalGroup(Y);;
gap> AlexanderPolynomial(G);
x_1^2-x_1+1
```

1.8 Constructing a regular CW-complex by attaching cells

The following example creates the projective plane Y as a regular CW-complex, and tests that it has the correct integral homology $H_0(Y,\mathbb{Z}) = \mathbb{Z}$, $H_1(Y,\mathbb{Z}) = \mathbb{Z}_2$, $H_2(Y,\mathbb{Z}) = 0$.

Example

```
gap> attch:=RegularCWComplex_AttachCellDestructive;; #Function for attaching cells

gap> Y:=RegularCWDiscreteSpace(3); #Discrete CW-complex consisting of points {1,2,3}
Regular CW-complex of dimension 0

gap> e1:=attch(Y,1,[1,2]);; #Attach 1-cell
gap> e2:=attch(Y,1,[1,2]);; #Attach 1-cell
gap> e3:=attch(Y,1,[1,3]);; #Attach 1-cell
gap> e4:=attch(Y,1,[1,3]);; #Attach 1-cell
gap> e5:=attch(Y,1,[2,3]);; #Attach 1-cell
gap> e6:=attch(Y,1,[2,3]);; #Attach 1-cell
gap> f1:=attch(Y,2,[e1,e3,e5]);; #Attach 2-cell
gap> f2:=attch(Y,2,[e2,e4,e5]);; #Attach 2-cell
gap> f3:=attch(Y,2,[e2,e3,e6]);; #Attach 2-cell
gap> f4:=attch(Y,2,[e1,e4,e6]);; #Attach 2-cell
gap> Homology(Y,0);
[ 0 ]
gap> Homology(Y,1);
[ 2 ]
gap> Homology(Y,2);
[  ]‘
```

The following example creates a 2-complex K corresponding to the group presentation $G = \langle x,y,z \,:\, xyx^{-1}y^{-1} = 1, yzy^{-1}z^{-1} = 1, zxz^{-1}x^{-1} = 1\rangle$.

The complex is shown to have the correct fundamental group and homology (since it is the 2-skeleton of the 3-torus $S^1 \times S^1 \times S^1$).

Example

```
gap> S1:=RegularCWSphere(1);;
gap> W:=WedgeSum(S1,S1,S1);;
gap> F:=FundamentalGroupWithPathReps(W);; x:=F.1;;y:=F.2;;z:=F.3;;
gap> K:=RegularCWComplexWithAttachedRelatorCells(W,F,Comm(x,y),Comm(y,z),Comm(x,z));
Regular CW-complex of dimension 2

gap> G:=FundamentalGroup(K);
<fp group on the generators [ f1, f2, f3 ]>
gap> RelatorsOfFpGroup(G);
[ f2^-1*f1*f2*f1^-1, f1^-1*f3*f1*f3^-1, f2^-1*f3*f2*f3^-1 ]
gap> Homology(K,1);
[ 0, 0, 0 ]
gap> Homology(K,2);
[ 0, 0, 0 ]
```

1.9 Constructing a regular CW-complex from its face lattice

The following example creats a 2-dimensional annulus A as a regular CW-complex, and testing that it has the correct integral homology $H_0(A,\mathbb{Z}) = \mathbb{Z}$, $H_1(A,\mathbb{Z}) = \mathbb{Z}$, $H_2(A,\mathbb{Z}) = 0$.

Example

```
gap> FL:=[];; #The face lattice
gap> FL[1]:=[[1,0],[1,0],[1,0],[1,0]];;
gap> FL[2]:=[[2,1,2],[2,3,4],[2,1,4],[2,2,3],[2,1,4],[2,2,3]];;
gap> FL[3]:=[[4,1,2,3,4],[4,1,2,5,6]];;
gap> FL[4]:=[];;
gap> A:=RegularCWComplex(FL);
Regular CW-complex of dimension 2

gap> Homology(A,0);
[ 0 ]
gap> Homology(A,1);
[ 0 ]
gap> Homology(A,2);
[  ]
```

Next we construct the direct product $Y = A \times A \times A \times A \times A$ of five copies of the annulus. This is a 10-dimensional CW complex involving 248832 cells. It will be homotopy equivalent $Y \simeq X$ to a CW complex X involving fewer cells. The CW complex X may be non-regular. We compute the cochain complex $D_* = \mathrm{Hom}_{\mathbb{Z}}(C_*(X),\mathbb{Z})$ from which the cohomology groups
$H^0(Y,\mathbb{Z}) = \mathbb{Z}$,
$H^1(Y,\mathbb{Z}) = \mathbb{Z}^5$,
$H^2(Y,\mathbb{Z}) = \mathbb{Z}^{10}$,
$H^3(Y,\mathbb{Z}) = \mathbb{Z}^{10}$,
$H^4(Y,\mathbb{Z}) = \mathbb{Z}^5$,

$H^5(Y,\mathbb{Z}) = \mathbb{Z}$,
$H^6(Y,\mathbb{Z}) = 0$
are obtained.

Example

```
gap> Y:=DirectProduct(A,A,A,A,A);
Regular CW-complex of dimension 10

gap> Size(Y);
248832
gap> C:=ChainComplex(Y);
Chain complex of length 10 in characteristic 0 .

gap> D:=HomToIntegers(C);
Cochain complex of length 10 in characteristic 0 .

gap> Cohomology(D,0);
[ 0 ]
gap> Cohomology(D,1);
[ 0, 0, 0, 0, 0 ]
gap> Cohomology(D,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Cohomology(D,3);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Cohomology(D,4);
[ 0, 0, 0, 0, 0 ]
gap> Cohomology(D,5);
[ 0 ]
gap> Cohomology(D,6);
[  ]
```

1.10 Cup products

STRATEGY 1: USE GEOMETRIC GROUP THEORY IN LOW DIMENSIONS.

Continuing with the previous example, we consider the first and fifth generators $g_1^1, g_5^1 \in H^1(Y,\mathbb{Z}) = \mathbb{Z}^5$ and establish that their cup product $g_1^1 \cup g_5^1 = -g_7^2 \in H^2(Y,\mathbb{Z}) = \mathbb{Z}^{10}$ is equal to minus the seventh generator of $H^2(Y,\mathbb{Z})$. We also verify that $g_5^1 \cup g_1^1 = -g_1^1 \cup g_5^1$.

Example

```
gap> cup11:=CupProduct(FundamentalGroup(Y));
function( a, b ) ... end

gap> cup11([1,0,0,0,0],[0,0,0,0,1]);
[ 0, 0, 0, 0, 0, 0, -1, 0, 0, 0 ]

gap> cup11([0,0,0,0,1],[1,0,0,0,0]);
[ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 ]
```

This computation of low-dimensional cup products is achieved using group-theoretic methods to approximate the diagonal map $\Delta: Y \to Y \times Y$ in dimensions ≤ 2. In order to construct cup products in higher degrees HAP invokes three further strategies.

STRATEGY 2: IMPLEMENT THE ALEXANDER-WHITNEY MAP FOR SIMPLICIAL COMPLEXES.

For simplicial complexes the cup product is implemented using the standard formula for the Alexander-Whitney chain map, together with homotopy equivalences to improve efficiency.

As a first example, the following commands construct simplicial complexes $K = (\mathbb{S}^1 \times \mathbb{S}^1)\#(\mathbb{S}^1 \times \mathbb{S}^1)$ and $L = (\mathbb{S}^1 \times \mathbb{S}^1) \vee \mathbb{S}^1 \vee \mathbb{S}^1$ and establish that they have the same cohomology groups. It is then shown that the cup products $\cup_K: H^2(K,\mathbb{Z}) \times H^2(K,\mathbb{Z}) \to H^4(K,\mathbb{Z})$ and $\cup_L: H^2(L,\mathbb{Z}) \times H^2(L,\mathbb{Z}) \to H^4(L,\mathbb{Z})$ are antisymmetric bilinear forms of different ranks; hence K and L have different homotopy types.

Example

```
gap> K:=ClosedSurface(2);
Simplicial complex of dimension 2.

gap> L:=WedgeSum(WedgeSum(ClosedSurface(1),Sphere(1)),Sphere(1));
Simplicial complex of dimension 2.

gap> Cohomology(K,0);Cohomology(L,0);
[ 0 ]
[ 0 ]
gap> Cohomology(K,1);Cohomology(L,1);
[ 0, 0, 0, 0 ]
[ 0, 0, 0, 0 ]
gap> Cohomology(K,2);Cohomology(L,2);
[ 0 ]
[ 0 ]
gap> gens:=[[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]];;
gap> cupK:=CupProduct(K);;
gap> cupL:=CupProduct(L);;
gap> A:=NullMat(4,4);;B:=NullMat(4,4);;
gap> for i in [1..4] do
> for j in [1..4] do
> A[i][j]:=cupK(1,1,gens[i],gens[j])[1];
> B[i][j]:=cupL(1,1,gens[i],gens[j])[1];
> od;od;
gap> Display(A);
[ [  0,  0,  0,  1 ],
  [  0,  0,  1,  0 ],
  [  0, -1,  0,  0 ],
  [ -1,  0,  0,  0 ] ]
gap> Display(B);
[ [  0,  1,  0,  0 ],
  [ -1,  0,  0,  0 ],
  [  0,  0,  0,  0 ],
  [  0,  0,  0,  0 ] ]
gap> Rank(A);
4
gap> Rank(B);
2
```

As a second example of the computation of cups products, the following commands construct the connected sums $V = M\#M$ and $W = M\#\overline{M}$ where M is the $K3$ complex surface which is stored as a pure simplicial complex of dimension 4 and where $\overline{M}$ denotes the opposite orientation on M. The simplicial structure on the $K3$ surface is taken from [SK11]. The commands then show that $H^2(V,\mathbb{Z}) = H^2(W,\mathbb{Z}) = \mathbb{Z}^{44}$ and $H^4(V,\mathbb{Z}) = H^4(W,\mathbb{Z}) = \mathbb{Z}$. The final commands compute the matrix $AV = (x \cup y)$ as x, y range over a generating set for $H^2(V,\mathbb{Z})$ and the corresponding matrix AW for W. These two matrices are seen to have a different number of positive eigenvalues from which we can conclude that V is not homotopy equivalent to W.

Example

```
gap> M:=SimplicialK3Surface();;
gap> V:=ConnectedSum(M,M,+1);
Simplicial complex of dimension 4.

gap> W:=ConnectedSum(M,M,-1);
Simplicial complex of dimension 4.

gap> Cohomology(V,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Cohomology(W,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Cohomology(V,4);
[ 0 ]
gap> Cohomology(W,4);
[ 0 ]
gap> cupV:=CupProduct(V);;
gap> cupW:=CupProduct(W);;
gap> AV:=NullMat(44,44);;
gap> AW:=NullMat(44,44);;
gap> gens:=IdentityMat(44);;
gap> for i in [1..44] do
> for j in [1..44] do
> AV[i][j]:=cupV(2,2,gens[i],gens[j])[1];
> AW[i][j]:=cupW(2,2,gens[i],gens[j])[1];
> od;od;
gap> SignatureOfSymmetricMatrix(AV);
rec( determinant := 1, negative_eigenvalues := 22, positive_eigenvalues := 22,
  zero_eigenvalues := 0 )
gap> SignatureOfSymmetricMatrix(AW);
rec( determinant := 1, negative_eigenvalues := 6, positive_eigenvalues := 38,
  zero_eigenvalues := 0 )
```

A cubical cubical version of the Alexander-Whitney formula, due to J.-P. Serre, could be used for computing the cohomology ring of a regular CW-complex whose cells all have a cubical combinatorial face lattice. This has not been implemented in HAP. However, the following more general approach has been implemented.

STRATEGY 3: IMPLEMENT A CELLULAR APPROXIMATION TO THE DIAGONAL MAP ON AN ARBITRARY FINITE REGULAR CW-COMPLEX.

The following example calculates the cup product $H^2(W,\mathbb{Z}) \times H^2(W,\mathbb{Z}) \to H^4(W,\mathbb{Z})$ for the

4-dimensional orientable manifold $W = M \times M$ where M is the closed surface of genus 2. The manifold W is stored as a regular CW-complex.

Example

```
gap> M:=RegularCWComplex(ClosedSurface(2));;
gap> W:=DirectProduct(M,M);
Regular CW-complex of dimension 4

gap> Size(W);
5776
gap> W:=SimplifiedComplex(W);;
gap> Size(W);
1024

gap> Homology(W,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(W,4);
[ 0 ]

gap> cup:=CupProduct(W);;
gap> SecondCohomologtGens:=IdentityMat(18);;
gap> A:=NullMat(18,18);;
gap> for i in [1..18] do
> for j in [1..18] do
> A[i][j]:=cup(2,2,SecondCohomologtGens[i],SecondCohomologtGens[j])[1];
> od;od;
gap> Display(A);
[ [    0,   -1,    0,    0,    0,    0,    3,   -2,    0,    0,    0,    1,   -1,    0,    0,
  [   -1,  -10,    1,    2,   -2,    1,    6,   -1,    0,   -3,    4,   -1,   -1,   -1,    4,
  [    0,    1,   -2,    1,    0,   -1,    0,    0,    1,    0,   -1,    1,    0,    0,    1,
  [    0,    2,    1,   -2,    1,    0,    0,   -1,    0,    1,    0,    0,    0,    0,   -1,
  [    0,   -2,    0,    1,    0,    0,    1,   -1,    0,    0,   -1,    0,    0,    0,    0,
  [    0,    1,   -1,    0,    0,    0,    0,    1,   -1,    1,    0,    0,    0,    0,    1,
  [    3,    6,    0,    0,    1,    0,   -4,    0,   -1,    2,    4,   -5,    2,   -1,    1,
  [   -2,   -1,    0,   -1,   -1,    1,    0,    4,   -2,    0,    0,    3,   -1,    1,   -1,
  [    0,    0,    1,    0,    0,   -1,   -1,   -2,    4,   -3,  -10,    1,    0,    0,   -3,
  [    0,   -3,    0,    1,    0,    1,    2,    0,   -3,    2,    3,    0,    0,    0,    1,
  [    0,    4,   -1,    0,   -1,    0,    4,    0,  -10,    3,   18,    1,    0,    0,    0,
  [    1,   -1,    1,    0,    0,    0,   -5,    3,    1,    0,    1,    0,    0,    0,   -2,
  [   -1,   -1,    0,    0,    0,    0,    2,   -1,    0,    0,    0,    0,    0,    0,    1,
  [    0,   -1,    0,    0,    0,    0,   -1,    1,    0,    0,    0,    0,    0,    0,    0,
  [    0,    4,    1,   -1,    0,    1,    1,   -1,   -3,    1,    0,   -2,    1,    0,    0,
  [    1,   -2,   -1,    2,   -1,   -1,    0,    0,    3,   -3,    4,   -1,    0,   -1,    2,
  [    0,   -2,    0,    0,    0,    0,    3,   -2,    0,    0,    0,   -1,    0,   -1,    2,
  [    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    1,    0,    0,    0,    0,

gap> SignatureOfSymmetricMatrix(A);
rec( determinant := -1, negative_eigenvalues := 9, positive_eigenvalues := 9,
  zero_eigenvalues := 0 )
```

The matrix A representing the cup product $H^2(W,\mathbb{Z}) \times H^2(W,\mathbb{Z}) \to H^4(W,\mathbb{Z})$ is shown to have 9 positive eigenvalues, 9 negative eigenvalues, and no zero eigenvalue.

STRATEGY 4: GUESS AND VERIFY A CELLULAR APPROXIMATION TO THE DIAGONAL MAP.

Many naturally occuring cell structures are neither simplicial nor cubical. For a general regular CW-complex we can attempt to construct a cellular inclusion $\overline{Y} \hookrightarrow Y \times Y$ with $\{(y,y) \ : \ y \in Y\} \subset \overline{Y}$ and with projection $p\colon \overline{Y} \twoheadrightarrow Y$ that induces isomorphisms on integral homology. The function `DiagonalApproximation(Y)` constructs a candidate inclusion, but the projection $p\colon \overline{Y} \twoheadrightarrow Y$ needs to be tested for homology equivalence. If the candidate inclusion passes this test then the function `CupProductOfRegularCWComplex_alt(Y)`, involving the candidate space, can be used for cup products. (I think the test is passed for all regular CW-complexes that are subcomplexes of some Euclidean space with all cells convex polytopes -- but a proof needs to be written down!)

The following example calculates $g_1^2 \cup g_2^2 \neq 0$ where $Y = T \times T$ is the direct product of two copies of a simplicial torus T, and where g_k^n denotes the k-th generator in some basis of $H^n(Y,\mathbb{Z})$. The direct product Y is a CW-complex which is not a simplicial complex.

Example

```
gap> K:=RegularCWComplex(ClosedSurface(1));;
gap> Y:=DirectProduct(K,K);;
gap> cup:=CupProductOfRegularCWComplex_alt(Y);;
gap> cup(2,2,[1,0,0,0,0,0],[0,1,0,0,0,0]);
[ 5 ]

gap> D:=DiagonalApproximation(Y);;
gap> p:=D!.projection;
Map of regular CW-complexes

gap> P:=ChainMap(p);
Chain Map between complexes of length 4 .

gap> IsIsomorphismOfAbelianFpGroups(Homology(P,0));
true
gap> IsIsomorphismOfAbelianFpGroups(Homology(P,2));
true
gap> IsIsomorphismOfAbelianFpGroups(Homology(P,3));
true
gap> IsIsomorphismOfAbelianFpGroups(Homology(P,4));
true
```

Of course, either of Strategies 2 or 3 could also be used for this example. To use the Alexander-Whitney formula of Strategy 2 we would need to give the direct product $Y = T \times T$ a simplicial structure. This could be obtained using the function `DirectProduct(T,T)`. The details are as follows. (The result is consistent with the preceding computation since the choice of a basis for cohomology groups is far from unique.)

Example

```
gap> K:=ClosedSurface(1);;
gap> KK:=DirectProduct(K,K);
Simplicial complex of dimension 4.

gap> cup:=CupProduct(KK);;
gap> cup(2,2,[1,0,0,0,0,0],[0,1,0,0,0,0]);
[ 0 ]
```

1.11 Intersection forms of 4-manifolds

The cup product gives rise to the intersection form of a connected, closed, orientable 4-manifold Y is a symmetric bilinear form

$qY: H^2(Y,\mathbb{Z})/Torsion \times H^2(Y,\mathbb{Z})/Torsion \longrightarrow \mathbb{Z}$

which we represent as a symmetric matrix.

The following example constructs the direct product $L = S^2 \times S^2$ of two 2-spheres, the connected sum $M = \mathbb{C}P^2 \# \overline{\mathbb{C}P^2}$ of the complex projective plane $\mathbb{C}P^2$ and its oppositely oriented version $\overline{\mathbb{C}P^2}$, and the connected sum $N = \mathbb{C}P^2 \# \mathbb{C}P^2$. The manifolds L, M and N are each shown to have a CW-structure involving one 0-cell, two 1-cells and one 2-cell. They are thus simply connected and have identical cohomology.

Example

```
gap> S:=Sphere(2);;
gap> S:=RegularCWComplex(S);;
gap> L:=DirectProduct(S,S);
Regular CW-complex of dimension 4

gap> M:=ConnectedSum(ComplexProjectiveSpace(2),ComplexProjectiveSpace(2),-1);
Simplicial complex of dimension 4.

gap> N:=ConnectedSum(ComplexProjectiveSpace(2),ComplexProjectiveSpace(2),+1);
Simplicial complex of dimension 4.

gap> CriticalCells(L);
[ [ 4, 1 ], [ 2, 13 ], [ 2, 56 ], [ 0, 16 ] ]
gap> CriticalCells(RegularCWComplex(M));
[ [ 4, 1 ], [ 2, 109 ], [ 2, 119 ], [ 0, 8 ] ]
gap> CriticalCells(RegularCWComplex(N));
[ [ 4, 1 ], [ 2, 119 ], [ 2, 149 ], [ 0, 12 ] ]
```

John Milnor showed (as a corollary to a theorem of J. H. C. Whitehead) that the homotopy type of a simply connected 4-manifold is determined by its quadratic form. More precisely, a form is said to be of *type I (properly primitive)* if some diagonal entry of its matrix is odd. If every diagonal entry is even, then the form is of *type II (improperly primitive)*. The *index* of a form is defined as the number of positive diagonal entries minus the number of negative ones, after the matrix has been diagonalized over the real numbers.

THEOREM. (Milnor [Mil58]) The oriented homotopy type of a simply connected, closed, orientable 4-manifold is determined by its second Betti number and the index and type of its intersetion form; except possibly in the case of a manifold with definite quadratic form of rank r > 9.

The following commands compute matrices representing the intersection forms qL, qM, qN.

Example

```
gap> qL:=IntersectionForm(L);;
gap> qM:=IntersectionForm(M);;
gap> qN:=IntersectionForm(N);;
gap> Display(qL);
[ [  -2,   1 ],
  [   1,   0 ] ]
gap> Display(qM);
[ [  1,  0 ],
```

```
    [  0,  1 ] ]
gap> Display(qN);
[ [   1,   0 ],
  [   0,  -1 ] ]
```

Since qL is of type II, whereas qM and qN are of type I we see that the oriented homotopy type of L is distinct to that of M and that of N. Since qM has index 2 and qN has index 0 we see that that M and N also have distinct oriented homotopy types.

1.12 Cohomology Rings

The cup product gives the cohomology $H^*(X,R)$ of a space X with coefficients in a ring R the structure of a graded commutitive ring. The function `CohomologyRing(Y,p)` returns the cohomology as an algebra for Y a simplicial complex and $R=\mathbb{Z}_p$ the field of p elements. For more general regular CW-complexes or $R=\mathbb{Z}$ the cohomology ring structure can be determined using the function `CupProduct(Y)`.

The folowing commands compute the mod 2 cohomology ring $H^*(W,\mathbb{Z}_2)$ of the above wedge sum $W=M\vee N$ of a 2-dimensional orientable simplicial surface of genus 2 and the $K3$ complex simplicial surface (of real dimension 4).

Example

```
gap> M:=ClosedSurface(2);;
gap> N:=SimplicialK3Surface();;
gap> W:=WedgeSum(M,N);;
gap> A:=CohomologyRing(W,2);
<algebra of dimension 29 over GF(2)>
gap> x:=Basis(A)[25];
v.25
gap> y:=Basis(A)[27];
v.27
gap> x*y;
v.29
```

The functions `CupProduct` and `IntersectionForm` can be used to determine integral cohomology rings. For example, the integral cohomology ring of an arbitrary closed surface was calculated in [GM15, Theorem 3.5]. For any given surface M this result can be recalculated using the intersection form. For instance, for an orientable surface of genus g it is well-known that $H^1(M,\mathbb{Z})=\mathbb{Z}^{2g}$, $H^2(M,\mathbb{Z})=\mathbb{Z}$. The ring structure multiplication is thus given by the matrix of the intersection form. For say $g=3$ the ring multiplication is given, with respect to some cohomology basis, in the following.

Example

```
gap> M:=ClosedSurface(3);;
gap> Display(IntersectionForm(M));
[ [   0,   0,   1,  -1,  -1,   0 ],
  [   0,   0,   0,   1,   1,   0 ],
  [  -1,   0,   0,   1,   1,  -1 ],
  [   1,  -1,  -1,   0,   0,   1 ],
  [   1,  -1,  -1,   0,   0,   0 ],
  [   0,   0,   1,  -1,   0,   0 ] ]
```

By changing the basis B for $H^1(M,\mathbb{Z})$ we obtain the following simpler matrix representing multiplication in $H^*(M,\mathbb{Z})$.

Example

```
gap> B:=[ [ 0, 1, -1, -1, 1, 0 ],
>         [ 1, 0, 1, 1, 0, 0 ],
>         [ 0, 0, 1, 0, 0, 0 ],
>         [ 0, 0, 0, 1, -1, 0 ],
>         [ 0, 0, 1, 1, 0, 0 ],
>         [ 0, 0, 1, 1, 0, 1 ] ];;
gap> Display(IntersectionForm(M,B));
[ [   0,   1,   0,   0,   0,   0 ],
  [  -1,   0,   0,   0,   0,   0 ],
  [   0,   0,   0,   0,   1,   0 ],
  [   0,   0,   0,   0,   0,   1 ],
  [   0,   0,  -1,   0,   0,   0 ],
  [   0,   0,   0,  -1,   0,   0 ] ]
```

1.13 Bockstein homomorphism

The following example evaluates the Bockstein homomorphism $\beta_2: H^*(X,\mathbb{Z}_2) \to H^{*+1}(X,\mathbb{Z}_2)$ on an additive basis for $X = \Sigma^{100}(\mathbb{R}P^2 \times \mathbb{R}P^2)$ the 100-fold suspension of the direct product of two projective planes.

Example

```
gap> P:=SimplifiedComplex(RegularCWComplex(ClosedSurface(-1)));
Regular CW-complex of dimension 2
gap> PP:=DirectProduct(P,P);;
gap> SPP:=Suspension(PP,100);
Regular CW-complex of dimension 104
gap> A:=CohomologyRing(SPP,2);
<algebra of dimension 9 over GF(2)>
gap> List(Basis(A),x->Bockstein(A,x));
[ 0*v.1, v.4, v.6, 0*v.1, v.7+v.8, 0*v.1, v.9, v.9, 0*v.1 ]
```

If only the Bockstein homomorphism is required, and not the cohomology ring structure, then the Bockstein could also be computedirectly from a chain complex. The following computes the Bockstein $\beta_2: H^2(Y,\mathbb{Z}_2) \to H^3(Y,\mathbb{Z}_2)$ for the direct product $Y = K \times K \times K \times K$ of four copies of the Klein bottle represented as a regular CW-complex with 331776 cells. The order of the kernel and image of β_2 are computed.

Example

```
gap> K:=ClosedSurface(-2);;
gap> K:=SimplifiedComplex(RegularCWComplex(K));;
gap> KKKK:=DirectProduct(K,K,K,K);
Regular CW-complex of dimension 8
gap> Size(KKKK);
331776
gap> C:=ChainComplex(KKKK);;
gap> bk:=Bockstein(C,2,2);;
```

```
gap> Order(Kernel(bk));
1024
gap> Order(Image(bk));
262144
```

1.14 Diagonal maps on associahedra and other polytopes

By a *diagonal approximation* on a regular CW-complex X we mean any cellular map $\Delta: X \to X \times X$ that is homotopic to the diagonal map $X \to X \times X, x \mapsto (x,x)$ and equal to the diagonal map when restricted to the 0-skeleton. Theoretical formulae for diagonal maps on a polytope X can have interesting combinatorial aspects. To illustrate this let us consider, for $n = 3$, the n-dimensional polytope $\mathscr{K}^{n+2}$ known as the associahedron. The following commands display the 1-skeleton of $\mathscr{K}^5$.

Example

```
gap> n:=3;;Y:=RegularCWAssociahedron(n+2);;
gap> Display(GraphOfRegularCWComplex(Y));
```

The induced chain map $C_*(\mathscr{K}^{n+2}) \to C_*(\mathscr{K}^{n+2} \times \mathscr{K}^{n+2})$ sends the unique free generator e_1^n of $C_n(\mathscr{K}^{n+2})$ to a sum $\Delta(e_1^n)$ of a number of distinct free generators of $C_n(\mathscr{K}^{n+2} \times \mathscr{K}^{n+2})$. Let $|\Delta(e_1^n)|$ denote the number of free generators. For $n = 3$ the following commands show that $|\Delta(e_1^3)| = 22$ with each free generator occurring with coefficient ± 1.

Example

```
gap> n:=3;;Y:=RegularCWAssociahedron(n+2);;
gap> D:=DiagonalChainMap(Y);;Filtered(D!.mapping([1],n),x->x<>0);
[ 1, 1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, 1, 1, 1, -1, -1, 1, 1, 1, 1 ]
```

Repeating this example for $0 \leq n \leq 6$ yields the sequence $|\Delta(e_1^n)| : 1,2,6,22,91,408,1938,\cdots$. The On-line Encyclopedia of Integer Sequences explains that this is the beginning of the sequence given by the number of canopy intervals in the Tamari lattices.

Repeating the same experiment for the permutahedron, using the command `RegularCWPermutahedron(n)`, yields the sequence $|\Delta(e_1^n)| : 1,2,8,50,432,4802,\cdots$. The On-line Encyclopedia of Integer Sequences explains that this is the beginning of the sequence given by the number of spanning trees in the graph K_n/e, which results from contracting an edge e in the complete graph K_n on n vertices.

Repeating the experiment for the cube, using the command `RegularCWCube(n)`, yields the sequence $|\Delta(e_1^n)| : 1,2,4,8,16,32,\cdots$.

Repeating the experiment for the simplex, using the command `RegularCWSimplex(n)`, yields the sequence $|\Delta(e_1^n)| : 1,2,3,4,5,6,\cdots$.

1.15 CW maps and induced homomorphisms

A *strictly cellular* map $f: X \to Y$ of regular CW-complexes is a cellular map for which the image of any cell is a cell (of possibly lower dimension). Inclusions of CW-subcomplexes, and projections

from a direct product to a factor, are examples of such maps. Strictly cellular maps can be represented in HAP, and their induced homomorphisms on (co)homology and on fundamental groups can be computed.

The following example begins by visualizing the trefoil knot $\kappa \in \mathbb{R}^3$. It then constructs a regular CW structure on the complement $Y = D^3 \setminus \mathrm{Nbhd}(\kappa)$ of a small tubular open neighbourhood of the knot lying inside a large closed ball D^3. The boundary of this tubular neighbourhood is a 2-dimensional CW-complex B homeomorphic to a torus $\mathbb{S}^1 \times \mathbb{S}^1$ with fundamental group $\pi_1(B) =< a,b : aba^{-1}b^{-1} = 1 >$. The inclusion map $f: B \hookrightarrow Y$ is constructed. Then a presentation $\pi_1(Y) =< x,y \,|\, xy^{-1}x^{-1}yx^{-1}y^{-1} >$ and the induced homomorphism $$\pi_1(B)\rightarrow \pi_1(Y), a\mapsto y^{-1}xy^2xy^{-1}, b\mapsto y $$ are computed. This induced homomorphism is an example of a *peripheral system* and is known to contain sufficient information to characterize the knot up to ambient isotopy.

Finally, it is verified that the induced homology homomorphism $H_2(B,\mathbb{Z}) \to H_2(Y,\mathbb{Z})$ is an isomorphism.

Example

```
gap> K:=PureCubicalKnot(3,1);;
gap> ViewPureCubicalKnot(K);;
```

Example

```
gap> K:=PureCubicalKnot(3,1);;
gap> f:=KnotComplementWithBoundary(ArcPresentation(K));
Map of regular CW-complexes

gap> G:=FundamentalGroup(Target(f));
<fp group of size infinity on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(G);
[ f1*f2^-1*f1^-1*f2*f1^-1*f2^-1 ]

gap> F:=FundamentalGroup(f);
[ f1, f2 ] -> [ f2^-1*f1*f2^2*f1*f2^-1, f1 ]


gap> phi:=ChainMap(f);
Chain Map between complexes of length 2 .

gap> H:=Homology(phi,2);
[ g1 ] -> [ g1 ]
```

1.16 Constructing a simplicial complex from a regular CW-complex

The following example constructs a 3-dimensional pure regular CW-complex K whose 3-cells are permutahedra. It then constructs the simplicial complex B by taking barycentric subdivision. It then constructes a smaller, homotopy equivalent, simplicial complex N by taking the nerve of the cover of K by the closures of its 3-cells.

Example

```
gap> K:=RegularCWComplex(PureComplexComplement(PurePermutahedralKnot(3,1)));
Regular CW-complex of dimension 3
```

```
gap> Size(K);
77923
gap> B:=BarycentricSubdivision(K);
Simplicial complex of dimension 3.

gap> Size(B);
1622517
gap> N:=Nerve(K);
Simplicial complex of dimension 3.

gap> Size(N);
48745
```

1.17 Some limitations to representing spaces as regular CW complexes

By a *classifying space* for a group G we mean a path-connected space BG with fundamental group $\pi_1(BG) \cong G$ isomorphic to G and with higher homotopy groups $\pi_n(BG) = 0$ trivial for all $n \geq 2$. The homology of the group G can be defined to be the homology of BG: $H_n(G,\mathbb{Z}) = H_n(BG,\mathbb{Z})$.

In principle BG can always be constructed as a regular CW-complex. For instance, the following extremely slow commands construct the 5-skeleton Y^5 of a regular CW-classifying space $Y = BG$ for the dihedral group of order 16 and use it to calculate $H_1(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$, $H_2(G,\mathbb{Z}) = \mathbb{Z}_2$, $H_3(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_8$, $H_4(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$. The final command shows that the constructed space Y^5 in this example is a 5-dimensional regular CW-complex with a total of 15289 cells.

Example

```
gap> Y:=ClassifyingSpaceFiniteGroup(DihedralGroup(16),5);
Regular CW-complex of dimension 5
gap> Homology(Y,1);
[ 2, 2 ]
gap> Homology(Y,2);
[ 2 ]
gap> Homology(Y,3);
[ 2, 2, 8 ]
gap> Homology(Y,4);
[ 2, 2 ]
gap> Size(Y);
15289
```

The n-skeleton of a regular CW-classifying space of a finite group necessarily involves a large number of cells. For the group $G = C_2$ of order two a classifying space can be take to be real projective space $BG = \mathbb{R}P^\infty$ with n-skeleton $BG^n = \mathbb{R}P^n$. To realize $BG^n = \mathbb{R}P^n$ as a simplicial complex it is known that one needs at least 6 vertices for $n = 2$, at least 11 vertices for $n = 3$ and at least 16 vertices for $n = 4$. One can do a bit better by allowing BG to be a regular CW-complex. For instance, the following creates $\mathbb{R}P^4$ as a regular CW-complex with 5 vertices. This construction of $\mathbb{R}P^4$ involves a total of 121 cells. A minimal triangulation of $\mathbb{R}P^4$ would require 991 simplices.

Example

```
gap> Y:=ClassifyingSpaceFiniteGroup(CyclicGroup(2),4);
Regular CW-complex of dimension 4
gap> Y!.nrCells(0);
5
gap> Y!.nrCells(1);
20
gap> Y!.nrCells(2);
40
gap> Y!.nrCells(3);
40
gap> Y!.nrCells(4);
16
```

The space $\mathbb{R}P^n$ can be given the structure of a regular CW-complex with $n+1$ vertices. Kuehnel has described a triangulation of $\mathbb{R}P^n$ with $2^{n+1}-1$ vertices.

The above examples suggest that it is inefficient/impractical to attempt to compute the n-th homology of a group G by first constructing a regular CW-complex corresponding for the $n+1$ of a classifying space BG, even for quite small groups G, since such spaces seem to require a large number of cells in each dimension. On the other hand, by dropping the requirement that BG must be regular we can obtain much smaller CW-complexes. The following example constructs $\mathbb{R}P^9$ as a regular CW-complex and then shows that it can be given a non-regular CW-structure with just one cell in each dimension.

Example

```
gap> Y:=ClassifyingSpaceFiniteGroup(CyclicGroup(2),9);
Regular CW-complex of dimension 9
gap> Size(Y);
29524
gap> CriticalCells(Y);
[ [ 9, 1 ], [ 8, 124 ], [ 7, 1215 ], [ 6, 1246 ], [ 5, 487 ], [ 4, 254 ],
  [ 3, 117 ], [ 2, 54 ], [ 1, 9 ], [ 0, 10 ] ]
```

It is of course well-known that $\mathbb{R}P^\infty$ admits a theoretically described CW-structure with just one cell in each dimension. The question is: how best to represent this on a computer?

1.18 Equivariant CW complexes

As just explained, the representations of spaces as simplicial complexes and regular CW complexes have their limitations. One limitation is that the number of cells needed to describe a space can be unnecessarily large. A minimal simplicial complex structure for the torus has 7 vertices, 21 edges and 14 triangles. A minimal regular CW-complex structure for the torus has 4 vertices, 8 edges and 4 cells of dimension 2. By using simplicial sets (which are like simplicial complexes except that they allow the freedom to attach simplicial cells by gluing their boundary non-homeomorphically) one obtains a minimal triangulation of the torus involving 1 vertex, 3 edges and 2 cells of dimension 2. By using non-regular CW-complexes one obtains a minimal cell structure involving 1 vertex, 2 edges and 1 cell of dimension 2. Minimal cell structures (in the four different categories) for the torus are illustrated as follows.

A second limitation to our representations of simplicial and regular CW-complexes is that they apply only to structures with finitely many cells. They do no apply, for instance, to the simplicial complex structure on the real line $\mathbb{R}$ in which each each integer n is a vertex and each interval $[n, n+1]$ is an edge.

Simplicial sets provide one approach to the efficient combinatorial representation of certain spaces. So too do cubical sets (the analogues of simplicial sets in which each cell has the combinatorics of an n-cube rather than an n-simplex). Neither of these two approaches has been implemented in HAP.

Simplicial sets endowed with the action of a (possibly infinite) group G provide for an efficient representation of (possibly infinite) cell structures on a wider class of spaces. Such a structure can be made precise and is known as a *simplicial group*. Some functionality for simplicial groups is implemented in HAP and described in Chapter 12.

A regular CW-complex endowed with the action of a (possibly infinite) group G is an alternative approach to the efficient combinatorial representation of (possibly infinite) cell structures on spaces. Much of HAP is focused on this approach. As a first example of the idea, the following commands construct the infinite regular CW-complex $Y = \widetilde{T}$ arising as the universal cover of the torus $T = \mathbb{S}^1 \times \mathbb{S}^1$ where T is given the above minimal non-regular CW structure involving 1 vertex, 2 edges, and 1 cell of dimension 2. The homology $H_n(T, \mathbb{Z})$ is computed and the fundamental group of the torus T is recovered.

Example

```
gap> F:=FreeGroup(2);;x:=F.1;;y:=F.2;;
gap> G:=F/[ x*y*x^-1*y^-1 ];;
gap> Y:=EquivariantTwoComplex(G);
Equivariant CW-complex of dimension 2

gap> C:=ChainComplexOfQuotient(Y);
Chain complex of length 2 in characteristic 0 .

gap> Homology(C,0);
[ 0 ]
gap> Homology(C,1);
[ 0, 0 ]
gap> Homology(C,2);
[ 0 ]
gap> FundamentalGroupOfQuotient(Y);
<fp group of size infinity on the generators [ f1, f2 ]>
```

As a second example, the following comands load group number 9 in the library of 3-dimensional crystallographic groups. They verify that G acts freely on $\mathbb{R}^3$ (i.e. G is a *Bieberbach group*) and then construct a G-equivariant CW-complex $Y = \mathbb{R}^3$ corresponding to the tessellation of $\mathbb{R}^3$ by a fundamental domain for G. Finally, the cohomology $H_n(M, \mathbb{Z})$ of the 3-dimensional closed manifold $M = \mathbb{R}^3/G$ is computed. The manifold M is seen to be non-orientable (since it's top-dimensional homology is trivial) and has a non-regular CW structure with 1 vertex, 3 edges, 3 cells of dimension 2, and 1 cell of dimension 3. (This example uses Polymake software.)

Example

```
gap> G:=SpaceGroup(3,9);;
gap> IsAlmostBieberbachGroup(Image(IsomorphismPcpGroup(G)));
true
gap> Y:=EquivariantEuclideanSpace(G,[0,0,0]);
Equivariant CW-complex of dimension 3
```

```
gap> Y!.dimension(0);
1
gap> Y!.dimension(1);
3
gap> Y!.dimension(2);
3
gap> Y!.dimension(3);
1
gap> C:=ChainComplexOfQuotient(Y);
Chain complex of length 3 in characteristic 0 .

gap> Homology(C,0);
[ 0 ]
gap> Homology(C,1);
[ 0, 0 ]
gap> Homology(C,2);
[ 2, 0 ]
gap> Homology(C,3);
[  ]
```

The fundamental domain for the action of G in the above example is constructed to be the Dirichlet-Voronoi region in $\mathbb{R}^3$ whose points are closer to the origin $v = (0,0,0)$ than to any other point v^g in the orbit of the origin under the action of G. This fundamental domain can be visualized as follows.

Example

```
gap> F:=FundamentalDomainStandardSpaceGroup([0,0,0],G);
<polymake object>
gap> Polymake(F,"VISUAL");
```

Other fundamental domains for the same group action can be obtained by choosing some other starting vector v. For example:

Example

```
gap> F:=FundamentalDomainStandardSpaceGroup([1/2,1/3,1/5],G);;
gap> Polymake(F,"VISUAL");

gap> F:=FundamentalDomainStandardSpaceGroup([1/7,1/2,1/2],G);
gap> Polymake(F,"VISUAL");
```

1.19 Orbifolds and classifying spaces

If a discrete group G acts on Euclidean space or hyperbolic space with finite stabilizer groups then we say that the quotient space obtained by killing the action of G an an *orbifold*. If the stabilizer groups are all trivial then the quotient is of course a manifold.

An orbifold is represented as a G-equivariant regular CW-complex together with the stabilizer group for a representative of each orbit of cells and its subgroup consisting of those group elements that preserve the cell orientation. HAP stores orbifolds using the data type of *non-free resolution* and uses them mainly as a first step in constructing free $\mathbb{Z}G$-resolutions of $\mathbb{Z}$.

The following commands use an 8-dimensional equivariant deformation retract of a $GL_3(\mathbb{Z}[\mathbf{i}])$-orbifold structure on hyperbolic space to compute $H_5(GL_3(\mathbb{Z}[\mathbf{i}],\mathbb{Z}) = \mathbb{Z}_2^5 \oplus \mathbb{Z}_4^2$. (The deformation retract is stored in a library and was supplied by Mathieu Dutour Sikiric.)

Example

```
gap> Orbifold:=ContractibleGcomplex("PGL(3,Z[i])");
Non-free resolution in characteristic 0 for matrix group .
No contracting homotopy available.

gap> R:=FreeGResolution(Orbifold,6);
Resolution of length 5 in characteristic 0 for matrix group .
No contracting homotopy available.

gap> Homology(TensorWithIntegers(R),5);
[ 2, 2, 2, 2, 2, 4, 4 ]
```

The next example computes an orbifold structure on $\mathbb{R}^4$, and then the first 12 degrees of a free resolution/classifying space, for the second 4-dimensional crystallographic group G in the library of crystallographic groups. The resolution is shown to be periodic of period 2 in degrees ≥ 5. The cohomology is seen to have 11 ring generators in degree 2 and no further ring generators. The cohomology groups are: $$H^n(G,\mathbb Z) =\left\{ \begin{array}{ll} 0, & {\rm odd~} n\ge 1\\ \mathbb Z_2^5 \oplus \mathbb Z^6, & n=2\\ \mathbb Z_2^{15}\oplus \mathbb Z, & n=4\\ \mathbb Z_2^{16}, & {\rm even~} n \ge 6 .\\ \end{array}\right.$$

Example

```
gap> G:=SpaceGroup(4,2);;
gap> R:=ResolutionCubicalCrystGroup(G,12);
Resolution of length 12 in characteristic 0 for <matrix group with
5 generators> .

gap> R!.dimension(5);
16
gap> R!.dimension(7);
16
gap> List([1..16],k->R!.boundary(5,k)=R!.boundary(7,k));
[ true, true, true, true, true, true, true, true, true, true, true, true,
  true, true, true, true ]

gap> C:=HomToIntegers(R);
Cochain complex of length 12 in characteristic 0 .

gap> Cohomology(C,0);
[ 0 ]
gap> Cohomology(C,1);
[  ]
gap> Cohomology(C,2);
[ 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0 ]
gap> Cohomology(C,3);
[  ]
gap> Cohomology(C,4);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0 ]
gap> Cohomology(C,5);
[  ]
```

```
gap> Cohomology(C,6);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
gap> Cohomology(C,7);
[  ]

gap> IntegralRingGenerators(R,1);
[  ]
gap> IntegralRingGenerators(R,2);
[ [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ], [ 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 ], [ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 ], [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ], [ 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 ], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0 ],
  [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1 ] ]
gap> IntegralRingGenerators(R,3);
[  ]
gap> IntegralRingGenerators(R,4);
[  ]
gap> IntegralRingGenerators(R,5);
[  ]
gap> IntegralRingGenerators(R,6);
[  ]
gap> IntegralRingGenerators(R,7);
[  ]
gap> IntegralRingGenerators(R,8);
[  ]
gap> IntegralRingGenerators(R,9);
[  ]
gap> IntegralRingGenerators(R,10);
[  ]
```

A group G with a finite index torsion free nilpotent subgroup admits a resolution which is periodic in sufficiently high degrees if and only if all of its finite index subgroups admit periodic resolutions. The following commands identify the 99 3-dimensional space groups (respectively, the 1191 4-dimensional space groups) that admit a resolution which is periodic in degrees > 3 (respectively, in degrees > 4).

Example

```
gap> L3:=Filtered([1..219],k->IsPeriodicSpaceGroup(SpaceGroup(3,k)));
[ 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 24, 26, 27, 28,
  29, 30, 31, 32, 33, 34, 36, 37, 39, 40, 41, 43, 45, 46, 52, 54, 55, 56, 58,
  61, 62, 74, 75, 76, 77, 78, 79, 80, 81, 84, 85, 87, 89, 92, 98, 101, 102,
  107, 111, 119, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151,
  152, 153, 154, 155, 157, 159, 161, 162, 163, 164, 165, 166, 168, 171, 172,
  174, 175, 176, 178, 180, 186, 189, 192, 196, 198, 209 ]

gap> L4:=Filtered([1..4783],k->IsPeriodicSpaceGroup(SpaceGroup(4,k)));
[ 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 22, 23, 25,
  26, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 46, 47, 48,
  49, 50, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 63, 65, 66, 68, 69, 70,
  71, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 89, 90, 91,
```

```
93, 94, 95, 96, 97, 98, 99, 101, 102, 103, 104, 105, 107, 108, 109, 110,
111, 113, 115, 116, 118, 119, 120, 121, 122, 124, 126, 127, 128, 130, 131,
134, 141, 144, 145, 149, 151, 153, 154, 155, 156, 157, 158, 159, 160, 162,
163, 165, 167, 168, 169, 170, 171, 172, 173, 174, 176, 178, 179, 180, 187,
188, 197, 198, 202, 204, 205, 206, 211, 212, 219, 220, 222, 226, 233, 237,
238, 239, 240, 241, 242, 243, 244, 245, 247, 248, 249, 250, 251, 253, 254,
255, 256, 257, 259, 260, 261, 263, 264, 265, 266, 267, 269, 270, 271, 273,
275, 277, 278, 279, 281, 283, 285, 290, 292, 296, 297, 298, 299, 300, 301,
303, 304, 305, 314, 316, 317, 319, 327, 328, 329, 333, 335, 342, 355, 357,
358, 359, 361, 362, 363, 365, 366, 367, 368, 369, 370, 372, 374, 376, 378,
381, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397,
398, 399, 400, 401, 402, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413,
414, 415, 416, 417, 418, 419, 421, 422, 423, 424, 425, 426, 427, 428, 429,
430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 442, 443, 444, 445,
446, 447, 448, 450, 451, 458, 459, 462, 464, 465, 466, 467, 469, 470, 473,
477, 478, 479, 482, 483, 484, 485, 486, 493, 495, 497, 501, 502, 503, 504,
505, 507, 508, 512, 514, 515, 516, 517, 522, 524, 525, 526, 527, 533, 537,
539, 540, 541, 542, 543, 544, 546, 548, 553, 555, 558, 562, 564, 565, 566,
567, 568, 571, 572, 573, 574, 576, 577, 580, 581, 582, 589, 590, 591, 593,
596, 598, 599, 612, 613, 622, 623, 624, 626, 632, 641, 647, 649, 651, 652,
654, 656, 657, 658, 659, 661, 662, 663, 665, 666, 667, 668, 669, 670, 671,
672, 674, 676, 677, 678, 679, 680, 682, 683, 684, 686, 688, 689, 690, 691,
692, 694, 696, 697, 698, 699, 700, 702, 708, 710, 712, 714, 716, 720, 722,
728, 734, 738, 739, 741, 742, 744, 745, 752, 754, 756, 757, 758, 762, 763,
769, 770, 778, 779, 784, 788, 790, 800, 801, 843, 845, 854, 855, 856, 857,
865, 874, 900, 904, 909, 911, 913, 915, 916, 917, 919, 920, 921, 922, 923,
924, 925, 926, 927, 929, 931, 932, 933, 934, 936, 938, 940, 941, 943, 945,
946, 953, 955, 956, 958, 963, 966, 972, 973, 978, 979, 981, 982, 983, 985,
987, 988, 989, 991, 992, 993, 995, 996, 998, 999, 1000, 1003, 1011, 1022,
1024, 1025, 1026, 1162, 1167, 1236, 1237, 1238, 1239, 1240, 1241, 1242,
1243, 1244, 1246, 1248, 1250, 1255, 1264, 1267, 1270, 1273, 1279, 1280,
1281, 1283, 1284, 1289, 1291, 1293, 1294, 1324, 1325, 1326, 1327, 1328,
1329, 1330, 1331, 1332, 1333, 1334, 1335, 1336, 1337, 1338, 1339, 1340,
1341, 1343, 1345, 1347, 1348, 1349, 1350, 1351, 1352, 1354, 1356, 1357,
1358, 1359, 1361, 1363, 1365, 1367, 1372, 1373, 1374, 1375, 1376, 1377,
1378, 1379, 1380, 1381, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389,
1390, 1393, 1395, 1397, 1399, 1400, 1401, 1404, 1405, 1408, 1410, 1419,
1420, 1421, 1422, 1424, 1425, 1426, 1428, 1429, 1438, 1440, 1441, 1442,
1443, 1444, 1445, 1449, 1450, 1451, 1456, 1457, 1460, 1461, 1462, 1464,
1465, 1470, 1472, 1473, 1477, 1480, 1481, 1487, 1488, 1489, 1493, 1494,
1495, 1501, 1503, 1506, 1509, 1512, 1515, 1518, 1521, 1524, 1527, 1530,
1532, 1533, 1534, 1537, 1538, 1541, 1542, 1544, 1547, 1550, 1552, 1553,
1554, 1558, 1565, 1566, 1568, 1573, 1644, 1648, 1673, 1674, 1700, 1702,
1705, 1713, 1714, 1735, 1738, 1740, 1741, 1742, 1743, 1744, 1745, 1746,
1747, 1748, 1749, 1750, 1751, 1752, 1753, 1754, 1755, 1756, 1757, 1759,
1761, 1762, 1763, 1765, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774,
1775, 1778, 1779, 1782, 1783, 1785, 1787, 1788, 1789, 1791, 1793, 1795,
1797, 1798, 1799, 1800, 1801, 1803, 1806, 1807, 1809, 1810, 1811, 1813,
1815, 1821, 1822, 1823, 1828, 1829, 1833, 1837, 1839, 1842, 1845, 1848,
1850, 1851, 1852, 1854, 1856, 1857, 1858, 1859, 1860, 1861, 1863, 1866,
1870, 1873, 1874, 1877, 1880, 1883, 1885, 1886, 1887, 1889, 1892, 1895,
1915, 1918, 1920, 1923, 1925, 1927, 1928, 1930, 1952, 1953, 1954, 1955,
```

```
2045, 2047, 2049, 2051, 2053, 2054, 2055, 2056, 2057, 2059, 2067, 2068,
2072, 2075, 2076, 2079, 2084, 2087, 2088, 2092, 2133, 2135, 2136, 2137,
2139, 2140, 2170, 2171, 2196, 2224, 2234, 2236, 2238, 2254, 2355, 2356,
2386, 2387, 2442, 2445, 2448, 2451, 2478, 2484, 2487, 2490, 2493, 2496,
2499, 2502, 2508, 2511, 2514, 2517, 2520, 2523, 2550, 2553, 2559, 2621,
2624, 2648, 2650, 3046, 3047, 3048, 3049, 3050, 3051, 3052, 3053, 3054,
3055, 3056, 3057, 3058, 3059, 3060, 3061, 3062, 3063, 3064, 3065, 3066,
3067, 3068, 3069, 3070, 3071, 3072, 3073, 3074, 3075, 3076, 3077, 3078,
3079, 3080, 3081, 3082, 3083, 3084, 3085, 3086, 3087, 3089, 3090, 3091,
3094, 3095, 3096, 3099, 3100, 3101, 3104, 3105, 3106, 3109, 3110, 3111,
3112, 3113, 3114, 3115, 3117, 3119, 3120, 3121, 3122, 3123, 3124, 3125,
3127, 3128, 3129, 3130, 3131, 3132, 3133, 3135, 3137, 3139, 3141, 3142,
3143, 3144, 3145, 3149, 3151, 3152, 3153, 3154, 3155, 3157, 3159, 3160,
3161, 3162, 3163, 3169, 3170, 3171, 3172, 3173, 3174, 3175, 3177, 3179,
3180, 3181, 3182, 3183, 3184, 3185, 3187, 3188, 3189, 3190, 3191, 3192,
3193, 3195, 3197, 3199, 3200, 3201, 3204, 3206, 3207, 3208, 3209, 3210,
3212, 3214, 3215, 3216, 3217, 3218, 3226, 3234, 3235, 3236, 3244, 3252,
3253, 3254, 3260, 3268, 3269, 3270, 3278, 3280, 3281, 3282, 3283, 3284,
3285, 3286, 3287, 3288, 3289, 3290, 3291, 3292, 3295, 3296, 3298, 3299,
3302, 3303, 3306, 3308, 3309, 3310, 3311, 3312, 3313, 3314, 3315, 3316,
3317, 3318, 3319, 3320, 3322, 3324, 3326, 3327, 3329, 3330, 3338, 3345,
3346, 3347, 3348, 3350, 3351, 3352, 3354, 3355, 3356, 3359, 3360, 3361,
3362, 3374, 3375, 3383, 3385, 3398, 3399, 3417, 3418, 3419, 3420, 3422,
3424, 3426, 3428, 3446, 3447, 3455, 3457, 3469, 3471, 3521, 3523, 3524,
3525, 3530, 3531, 3534, 3539, 3542, 3545, 3548, 3550, 3551, 3554, 3557,
3579, 3580, 3830, 3831, 3832, 3833, 3835, 3837, 3839, 3849, 3851, 3877,
3938, 3939, 3949, 3951, 3952, 3958, 3960, 3962, 3963, 3964, 3966, 3968,
3972, 3973, 3975, 4006, 4029, 4030, 4033, 4034, 4037, 4038, 4046, 4048,
4050, 4062, 4064, 4067, 4078, 4081, 4089, 4090, 4114, 4138, 4139, 4140,
4141, 4146, 4147, 4148, 4149, 4154, 4155, 4169, 4171, 4175, 4180, 4183,
4188, 4190, 4204, 4205, 4223, 4224, 4225, 4254, 4286, 4289, 4391, 4397,
4496, 4499, 4500, 4501, 4502, 4504, 4508, 4510, 4521, 4525, 4544, 4559,
4560, 4561, 4562, 4579, 4580, 4581, 4583, 4587, 4597, 4598, 4599, 4600,
4651, 4759, 4760, 4761, 4762, 4766 ]
```

Chapter 2

Cubical complexes & permutahedral complexes

2.1 Cubical complexes

A *finite simplicial complex* can be defined to be a CW-subcomplex of the canonical regular CW-structure on a simplex Δ^n of some dimension n. Analogously, a *finite cubical complex* is a CW-subcomplex of the regular CW-structure on a cube $[0,1]^n$ of some dimension n. Equivalently, but more conveniently, we can replace the unit interval $[0,1]$ by an interval $[0,k]$ with CW-structure involving $2k+1$ cells, namely one 0-cell for each integer $0 \leq j \leq k$ and one 1-cell for each open interval $(j, j+1)$ for $0 \leq j \leq k-1$. A *finite cuical complex* M is a CW-subcompex $M \subset [0,k_1] \times [0,k_2] \times \cdots [0,k_n]$ of a direct product of intervals, the direct product having the usual direct product CW-structure. The equivalence of these two definitions follows from the Gray code embedding of a mesh into a hypercube. We say that the cubical complex has *ambient dimension* n. A cubical complex M of ambient dimension n is said to be *pure* if each cell lies in the boundary of an n-cell. In other words, M is pure if it is a union of unit n-cubes in $\mathbb{R}^n$, each unit cube having vertices with integer coordinates.

HAP has a datatype for finite cubical complexes, and a slightly different datatype for pure cubical complexes.

The following example constructs the granny knot (the sum of a trefoil knot with its reflection) as a 3-dimensional pure cubical complex, and then displays it.

Example

```
gap> K:=PureCubicalKnot(3,1);
prime knot 1 with 3 crossings

gap> L:=ReflectedCubicalKnot(K);
Reflected( prime knot 1 with 3 crossings )

gap> M:=KnotSum(K,L);
prime knot 1 with 3 crossings + Reflected( prime knot 1 with 3 crossings )

gap> Display(M);
```

Next we construct the complement $Y = D^3 \setminus \mathring{M}$ of the interior of the pure cubical complex M. Here D^3 is a rectangular region with $M \subset \mathring{D}^3$. This pure cubical complex Y is a union of 5891 unit

3-cubes. We contract Y to get a homotopy equivalent pure cubical complex YY consisting of the union of just 775 unit 3-cubes. Then we convert YY to a regular CW-complex W involving 11939 cells. We contract W to obtain a homotopy equivalent regular CW-complex WW involving 5993 cells. Finally we compute the fundamental group of the complement of the granny knot, and use the presentation of this group to establish that the Alexander polynomial $P(x)$ of the granny is

$$P(x) = x^4 - 2x^3 + 3x^2 - 2x + 1 \ .$$

Example

```
gap> Y:=PureComplexComplement(M);
Pure cubical complex of dimension 3.

gap> Size(Y);
5891

gap> YY:=ZigZagContractedComplex(Y);
Pure cubical complex of dimension 3.

gap> Size(YY);
775

gap> W:=RegularCWComplex(YY);
Regular CW-complex of dimension 3

gap> Size(W);
11939

gap> WW:=ContractedComplex(W);
Regular CW-complex of dimension 2

gap> Size(WW);
5993

gap> G:=FundamentalGroup(WW);
<fp group of size infinity on the generators [ f1, f2, f3 ]>

gap> AlexanderPolynomial(G);
x_1^4-2*x_1^3+3*x_1^2-2*x_1+1
```

2.2 Permutahedral complexes

A finite pure cubical complex is a union of finitely many cubes in a tessellation of $\mathbb{R}^n$ by unit cubes. One can also tessellate $\mathbb{R}^n$ by permutahedra, and we define a finite n-dimensional pure *permutahedral complex* to be a union of finitely many permutahdra from such a tessellation. There are two features of pure permutahedral complexes that are particularly useful in some situations:

- Pure permutahedral complexes are topological manifolds with boundary.
- The method used for finding a smaller pure cubical complex M' homotopy equivalent to a given pure cubical complex M retains the homeomorphism type, and not just the homotopy type, of the space M.

EXAMPLE 1

To illustrate these features the following example begins by reading in a protein backbone from the online Protein Database, and storing it as a pure cubical complex K. The ends of the protein have been joined, and the homology $H_i(K,\mathbb{Z}) = \mathbb{Z}$, $i = 0,1$ is seen to be that of a circle. We can thus regard the protein as a knot $K \subset \mathbb{R}^3$. The protein is visualized as a pure permutahedral complex.

Example

```
gap> file:=HapFile("data1V2X.pdb");;
gap> K:=ReadPDBfileAsPurePermutahedralComplex("file");
Pure permutahedral complex of dimension 3.

gap> Homology(K,0);
[ 0 ]
gap> Homology(K,1);
[ 0 ]

Display(K);
```

An alternative method for seeing that the pure permutahedral complex K has the homotopy type of a circle is to note that it is covered by open permutahedra (small open neighbourhoods of the closed 3-dimensional permutahedral titles) and to form the nerve $N = Nerve(\mathscr{U})$ of this open covering $\mathscr{U}$. The nerve N has the same homotopy type as K. The following commands establish that N is a 1-dimensional simplicial complex and display N as a circular graph.

Example

```
gap> N:=Nerve(K);
Simplicial complex of dimension 1.

gap> Display(GraphOfSimplicialComplex(N));
```

The boundary of the pure permutahedral complex K is a 2-dimensional CW-complex B homeomorphic to a torus. We next use the advantageous features of pure permutahedral complexes to compute the homomorphism

$\phi: \pi_1(B) \to \pi_1(\mathbb{R}^3 \setminus \mathring{K}), a \mapsto yx^{-3}y^2x^{-2}yxy^{-1}, b \mapsto yx^{-1}y^{-1}x^2y^{-1}$

where

$\pi_1(B) =< a,b : aba^{-1}b^{-1} = 1 >,$

$\pi_1(\mathbb{R}^3 \setminus \mathring{K}) \cong < x,y : y^2x^{-2}yxy^{-1} = 1, yx^{-2}y^{-1}x(xy^{-1})^2 = 1 >.$

Example

```
gap> Y:=PureComplexComplement(K);
Pure permutahedral complex of dimension 3.
gap> Size(Y);
418922

gap> YY:=ZigZagContractedComplex(Y);
Pure permutahedral complex of dimension 3.
gap> Size(YY);
3438

gap> W:=RegularCWComplex(YY);
Regular CW-complex of dimension 3
```

```
gap> f:=BoundaryMap(W);
Map of regular CW-complexes

gap> CriticalCells(Source(f));
[ [ 2, 1 ], [ 2, 261 ], [ 1, 1043 ], [ 1, 1626 ], [ 0, 2892 ], [ 0, 24715 ] ]

gap> F:=FundamentalGroup(f,2892);
[ f1, f2 ] -> [ f2*f1^-3*f2^2*f1^-2*f2*f1*f2^-1, f2*f1^-1*f2^-1*f1^2*f2^-1 ]

gap> G:=Target(F);
<fp group on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(G);
[ f2^2*f1^-2*f2*f1*f2^-1, f2*f1^-2*f2^-1*f1*(f1*f2^-1)^2 ]
```

EXAMPLE 2

The next example of commands begins by readng two synthetic knots from a CSV file (containing the coordinates of the two sequences of vertices) and producing a pure permutahedral complex model of the two knots. The linking number of two knots is given by the low-dimension cup product of the complement of the knots. This linking number is computed to be 6.

Example

```
gap> file1:=HapFile("data175_1.csv");;
gap> file2:=HapFile("data175_2.csv");;
gap> K:=ReadCSVfileAsPureCubicalKnot( [file1, file2]);;
gap> K:=PurePermutahedralComplex(K!.binaryArray);;
gap> K:=ThickenedPureComplex(K);;
gap> K:=ContractedComplex(K);;
gap> #K is a permutahedral complex model of the two input knots
gap> Display(K);


gap> Y:=PureComplexComplement(K);;
gap> W:=ZigZagContractedComplex(Y,2);;
gap> W:=RegularCWComplex(W);;
gap> W:=ContractedComplex(W);;
gap> G:=FundamentalGroup(W);;
gap> cup:=CupProduct(G);;
gap> cup([1,0],[0,1]);
[ -6, 0 ]
```

2.3 Constructing pure cubical and permutahedral complexes

An n-dimensional pure cubical or permutahedral complex can be created from an n-dimensional array of 0s and 1s. The following example creates and displays two 3-dimensional complexes.

Example

```
gap> A:=[[[0,0,0],[0,0,0],[0,0,0]],
>        [[1,1,1],[1,0,1],[1,1,1]],
```

```
>        [[0,0,0],[0,0,0],[0,0,0]]];;
gap> M:=PureCubicalComplex(A);
Pure cubical complex of dimension 3.

gap> P:=PurePermutahedralComplex(A);
Pure permutahedral complex of dimension 3.

gap> Display(M);
gap> Display(P);
```

2.4 Computations in dynamical systems

Pure cubical complexes can be useful for rigourous interval arithmetic calculations in numerical analysis. They can also be useful for trying to estimate approximations of certain numerical quantities. To illustrate the latter we consider the *Henon map*

$$f:\mathbb{R}^2 \to \mathbb{R}^2, \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} y+1-ax^2 \\ bx \end{pmatrix}.$$

Starting with $(x_0, y_0) = (0,0)$ and iterating $(x_{n+1}, y_{n+1}) = f(x_n, y_n)$ with the parameter values $a = 1.4$, $b = 0.3$ one obtains a sequence of points which is known to be dense in the so called *strange attractor* $\mathscr{A}$ of the Henon map. The first 10 million points in this sequence are plotted in the following example, with arithmetic performed to 100 decimal places of accuracy. The sequence is stored as a 2-dimensional pure cubical complex where each 2-cell is square of side equal to $\varepsilon = 1/500$.

Example

```
gap> M:=HenonOrbit([0,0],14/10,3/10,10^7,500,100);
Pure cubical complex of dimension 2.

gap> Size(M);
10287

gap> Display(M);
```

Repeating the computation but with squares of side $\varepsilon = 1/1000$

Example

```
gap> M:=HenonOrbit([0,0],14/10,3/10,10^7,1000,100);

gap> Size(M);
24949
```

we obtain the heuristic estimate
$\delta \simeq \frac{\log 24949 - \log 10287}{\log 2} = 1.277$
for the box-counting dimension of the attractor $\mathscr{A}$.

Chapter 3

Covering spaces

Let Y denote a finite regular CW-complex. Let $\widetilde{Y}$ denote its universal covering space. The covering space inherits a regular CW-structure which can be computed and stored using the datatype of a $\pi_1 Y$-equivariant CW-complex. The cellular chain complex $C_*\widetilde{Y}$ of $\widetilde{Y}$ can be computed and stored as an equivariant chain complex. Given an admissible discrete vector field on Y, we can endow Y with a smaller non-regular CW-structre whose cells correspond to the critical cells in the vector field. This smaller CW-structure leads to a more efficient chain complex $C_*\widetilde{Y}$ involving one free generator for each critical cell in the vector field.

3.1 Cellular chains on the universal cover

The following commands construct a 6-dimensional regular CW-complex $Y \simeq S^1 \times S^1 \times S^1$ homotopy equivalent to a product of three circles.

Example

```
gap> A:=[[1,1,1],[1,0,1],[1,1,1]];;
gap> S:=PureCubicalComplex(A);;
gap> T:=DirectProduct(S,S,S);;
gap> Y:=RegularCWComplex(T);;
Regular CW-complex of dimension 6

gap> Size(Y);
110592
```

The CW-somplex Y has 110592 cells. The next commands construct a free $\pi_1 Y$-equivariant chain complex $C_*\widetilde{Y}$ homotopy equivalent to the chain complex of the universal cover of Y. The chain complex $C_*\widetilde{Y}$ has just 8 free generators.

Example

```
gap> Y:=ContractedComplex(Y);;
gap> CU:=ChainComplexOfUniversalCover(Y);;
gap> List([0..Dimension(Y)],n->CU!.dimension(n));
[ 1, 3, 3, 1 ]
```

The next commands construct a subgroup $H < \pi_1 Y$ of index 50 and the chain complex $C_*\widetilde{Y} \otimes_{\mathbb{Z}H} \mathbb{Z}$ which is homotopy equivalent to the cellular chain complex $C_*\widetilde{Y}_H$ of the 50-fold cover $\widetilde{Y}_H$ of Y corresponding to H.

Example

```
gap> L:=LowIndexSubgroupsFpGroup(CU!.group,50);;
gap> H:=L[Length(L)-1];;
gap> Index(CU!.group,H);
50
gap> D:=TensorWithIntegersOverSubgroup(CU,H);
Chain complex of length 3 in characteristic 0 .

gap> List([0..3],D!.dimension);
[ 50, 150, 150, 50 ]
```

General theory implies that the 50-fold covering space $\widetilde{Y}_H$ should again be homotopy equivalent to a product of three circles. In keeping with this, the following commands verify that $\widetilde{Y}_H$ has the same integral homology as $S^1 \times S^1 \times S^1$.

Example

```
gap> Homology(D,0);
[ 0 ]
gap> Homology(D,1);
[ 0, 0, 0 ]
gap> Homology(D,2);
[ 0, 0, 0 ]
gap> Homology(D,3);
[ 0 ]
```

3.2 Spun knots and the Satoh tube map

We'll contruct two spaces Y, W with isomorphic fundamental groups and isomorphic intergal homology, and use the integral homology of finite covering spaces to establsh that the two spaces have distinct homotopy types.

By *spinning* a link $K \subset \mathbb{R}^3$ about a plane $P \subset \mathbb{R}^3$ with $P \cap K = \emptyset$, we obtain a collection $Sp(K) \subset \mathbb{R}^4$ of knotted tori. The following commands produce the two tori obtained by spinning the Hopf link K and show that the space $Y = \mathbb{R}^4 \setminus Sp(K) = Sp(\mathbb{R}^3 \setminus K)$ is connected with fundamental group $\pi_1 Y = \mathbb{Z} \times \mathbb{Z}$ and homology groups $H_0(Y) = \mathbb{Z}$, $H_1(Y) = \mathbb{Z}^2$, $H_2(Y) = \mathbb{Z}^4$, $H_3(Y, \mathbb{Z}) = \mathbb{Z}^2$. The space Y is only constructed up to homotopy, and for this reason is 3-dimensional.

Example

```
gap> Hopf:=PureCubicalLink("Hopf");
Pure cubical link.

gap> Y:=SpunAboutInitialHyperplane(PureComplexComplement(Hopf));
Regular CW-complex of dimension 3

gap> Homology(Y,0);
[ 0 ]
gap> Homology(Y,1);
[ 0, 0 ]
gap> Homology(Y,2);
[ 0, 0, 0, 0 ]
gap> Homology(Y,3);
```

```
[ 0, 0 ]
gap> Homology(Y,4);
[  ]
gap> GY:=FundamentalGroup(Y);;
gap> GeneratorsOfGroup(GY);
[ f2, f3 ]
gap> RelatorsOfFpGroup(GY);
[ f3^-1*f2^-1*f3*f2 ]
```

An alternative embedding of two tori $L \subset \mathbb{R}^4$ can be obtained by applying the 'tube map' of Shin Satoh to a welded Hopf link [Sat00]. The following commands construct the complement $W = \mathbb{R}^4 \setminus L$ of this alternative embedding and show that W has the same fundamental group and integral homology as Y above.

Example

```
gap> L:=HopfSatohSurface();
Pure cubical complex of dimension 4.

gap> W:=ContractedComplex(RegularCWComplex(PureComplexComplement(L)));
Regular CW-complex of dimension 3

gap> Homology(W,0);
[ 0 ]
gap> Homology(W,1);
[ 0, 0 ]
gap> Homology(W,2);
[ 0, 0, 0, 0 ]
gap> Homology(W,3);
[ 0, 0 ]
gap> Homology(W,4);
[  ]

gap> GW:=FundamentalGroup(W);;
gap> GeneratorsOfGroup(GW);
[ f1, f2 ]
gap> RelatorsOfFpGroup(GW);
[ f1^-1*f2^-1*f1*f2 ]
```

Despite having the same fundamental group and integral homology groups, the above two spaces Y and W were shown by Kauffman and Martins [KFM08] to be not homotopy equivalent. Their technique involves the fundamental crossed module derived from the first three dimensions of the universal cover of a space, and counts the representations of this fundamental crossed module into a given finite crossed module. This homotopy inequivalence is recovered by the following commands which involves the 5-fold covers of the spaces.

Example

```
gap> CY:=ChainComplexOfUniversalCover(Y);
Equivariant chain complex of dimension 3
gap> LY:=LowIndexSubgroups(CY!.group,5);;
gap> invY:=List(LY,g->Homology(TensorWithIntegersOverSubgroup(CY,g),2));;
```

```
gap> CW:=ChainComplexOfUniversalCover(W);
Equivariant chain complex of dimension 3
gap> LW:=LowIndexSubgroups(CW!.group,5);;
gap> invW:=List(LW,g->Homology(TensorWithIntegersOverSubgroup(CW,g),2));;

gap> SSortedList(invY)=SSortedList(invW);
false
```

3.3 Cohomology with local coefficients

The $\pi_1 Y$-equivariant cellular chain complex $C_*\widetilde{Y}$ of the universal cover $\widetilde{Y}$ of a regular CW-complex Y can be used to compute the homology $H_n(Y,A)$ and cohomology $H^n(Y,A)$ of Y with local coefficients in a $\mathbb{Z}\pi_1 Y$-module A. To illustrate this we consister the space Y arising as the complement of the trefoil knot, with fundamental group $\pi_1 Y = \langle x,y : xyx = yxy\rangle$. We take $A = \mathbb{Z}$ to be the integers with non-trivial $\pi_1 Y$-action given by $x.1 = -1, y.1 = -1$. We then compute

$$\begin{aligned} H_0(Y,A) &= \mathbb{Z}_2\,, \\ H_1(Y,A) &= \mathbb{Z}_3\,, \\ H_2(Y,A) &= \mathbb{Z}\,. \end{aligned}$$

Example

```
gap> K:=PureCubicalKnot(3,1);;
gap> Y:=PureComplexComplement(K);;
gap> Y:=ContractedComplex(Y);;
gap> Y:=RegularCWComplex(Y);;
gap> Y:=SimplifiedComplex(Y);;
gap> C:=ChainComplexOfUniversalCover(Y);;
gap> G:=C!.group;;
gap> GeneratorsOfGroup(G);
[ f1, f2 ]
gap> RelatorsOfFpGroup(G);
[ f2^-1*f1^-1*f2^-1*f1*f2*f1, f1^-1*f2^-1*f1^-1*f2*f1*f2 ]
gap> hom:=GroupHomomorphismByImages(G,Group([[-1]]),[G.1,G.2],[[[-1]],[[-1]]]);;
gap> A:=function(x); return Determinant(Image(hom,x)); end;;
gap> D:=TensorWithTwistedIntegers(C,A); #Here the function A represents
gap> #the integers with twisted action of G.
Chain complex of length 3 in characteristic 0 .
gap> Homology(D,0);
[ 2 ]
gap> Homology(D,1);
[ 3 ]
gap> Homology(D,2);
[ 0 ]
```

3.4 Distinguishing between two non-homeomorphic homotopy equivalent spaces

The granny knot is the sum of the trefoil knot and its mirror image. The reef knot is the sum of two identical copies of the trefoil knot. The following commands show that the degree 1 homology homomorphisms

$H_1(p^{-1}(B),\mathbb{Z}) \to H_1(\widetilde{X}_H,\mathbb{Z})$

distinguish between the homeomorphism types of the complements $X \subset \mathbb{R}^3$ of the granny knot and the reef knot, where $B \subset X$ is the knot boundary, and where $p{:}\widetilde{X}_H \to X$ is the covering map corresponding to the finite index subgroup $H < \pi_1 X$. More precisely, $p^{-1}(B)$ is in general a union of path components

$p^{-1}(B) = B_1 \cup B_2 \cup \cdots \cup B_t$.

The function `FirstHomologyCoveringCokernels(f,c)` inputs an integer c and the inclusion $f{:}B \hookrightarrow X$ of a knot boundary B into the knot complement X. The function returns the ordered list of the lists of abelian invariants of cokernels

coker($H_1(p^{-1}(B_i),\mathbb{Z}) \to H_1(\widetilde{X}_H,\mathbb{Z})$)

arising from subgroups $H < \pi_1 X$ of index c. To distinguish between the granny and reef knots we use index $c = 6$.

Example

```
gap> K:=PureCubicalKnot(3,1);;
gap> L:=ReflectedCubicalKnot(K);;
gap> granny:=KnotSum(K,L);;
gap> reef:=KnotSum(K,K);;
gap> fg:=KnotComplementWithBoundary(ArcPresentation(granny));;
gap> fr:=KnotComplementWithBoundary(ArcPresentation(reef));;
gap> a:=FirstHomologyCoveringCokernels(fg,6);;
gap> b:=FirstHomologyCoveringCokernels(fr,6);;
gap> a=b;
false
```

3.5 Second homotopy groups of spaces with finite fundamental group

If $p : \widetilde{Y} \to Y$ is the universal covering map, then the fundamental group of $\widetilde{Y}$ is trivial and the Hurewicz homomorphism $\pi_2\widetilde{Y} \to H_2(\widetilde{Y},\mathbb{Z})$ from the second homotopy group of $\widetilde{Y}$ to the second integral homology of $\widetilde{Y}$ is an isomorphism. Furthermore, the map p induces an isomorphism $\pi_2\widetilde{Y} \to \pi_2 Y$. Thus $H_2(\widetilde{Y},\mathbb{Z})$ is isomorphic to the second homotopy group $\pi_2 Y$.

If the fundamental group of Y happens to be finite, then in principle we can calculate $H_2(\widetilde{Y},\mathbb{Z}) \cong \pi_2 Y$. We illustrate this computation for Y equal to the real projective plane. The above computation shows that Y has second homotopy group $\pi_2 Y \cong \mathbb{Z}$.

Example

```
gap> K:=[ [1,2,3], [1,3,4], [1,2,6], [1,5,6], [1,4,5],
>         [2,3,5], [2,4,5], [2,4,6], [3,4,6], [3,5,6]];;

gap> K:=MaximalSimplicesToSimplicialComplex(K);
Simplicial complex of dimension 2.

gap> Y:=RegularCWComplex(K);
```

```
Regular CW-complex of dimension 2
gap> # Y is a regular CW-complex corresponding to the projective plane.

gap> U:=UniversalCover(Y);
Equivariant CW-complex of dimension 2

gap> G:=U!.group;;
gap> # G is the fundamental group of Y, which by the next command
gap> # is finite of order 2.
gap> Order(G);
2

gap> U:=EquivariantCWComplexToRegularCWComplex(U,Group(One(G)));
Regular CW-complex of dimension 2
gap> #U is the universal cover of Y

gap> Homology(U,0);
[ 0 ]
gap> Homology(U,1);
[  ]
gap> Homology(U,2);
[ 0 ]
```

3.6 Third homotopy groups of simply connected spaces

3.6.1 First example: Whitehead's certain exact sequence

For any path connected space Y with universal cover $\widetilde{Y}$ there is an exact sequence

$\to \pi_4\widetilde{Y} \to H_4(\widetilde{Y},\mathbb{Z}) \to H_4(K(\pi_2\widetilde{Y},2),\mathbb{Z}) \to \pi_3\widetilde{Y} \to H_3(\widetilde{Y},\mathbb{Z}) \to 0$

due to J.H.C.Whitehead. Here $K(\pi_2(\widetilde{Y}),2)$ is an Eilenberg-MacLane space with second homotopy group equal to $\pi_2\widetilde{Y}$.

Continuing with the above example where Y is the real projective plane, we see that $H_4(\widetilde{Y},\mathbb{Z}) = H_3(\widetilde{Y},\mathbb{Z}) = 0$ since $\widetilde{Y}$ is a 2-dimensional CW-space. The exact sequence implies $\pi_3\widetilde{Y} \cong H_4(K(\pi_2\widetilde{Y},2),\mathbb{Z})$. Furthermore, $\pi_3\widetilde{Y} = \pi_3 Y$. The following commands establish that $\pi_3 Y \cong \mathbb{Z}$.

Example

```
gap> A:=AbelianPcpGroup([0]);
Pcp-group with orders [ 0 ]

gap> K:=EilenbergMacLaneSimplicialGroup(A,2,5);;
gap> C:=ChainComplexOfSimplicialGroup(K);
Chain complex of length 5 in characteristic 0 .

gap> Homology(C,4);
[ 0 ]
```

3.6.2 Second example: the Hopf invariant

The following commands construct a 4-dimensional simplicial complex Y with 9 vertices and 36 4-dimensional simplices, and establish that

$\pi_1 Y = 0, \pi_2 Y = \mathbb{Z}, H_3(Y,\mathbb{Z}) = 0, H_4(Y,\mathbb{Z}) = \mathbb{Z}$.

Example

```
gap> smp:=[ [ 1, 2, 4, 5, 6 ], [ 1, 2, 4, 5, 9 ], [ 1, 2, 5, 6, 8 ],
>         [ 1, 2, 6, 4, 7 ], [ 2, 3, 4, 5, 8 ], [ 2, 3, 5, 6, 4 ],
>         [ 2, 3, 5, 6, 7 ], [ 2, 3, 6, 4, 9 ], [ 3, 1, 4, 5, 7 ],
>         [ 3, 1, 5, 6, 9 ], [ 3, 1, 6, 4, 5 ], [ 3, 1, 6, 4, 8 ],
>         [ 4, 5, 7, 8, 3 ], [ 4, 5, 7, 8, 9 ], [ 4, 5, 8, 9, 2 ],
>         [ 4, 5, 9, 7, 1 ], [ 5, 6, 7, 8, 2 ], [ 5, 6, 8, 9, 1 ],
>         [ 5, 6, 8, 9, 7 ], [ 5, 6, 9, 7, 3 ], [ 6, 4, 7, 8, 1 ],
>         [ 6, 4, 8, 9, 3 ], [ 6, 4, 9, 7, 2 ], [ 6, 4, 9, 7, 8 ],
>         [ 7, 8, 1, 2, 3 ], [ 7, 8, 1, 2, 6 ], [ 7, 8, 2, 3, 5 ],
>         [ 7, 8, 3, 1, 4 ], [ 8, 9, 1, 2, 5 ], [ 8, 9, 2, 3, 1 ],
>         [ 8, 9, 2, 3, 4 ], [ 8, 9, 3, 1, 6 ], [ 9, 7, 1, 2, 4 ],
>         [ 9, 7, 2, 3, 6 ], [ 9, 7, 3, 1, 2 ], [ 9, 7, 3, 1, 5 ] ];;

gap> K:=MaximalSimplicesToSimplicialComplex(smp);
Simplicial complex of dimension 4.

gap> Y:=RegularCWComplex(Y);
Regular CW-complex of dimension 4

gap> Order(FundamentalGroup(Y));
1
gap> Homology(Y,2);
[ 0 ]
gap> Homology(Y,3);
[  ]
gap> Homology(Y,4);
[ 0 ]
```

Previous commands have established $H_4(K(\mathbb{Z},2),\mathbb{Z}) = \mathbb{Z}$. So Whitehead's sequence reduces to an exact sequence

$\mathbb{Z} \to \mathbb{Z} \to \pi_3 Y \to 0$

in which the first map is $H_4(Y,\mathbb{Z}) = \mathbb{Z} \to H_4(K(\pi_2 Y,2),\mathbb{Z}) = \mathbb{Z}$. Hence $\pi_3 Y$ is cyclic.

HAP is currently unable to compute the order of $\pi_3 Y$ directly from Whitehead's sequence. Instead, we can use the *Hopf invariant*. For any map $\phi: S^3 \to S^2$ we consider the space $C(\phi) = S^2 \cup_\phi e^4$ obtained by attaching a 4-cell e^4 to S^2 via the attaching map ϕ. The cohomology groups $H^2(C(\phi),\mathbb{Z}) = \mathbb{Z}$, $H^4(C(\phi),\mathbb{Z}) = \mathbb{Z}$ are generated by elements α, β say, and the cup product has the form

$-\cup -: H^2(C(\phi),\mathbb{Z}) \times H^2(C(\phi),\mathbb{Z}) \to H^4(C(\phi),\mathbb{Z}), (\alpha,\alpha) \mapsto h_\phi \beta$

for some integer h_ϕ. The integer h_ϕ is the HOPF INVARIANT. The function $h: \pi_3(S^3) \to \mathbb{Z}$ is a homomorphism and there is an isomorphism

$\pi_3(S^2 \cup e^4) \cong \mathbb{Z}/\langle h_\phi \rangle$.

The following commands begin by simplifying the cell structure on the above CW-complex $Y \cong K$ to obtain a homeomorphic CW-complex W with fewer cells. They then create a space S by removing one 4-cell from W. The space S is seen to be homotopy equivalent to a CW-complex $e^2 \cup e^0$ with a

single 0-cell and single 2-cell. Hence $S \simeq S^2$ is homotopy equivalent to the 2-sphere. Consequently $Y \simeq C(\phi) = S^2 \cup_\phi e^4$ for some map $\phi: S^3 \to S^2$.

Example

```
gap> W:=SimplifiedComplex(Y);
Regular CW-complex of dimension 4

gap> S:=RegularCWComplexWithRemovedCell(W,4,6);
Regular CW-complex of dimension 4

gap> CriticalCells(S);
[ [ 2, 6 ], [ 0, 5 ] ]
```

The next commands show that the map ϕ in the construction $Y \simeq C(\phi)$ has Hopf invariant -1. Hence $h: \pi_3(S^3) \to \mathbb{Z}$ is an isomorphism. Therefore $\pi_3 Y = 0$.

Example

```
gap> IntersectionForm(K);
[ [ -1 ] ]
```

[The simplicial complex K in this second example is due to W. Kuehnel and T. F. Banchoff and is homeomorphic to the complex projective plane.]

3.7 Computing the second homotopy group of a space with infinite fundamental group

The following commands compute the second integral homology

$H_2(\pi_1 W, \mathbb{Z}) = \mathbb{Z}$

of the fundamental group $\pi_1 W$ of the complement W of the Hopf-Satoh surface.

Example

```
gap> L:=HopfSatohSurface();
Pure cubical complex of dimension 4.

gap> W:=ContractedComplex(RegularCWComplex(PureComplexComplement(L)));
Regular CW-complex of dimension 3

gap> GW:=FundamentalGroup(W);;
gap> IsAspherical(GW);
Presentation is aspherical.
true
gap> R:=ResolutionAsphericalPresentation(GW);;
gap> Homology(TensorWithIntegers(R),2);
[ 0 ]
```

From Hopf's exact sequence

$\pi_2 W \xrightarrow{h} H_2(W, \mathbb{Z}) \twoheadrightarrow H_2(\pi_1 W, \mathbb{Z}) \to 0$

and the computation $H_2(W, \mathbb{Z}) = \mathbb{Z}^4$ we see that the image of the Hurewicz homomorphism is $\mathrm{im}(h) = \mathbb{Z}^3$. The image of h is referred to as the subgroup of *spherical homology classes* and often denoted by $\Sigma^2 W$.

The following command computes the presentation of $\pi_1 W$ corresponding to the 2-skeleton W^2 and establishes that $W^2 = S^2 \vee S^2 \vee S^2 \vee (S^1 \times S^1)$ is a wedge of three spheres and a torus.

Example

```
gap> F:=FundamentalGroupOfRegularCWComplex(W,"no simplification");
< fp group on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(F);
[ < identity ...>, f1^-1*f2^-1*f1*f2, < identity ...>, <identity ...> ]
```

The next command shows that the 3-dimensional space W has two 3-cells each of which is attached to the base-point of W with trivial boundary (up to homotopy in W^2). Hence $W = S^3 \vee S^3 \vee S^2 \vee S^2 \vee S^2 \vee (S^1 \times S^1)$.

Example

```
gap> CriticalCells(W);
[ [ 3, 1 ], [ 3, 3148 ], [ 2, 6746 ], [ 2, 20510 ], [ 2, 33060 ],
  [ 2, 50919 ], [ 1, 29368 ], [ 1, 50822 ], [ 0, 21131 ] ]
gap> CriticalBoundaryCells(W,3,1);
[  ]
gap> CriticalBoundaryCells(W,3,3148);
[ -50919, 50919 ]
```

Therefore $\pi_1 W$ is the free abelian group on two generators, and $\pi_2 W$ is the free $\mathbb{Z}\pi_1 W$-module on three free generators.

Chapter 4

Three Manifolds

4.1 Dehn Surgery

The following example constructs, as a regular CW-complex, a closed orientable 3-manifold W obtained from the 3-sphere by drilling out a tubular neighbourhood of a trefoil knot and then gluing a solid torus to the boundary of the cavity via a homeomorphism corresponding to a Dehn surgery coefficient $p/q = 17/16$.

Example

```
gap> ap:=ArcPresentation(PureCubicalKnot(3,1));;
gap> p:=17;;q:=16;;
gap> W:=ThreeManifoldViaDehnSurgery(ap,p,q);
Regular CW-complex of dimension 3
```

The next commands show that this 3-manifold W has integral homology
$H_0(W,\mathbb{Z}) = \mathbb{Z}$, $H_1(W,\mathbb{Z}) = \mathbb{Z}_{16}$, $H_2(W,\mathbb{Z}) = 0$, $H_3(W,\mathbb{Z}) = \mathbb{Z}$
and that the fundamental group $\pi_1(W)$ is non-abelian.

Example

```
gap> Homology(W,0);Homology(W,1);Homology(W,2);Homology(W,3);
[ 0 ]
[ 16 ]
[  ]
[ 0 ]

gap> F:=FundamentalGroup(W);;
gap> L:=LowIndexSubgroupsFpGroup(F,10);;
gap> List(L,AbelianInvariants);
[ [ 16 ], [ 3, 8 ], [ 3, 4 ], [ 2, 3 ], [ 16, 43 ], [ 8, 43, 43 ] ]
```

The following famous result of Lickorish and (independently) Wallace shows that Dehn surgery on knots leads to an interesting range of spaces.

THEOREM: *Every closed, orientable, connected 3-manifold can be obtained by surgery on a link in S^3. (Moreover, one can always perform the surgery with surgery coefficients ± 1 and with each individual component of the link unknotted.)*

4.2 Connected Sums

The following example constructs the connected sum $W = A\#B$ of two 3-manifolds, where A is obtained from a $5/1$ Dehn surgery on the complement of the first prime knot on 11 crossings and B is obtained by a $5/1$ Dehn surgery on the complement of the second prime knot on 11 crossings. The homology groups

$H_1(W,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_{594}$, $H_2(W,\mathbb{Z}) = 0$, $H_3(W,\mathbb{Z}) = \mathbb{Z}$

are computed.

Example

```
gap> ap1:=ArcPresentation(PureCubicalKnot(11,1));;
gap> A:=ThreeManifoldViaDehnSurgery(ap1,5,1);;
gap> ap2:=ArcPresentation(PureCubicalKnot(11,2));;
gap> B:=ThreeManifoldViaDehnSurgery(ap2,5,1);;
gap> W:=ConnectedSum(A,B); #W:=ConnectedSum(A,B,-1) would yield A#-B where -B has the opposite or
Regular CW-complex of dimension 3

gap> Homology(W,1);
[ 2, 594 ]
gap> Homology(W,2);
[  ]
gap> Homology(W,3);
[ 0 ]
```

4.3 Dijkgraaf-Witten Invariant

Given a closed connected orientable 3-manifold W, a finite group G and a 3-cocycle $\alpha \in H^3(BG,U(1))$ Dijkgraaf and Witten define the complex number

$$ Z^{G,\alpha}(W) = \frac{1}{|G|}\sum_{\gamma\in {\rm Hom}(\pi_1W, G)} \langle \gamma^\ast[\alpha], [M]\rangle \ \in\ \mathbb C\ $$ where γ ranges over all group homomorphisms $\gamma\colon \pi_1 W \to G$. This complex number is an invariant of the homotopy type of W and is useful for distinguishing between certain homotopically distinct 3-manifolds.

A homology version of the Dijkgraaf-Witten invariant can be defined as the set of homology homomorphisms $$D_G(W) =\{ \gamma_\ast\colon H_3(W,\mathbb Z) \longrightarrow H_3(BG,\mathbb Z) \}_{\gamma\in {\rm Hom}(\pi_1W, G)}.$$ Since $H_3(W,\mathbb{Z}) \cong \mathbb{Z}$ we represent $D_G(W)$ by the set $D_G(W) = \{\gamma_*(1)\}_{\gamma\in \mathrm{Hom}(\pi_1 W,G)}$ where 1 denotes one of the two possible generators of $H_3(W,\mathbb{Z})$.

For any coprime integers $p,q \geq 1$ the *lens space* $L(p,q)$ is obtained from the 3-sphere by drilling out a tubular neighbourhood of the trivial knot and then gluing a solid torus to the boundary of the cavity via a homeomorphism corresponding to a Dehn surgery coefficient p/q. Lens spaces have cyclic fundamental group $\pi_1(L(p,q)) = C_p$ and homology $H_1(L(p,q),\mathbb{Z}) \cong \mathbb{Z}_p$, $H_2(L(p,q),\mathbb{Z}) \cong 0$, $H_3(L(p,q),\mathbb{Z}) \cong \mathbb{Z}$. It was proved by J.H.C. Whitehead that two lens spaces $L(p,q)$ and $L(p',q')$ are homotopy equivalent if and only if $p = p'$ and $qq' \equiv \pm n^2 \bmod p$ for some integer n.

The following session constructs the two lens spaces $L(5,1)$ and $L(5,2)$. The homology version of the Dijkgraaf-Witten invariant is used with $G = C_5$ to demonstrate that the two lens spaces are not homotopy equivalent.

Example

```
gap> ap:=[[2,1],[2,1]];; #Arc presentation for the trivial knot
```

```
gap> L51:=ThreeManifoldViaDehnSurgery(ap,5,1);;
gap> D:=DijkgraafWittenInvariant(L51,CyclicGroup(5));
[ g1^4, g1^4, g1, g1, id ]

gap> L52:=ThreeManifoldViaDehnSurgery(ap,5,2);;
gap> D:=DijkgraafWittenInvariant(L52,CyclicGroup(5));
[ g1^3, g1^3, g1^2, g1^2, id ]
```

A theorem of Fermat and Euler states that if a prime p is congruent to 3 modulo 4, then for any q exactly one of $\pm q$ is a quadratic residue mod p. For all other primes p either both or neither of $\pm q$ is a quadratic residue mod p. Thus for fixed $p \equiv 3 \bmod 4$ the lens spaces $L(p,q)$ form a single homotopy class. There are precisely two homotopy classes of lens spaces for other p.

The following commands confirm that $L(13,1) \not\simeq L(13,2)$.

Example

```
gap> L13_1:=ThreeManifoldViaDehnSurgery([[1,2],[1,2]],13,1);;
gap> DijkgraafWittenInvariant(L13_1,CyclicGroup(13));
[ g1^12, g1^12, g1^10, g1^10, g1^9, g1^9, g1^4, g1^4, g1^3, g1^3, g1, g1, id ]
gap> L13_2:=ThreeManifoldViaDehnSurgery([[1,2],[1,2]],13,2);;
gap> DijkgraafWittenInvariant(L13_2,CyclicGroup(13));
[ g1^11, g1^11, g1^8, g1^8, g1^7, g1^7, g1^6, g1^6, g1^5, g1^5, g1^2, g1^2,
  id ]
```

4.4 Cohomology rings

The following commands construct the multiplication table (with respect to some basis) for the cohomology rings $H^*(L(13,1),\mathbb{Z}_{13})$ and $H^*(L(13,2),\mathbb{Z}_{13})$. These rings are isomorphic and so fail to distinguish between the homotopy types of the lens spaces $L(13,1)$ and $L(13,2)$.

Example

```
gap> L13_1:=ThreeManifoldViaDehnSurgery([[1,2],[1,2]],13,1);;
gap> L13_2:=ThreeManifoldViaDehnSurgery([[1,2],[1,2]],13,2);;
gap> L13_1:=BarycentricSubdivision(L13_1);;
gap> L13_2:=BarycentricSubdivision(L13_2);;
gap> A13_1:=CohomologyRing(L13_1,13);;
gap> A13_2:=CohomologyRing(L13_2,13);;
gap> M13_1:=List([1..4],i->[]);;
gap> B13_1:=CanonicalBasis(A13_1);;
gap> M13_2:=List([1..4],i->[]);;
gap> B13_2:=CanonicalBasis(A13_2);;
gap> for i in [1..4] do
> for j in [1..4] do
> M13_1[i][j]:=B13_1[i]*B13_1[j];
> od;od;
gap> for i in [1..4] do
> for j in [1..4] do
> M13_2[i][j]:=B13_2[i]*B13_2[j];
> od;od;
gap> Display(M13_1);
[ [            v.1,           v.2,           v.3,           v.4 ],
```

```
  [            v.2,          0*v.1,  (Z(13)^6)*v.4,          0*v.1 ],
  [            v.3,  (Z(13)^6)*v.4,          0*v.1,          0*v.1 ],
  [            v.4,          0*v.1,          0*v.1,          0*v.1 ] ]
gap> Display(M13_2);
[ [          v.1,          v.2,          v.3,          v.4 ],
  [          v.2,        0*v.1,  (Z(13))*v.4,        0*v.1 ],
  [          v.3,  (Z(13))*v.4,        0*v.1,        0*v.1 ],
  [          v.4,        0*v.1,        0*v.1,        0*v.1 ] ]
```

4.5 Linking Form

Given a closed connected ORIENTED 3-manifold W let $\tau H_1(W,\mathbb{Z})$ denote the torsion subgroup of the first integral homology. The *linking form* is a bilinear mapping

$Lk_W : \tau H_1(W,\mathbb{Z}) \times \tau H_1(W,\mathbb{Z}) \longrightarrow \mathbb{Q}/\mathbb{Z}$.

To construct this form note that we have a Poincare duality isomorphism

$\rho : H^2(W,\mathbb{Z}) \xrightarrow{\cong} H_1(W,\mathbb{Z}), z \mapsto z \cap [W]$

involving the cap product with the fundamental class $[W] \in H^3(W,\mathbb{Z})$. That is, $[M]$ is the generator of $H^3(W,\mathbb{Z}) \cong \mathbb{Z}$ determining the orientation. The short exact sequence $\mathbb{Z} \rightarrowtail \mathbb{Q} \twoheadrightarrow \mathbb{Q}/\mathbb{Z}$ gives rise to a cohomology exact sequence

$$\to H^1(W,\mathbb{Q}) \to H^1(W,\mathbb{Q}/\mathbb{Z}) \xrightarrow{\beta} H^2(W,\mathbb{Z}) \to H^2(W,\mathbb{Q}) \to$$

from which we obtain the isomorphism $\beta : \tau H^1(W,\mathbb{Q}/\mathbb{Z}) \xrightarrow{\cong} \tau H^2(W,\mathbb{Z})$. The linking form Lk_W can be defined as the composite

$$Lk_W : \tau H_1(W,\mathbb{Z}) \times \tau H_1(W,\mathbb{Z}) \xrightarrow{1\times\rho^{-1}} \tau H_1(W,\mathbb{Z}) \times \tau H^2(W,\mathbb{Z}) \xrightarrow{1\times\beta^{-1}} \tau H_1(W,\mathbb{Z}) \times \tau H^1(W,\mathbb{Q}/\mathbb{Z}) \xrightarrow{ev} \mathbb{Q}/\mathbb{Z}$$

where $ev(x,\alpha)$ evaluates a 1-cocycle α on a 1-cycle x.

The linking form can be used to define the set

$I^O(W) = \{Lk_W(g,g) : g \in \tau H_1(W,\mathbb{Z})\}$

which is an oriented-homotopy invariant of W. Letting W^+ and W^- denote the two possible orientations on the manifold, the set

$I(W) = \{I^O(W^+), I^O(W^-)\}$

is a homotopy invariant of W which in this manual we refer to as the *linking form homotopy invariant*.

The following commands compute the linking form homotopy invariant for the lens spaces $L(13,q)$ with $1 \le q \le 12$. This invariant distinguishes between the two homotopy types that arise.

Example

```
gap> LensSpaces:=[];;
gap> for q in [1..12] do
> Add(LensSpaces,ThreeManifoldViaDehnSurgery([[1,2],[1,2]],13,q));
> od;
gap> Display(List(LensSpaces,LinkingFormHomotopyInvariant));;
[ [ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
      [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ],

  [ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ], [ 0, 2/13,
        2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],
```

```
[ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
    [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ],

[ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
    [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ],

[ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ],
    [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],

[ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ],
    [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],

[ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ],
    [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],

[ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ],
    [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],

[ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
    [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ],

[ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
    [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ],

[ [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ],
    [ 0, 2/13, 2/13, 5/13, 5/13, 6/13, 6/13, 7/13, 7/13, 8/13, 8/13, 11/13, 11/13 ] ],

[ [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ],
    [ 0, 1/13, 1/13, 3/13, 3/13, 4/13, 4/13, 9/13, 9/13, 10/13, 10/13, 12/13, 12/13 ] ] ]
```

4.6 Determining the homeomorphism type of a lens space

In 1935 K. Reidemeister [Rei35] classified lens spaces up to orientation preserving PL-homeomorphism. This was generalized by E. Moise [Moi52] in 1952 to a classification up to homeomorphism -- his method requred the proof of the Hauptvermutung for 3-dimensional manifolds. In 1960, following a suggestion of R. Fox, a proof was given by E.J. Brody [Bro60] that avoided the need for the Hauptvermutung. Reidemeister's method, using what is know termed *Reidermeister torsion*, and Brody's method, using tubular neighbourhoods of 1-cycles, both require identifying a suitable "preferred" generator of $H_1(L(p,q),\mathbb{Z})$. In 2003 J. Przytycki and A. Yasukhara [PY03] provided an alternative method for classifying lens spaces, which uses the linking form and again requires the identification of a "preferred" generator of $H_1(L(p,q),\mathbb{Z})$.

Przytycki and Yasukhara proved the following.

THEOREM. *Let $\rho: S^3 \to L(p,q)$ be the p-fold cyclic cover and K a knot in $L(p,q)$ that represents a generator of $H_1(L(p,q),\mathbb{Z})$. If $\rho^{-1}(K)$ is the trivial knot, then $Lk_{L(p,q)}([K],[K]) = q/p$ or $= \overline{q}/p \in \mathbb{Q}/\mathbb{Z}$ where $q\overline{q} \equiv 1 \bmod p$.*

The ingredients of this theorem can be applied in HAP, but at present only to small examples of lens spaces. The obstruction to handling large examples is that the current default method for

computing the linking form involves barycentric subdivision to produce a simplicial complex from a regular CW-complex, and then a homotopy equivalence from this typically large simplicial complex to a smaller non-regular CW-complex. However, for homeomorphism invariants that are not homotopy invariants there is a need to avoid homotopy equivalences. In the current version of HAP this means that in order to obtain delicate homeomorphism invariants we have to perform homology computations on typically large simplicial complexes. In a future version of HAP we hope to avoid the obstruction by implementing cup products, cap products and linking forms entirely within the category of regular CW-complexes.

The following commands construct a small lens space $L = L(p,q)$ with unknown values of p,q. Subsequent commands will determine the homeomorphism type of L.

Example

```
gap> p:=Random([2,3,5,7,11,13,17]);;
gap> q:=Random([1..p-1]);;
gap> L:=ThreeManifoldViaDehnSurgery([[1,2],[1,2]],p,q);
Regular CW-complex of dimension 3
```

We can readily determine the value of $p = 11$ by calculating the order of $\pi_1(L)$.

Example

```
gap> F:=FundamentalGroupWithPathReps(L);;
gap> StructureDescription(F);
"C11"
```

The next commands take the default edge path $\gamma: S^1 \to L$ representing a generator of the cyclic group $\pi_1(L)$ and lift it to an edge path $\tilde{\gamma}: S^1 \to \tilde{L}$.

Example

```
gap> U:=UniversalCover(L);;
gap> G:=U!.group;;
gap> p:=EquivariantCWComplexToRegularCWMap(U,Group(One(G)));;
gap> U:=Source(p);;
gap> gamma:=[];;
gap> gamma[2]:=F!.loops[1];;
gap> gamma[2]:=List(gamma[2],AbsInt);;
gap> gamma[1]:=List(gamma[2],k->L!.boundaries[2][k]);;
gap> gamma[1]:=SSortedList(Flat(gamma[1]));;
gap>
gap> gammatilde:=[[],[],[],[]];;
gap> for k in [1..U!.nrCells(0)] do
> if p!.mapping(0,k) in gamma[1] then Add(gammatilde[1],k); fi;
> od;
gap> for k in [1..U!.nrCells(1)] do
> if p!.mapping(1,k) in gamma[2] then Add(gammatilde[2],k); fi;
> od;
gap> gammatilde:=CWSubcomplexToRegularCWMap([U,gammatilde]);
Map of regular CW-complexes
```

The next commands check that the path $\tilde{\gamma}$ is unknotted in $\tilde{L} \cong S^3$ by checking that $\pi_1(\tilde{L} \setminus \text{image}(\tilde{\gamma}))$ is infinite cyclic.

Example
```
gap> C:=RegularCWComplexComplement(gammatilde);
Regular CW-complex of dimension 3

gap> G:=FundamentalGroup(C);
<fp group of size infinity on the generators [ f2 ]>
```

Since $\tilde{\gamma}$ is unkotted the cycle γ represents the preferred generator $[\gamma] \in H_1(L,\mathbb{Z})$. The next commands compute $Lk_L([\gamma],[\gamma]) = 7/11$.

Example
```
gap> LinkingFormHomeomorphismInvariant(L);
[ 7/11 ]
```

The classification of Moise/Brody states that $L(p,q) \cong L(p,q')$ if and only if $qq' \equiv \pm 1 \bmod p$. Hence the lens space L has the homeomorphism type
$L \cong L(11,7) \cong L(11,8) \cong L(11,4) \cong L(11,3)$.

4.7 Surgeries on distinct knots can yield homeomorphic manifolds

The lens space $L(5,1)$ is a quotient of the 3-sphere S^3 by a certain action of the cyclic group C_5. It can be realized by a $p/q = 5/1$ Dehn filling of the complement of the trivial knot. It can also be realized by Dehn fillings of other knots. To see this, the following commands compute the manifold W obtained from a $p/q = 1/5$ Dehn filling of the complement of the trefoil and show that W at least has the same integral homology and same fundamental group as $L(5,1)$.

Example
```
gap> ap:=ArcPresentation(PureCubicalKnot(3,1));;
gap> W:=ThreeManifoldViaDehnSurgery(ap,1,5);;
gap> Homology(W,1);
[ 5 ]
gap> Homology(W,2);
[  ]
gap> Homology(W,3);
[ 0 ]

gap> F:=FundamentalGroup(W);;
gap> StructureDescription(F);
"C5"
```

The next commands construct the universal cover $\widetilde{W}$ and show that it has the same homology as S^3 and trivivial fundamental group $\pi_1(\widetilde{W}) = 0$.

Example
```
gap> U:=UniversalCover(W);;
gap> G:=U!.group;;
gap> Wtilde:=EquivariantCWComplexToRegularCWComplex(U,Group(One(G)));
Regular CW-complex of dimension 3
```

```
gap> Homology(Wtilde,1);
[  ]
gap> Homology(Wtilde,2);
[  ]
gap> Homology(Wtilde,3);
[ 0 ]

gap> F:=FundamentalGroup(Wtilde);
<fp group on the generators [  ]>
```

By construction the space $\widetilde{W}$ is a manifold. Had we not known how the regular CW-complex $\widetilde{W}$ had been constructed then we could prove that it is a closed 3-manifold by creating its barycentric subdivision $K = sd\widetilde{W}$, which is homeomorphic to $\widetilde{W}$, and verifying that the link of each vertex in the simplicial complex $sd\widetilde{W}$ is a 2-sphere. The following command carries out this proof.

Example

```
gap> IsClosedManifold(Wtilde);

true
```

The Poincare conjecture (now proven) implies that $\widetilde{W}$ is homeomorphic to S^3. Hence $W = S^3/C_5$ is a quotient of the 3-sphere by an action of C_5 and is hence a lens space $L(5,q)$ for some q.

The next commands determine that W is homeomorphic to $L(5,4) \cong L(5,1)$.

Example

```
gap> Lk:=LinkingFormHomeomorphismInvariant(W);
[ 0, 4/5 ]
```

Moser [Mos71] gives a precise decription of the lens spaces arising from surgery on the trefoil knot and more generally from surgery on torus knots. Greene [Gre13] determines the lens spaces that arise by integer Dehn surgery along a knot in the three-sphere

4.8 Finite fundamental groups of 3-manifolds

Lens spaces are examples of 3-manifolds with finite fundamental groups. The complete list of finite groups G arising as fundamental groups of closed connected 3-manifolds is recalled in 7.12 where one method for computing their cohomology rings is presented. Their cohomology could also be computed from explicit 3-manifolds W with $\pi_1 W = G$. For instance, the following commands realize a closed connected 3-manifold W with $\pi_1 W = C_{11} \times SL_2(\mathbb{Z}_5)$.

Example

```
gap> ap:=ArcPresentation(PureCubicalKnot(3,1));;
gap> W:=ThreeManifoldViaDehnSurgery(ap,1,11);;
gap> F:=FundamentalGroup(W);;
gap> Order(F);
1320
gap> AbelianInvariants(F);
[ 11 ]
gap> StructureDescription(F);
```

```
"C11 x SL(2,5)"
```

Hence the group $G = C_{11} \times SL_2(\mathbb{Z}_5)$ of order 1320 acts freely on the 3-sphere $\widetilde{W}$. It thus has periodic cohomology with

$$H_n(G,\mathbb{Z}) = \begin{cases} \mathbb{Z}_{11} & n \equiv 1 \bmod 4 \\ 0 & n \equiv 2 \bmod 4 \\ \mathbb{Z}_{1320} & n \equiv 3 \bmod 4 \\ \not\vdash & n \equiv 0 \bmod 4 \end{cases}$$

for $n > 0$.

4.9 Poincare's cube manifolds

In his seminal paper on "Analysis situs", published in 1895, Poincare constructed a series of closed 3-manifolds which played an important role in the development of his theory. A good account of these manifolds is given in the online Manifold Atlas Project (MAP). Most of his examples are constructed by identifications on the faces of a (solid) cube. The function `PoincareCubeCWComplex()` can be used to construct any 3-dimensional CW-complex arising from a cube by identifying the six faces pairwise; the vertices and faces of the cube are numbered as follows

and barycentric subdivision is used to ensure that the quotient is represented as a regular CW-complex.

Examples 3 and 4 from Poincare's paper, described in the following figures taken from MAP,

are constructed in the following example. Both are checked to be orientable manifolds, and are shown to have different homology. (Note that the second example in Poincare's paper is not a manifold -- the links of some of its vertices are not homeomorphic to a 2-sphere.)

Example

```
gap> A:=1;;C:=2;;D:=3;;B:=4;;
gap> Ap:=5;;Cp:=6;;Dp:=7;;Bp:=8;;

gap> L:=[[A,B,D,C],[Bp,Dp,Cp,Ap]];;
gap> M:=[[A,B,Bp,Ap],[Cp,C,D,Dp]];;
gap> N:=[[A,C,Cp,Ap],[D,Dp,Bp,B]];;
gap> Ex3:=PoincareCubeCWComplex(L,M,N);
Regular CW-complex of dimension 3

gap> IsClosedManifold(Ex3);

true

gap> L:=[[A,B,D,C],[Bp,Dp,Cp,Ap]];;
gap> M:=[[A,B,Bp,Ap],[C,D,Dp,Cp]];;
gap> N:=[[A,C,Cp,Ap],[B,D,Dp,Bp]];;
gap> Ex4:=PoincareCubeCWComplex(L,M,N);
Regular CW-complex of dimension 3

gap> IsClosedManifold(Ex4);
true

gap> List([0..3],k->Homology(Ex3,k));
```

```
[ [ 0 ], [ 2, 2 ], [  ], [ 0 ] ]
gap> List([0..3],k->Homology(Ex4,k));
[ [ 0 ], [ 2, 0 ], [ 0 ], [ 0 ] ]
```

4.10 There are at least 25 distinct cube manifolds

The function `PoincareCubeCWComplex(A,G)` can also be applied to two inputs where A is a pairing of the six faces such as $A = [[1,2],[3,4],[5,6]]$ and G is a list of three elements of the dihedral group of order 8 such as $G = [(2,4),(2,4),(2,4)*(1,3)]$. The dihedral elements specify how each pair of faces are glued together. With these inputs it is easy to iterate over all possible values of A and G in order to construct all possible closed 3-manifolds arising from the pairwise identification of faces of a cube. We call such a manifold a CUBE MANIFOLD. Distinct values of A and G can of course yield homeomorphic spaces. To ensure that each possible cube manifold is constructed, at least once, up to homeomorphism it suffices to consider

$A = [[1,2],[3,4],[5,6]], A = [[1,2],[3,5],[4,6]], A = [[1,4],[2,6],[3,5]]$

and all G in $D_8 \times D_8 \times D_8$.

The following commands iterate through these $3 \times 8^3 = 1536$ pairs (A,G) and show that in precisely 163 cases (just over 10% of cases) the quotient CW-complex is a closed 3-manifold.

Example

```
gap> A1:= [ [1,2], [3,4], [5,6] ];;
gap> A2:=[ [1,2], [3,5], [4,6] ];;
gap> A3:=[ [1,4], [2,6], [3,5] ];;
gap> D8:=DihedralGroup(IsPermGroup,8);;

gap> Manifolds:=[];;
gap> for A in [A1,A2,A3] do
> for x in D8 do
> for y in D8 do
> for z in D8 do
> G:=[x,y,z];
> F:=PoincareCubeCWComplex(A,G);
> b:=IsClosedManifold(F);
> if b=true then Add(Manifolds,F); fi;
> od;od;od;od;

gap> Size(Manifolds);
163
```

The following additional commands use integral homology and low index subgroups of fundamental groups to establish that the 163 cube manifolds represent at least 25 distinct homotopy equivalence classes of manifolds. One homotopy class is represented by up to 40 of the manifolds, and at least four of the homotopy classes are each represented by a single manifold..

Example

```
gap> invariant1:=function(m);
> return List([1..3],k->Homology(m,k));
> end;;
```

```
gap> C:=Classify(Manifolds,invariant1);;

gap> invariant2:=function(m)
> local L;
> L:=FundamentalGroup(m);
> if GeneratorsOfGroup(L)= [] then return [];fi;
> L:=LowIndexSubgroupsFpGroup(L,5);
> L:=List(L,AbelianInvariants);
> L:=SortedList(L);
> return L;
> end;;

gap> D:=RefineClassification(C,invariant2);;
gap> List(D,Size);
[ 40, 2, 10, 15, 8, 6, 2, 6, 2, 5, 7, 1, 4, 11, 7, 7, 10, 4, 4, 2, 1, 3, 1,
  1, 4 ]
```

The next commands construct a list of 18 orientable cube manifolds and a list of 7 non-orientable cube manifolds.

Example

```
gap> Manifolds:=List(D,x->x[1]);;
gap> OrientableManifolds:=Filtered(Manifolds,m->Homology(m,3)=[0]);;
gap> NonOrientableManifolds:=Filtered(Manifolds,m->Homology(m,3)=[]);;
gap> Length(OrientableManifolds);
18
gap> Length(NonOrientableManifolds);
7
```

The next commands show that the 7 non-orientable cube manifolds all have infinite fundamental groups.

Example

```
gap> List(NonOrientableManifolds,m->IsFinite(FundamentalGroup(m)));
[ false, false, false, false, false, false, false ]
```

The final commands show that (at least) 9 of the orientable manifolds have finite fundamental groups and list the isomorphism types of these finite groups. Note that it is now known that any closed 3-manifold with finite fundamental group is spherical (i.e. is a quotient of the 3-sphere). Spherical manifolds with cyclic fundamental group are, by definition, lens spaces.

Example

```
gap> List(OrientableManifolds{[4,8,10,11,12,13,15,16,18]},m->
          IsFinite(FundamentalGroup(m)));
[ true, true, true, true, true, true, true, true, true ]

gap> List(OrientableManifolds{[4,8,10,11,12,13,15,16,18]},m->
          StructureDescription(FundamentalGroup(m)));
[ "Q8", "C2", "C4", "C3 : C4", "C12", "C8", "C14", "C6", "1" ]
```

4.10.1 Face pairings for 25 distinct cube manifolds

The following are the face pairings of 25 non-homeomorphic cube manifolds, with vertices of the cube numbered as describe above.

Example

```
gap> for i in [1..25] do
> p:=Manifolds[i]!.cubeFacePairings;
> Print("Manifold ",i," has face pairings:\n");
> Print(p[1],"\n",p[2],"\n",p[3],"\n");
> Print("Fundamental group is:  ");
> if i in [ 1, 9, 12, 14, 15, 16, 17, 18, 19, 20, 22, 23, 25 ] then
> Print(StructureDescription(FundamentalGroup(Manifolds[i])),"\n");
> else Print("infinite non-cyclic\n"); fi;
> if Homology(Manifolds[i],3)=[0] then Print("Orientable, ");
> else Print("Non orientable, "); fi;
> Print(ManifoldType(Manifolds[i]),"\n");
> for x in Manifolds[i]!.edgeDegrees do
> Print(x[2]," edges of \"degree\" ",x[1],",  ");
> od;
> Print("\n\n");
> od;

Manifold 1 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 7, 8, 4 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 8, 7, 6 ] ]
[ [ 1, 4, 8, 5 ], [ 3, 2, 6, 7 ] ]
Fundamental group is:  Z x C2
Non orientable, other
4 edges of "degree" 2,  4 edges of "degree" 4,

Manifold 2 has face pairings:
[ [ 1, 5, 6, 2 ], [ 7, 8, 4, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
[ [ 5, 8, 7, 6 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Non orientable, other
2 edges of "degree" 1,  2 edges of "degree" 3,  2 edges of "degree" 8,

Manifold 3 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 7, 8, 4 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
[ [ 1, 4, 8, 5 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  infinite non-cyclic
Non orientable, euclidean
6 edges of "degree" 4,

Manifold 4 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 7, 8, 4 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
[ [ 1, 4, 8, 5 ], [ 6, 7, 3, 2 ] ]
Fundamental group is:  infinite non-cyclic
Non orientable, euclidean
6 edges of "degree" 4,
```

```
Manifold 5 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 7, 8, 4 ] ]
[ [ 1, 2, 3, 4 ], [ 6, 5, 8, 7 ] ]
[ [ 1, 4, 8, 5 ], [ 2, 6, 7, 3 ] ]
Fundamental group is:  infinite non-cyclic
Non orientable, euclidean
6 edges of "degree" 4,

Manifold 6 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
[ [ 1, 4, 8, 5 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, euclidean
6 edges of "degree" 4,

Manifold 7 has face pairings:
[ [ 1, 5, 6, 2 ], [ 7, 3, 4, 8 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
[ [ 5, 8, 7, 6 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, other
2 edges of "degree" 1,  2 edges of "degree" 3,  2 edges of "degree" 8,

Manifold 8 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 7, 8, 5, 6 ] ]
[ [ 1, 4, 8, 5 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, other
4 edges of "degree" 2,  2 edges of "degree" 8,

Manifold 9 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 8, 5, 6, 7 ] ]
[ [ 1, 4, 8, 5 ], [ 6, 2, 3, 7 ] ]
Fundamental group is:  Q8
Orientable, spherical
8 edges of "degree" 3,

Manifold 10 has face pairings:
[ [ 1, 5, 6, 2 ], [ 4, 8, 7, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 7, 8, 5, 6 ] ]
[ [ 1, 4, 8, 5 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, other
4 edges of "degree" 2,  4 edges of "degree" 4,

Manifold 11 has face pairings:
[ [ 1, 5, 6, 2 ], [ 4, 3, 7, 8 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
```

```
[ [ 1, 4, 8, 5 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  infinite non-cyclic
Non orientable, euclidean
6 edges of "degree" 4,

Manifold 12 has face pairings:
[ [ 1, 5, 6, 2 ], [ 4, 8, 7, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
[ [ 1, 4, 8, 5 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  Z x Z x Z
Orientable, euclidean
6 edges of "degree" 4,

Manifold 13 has face pairings:
[ [ 1, 5, 6, 2 ], [ 4, 8, 7, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 6, 7, 8 ] ]
[ [ 1, 4, 8, 5 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, euclidean
6 edges of "degree" 4,

Manifold 14 has face pairings:
[ [ 1, 5, 6, 2 ], [ 7, 3, 4, 8 ] ]
[ [ 1, 2, 3, 4 ], [ 7, 8, 5, 6 ] ]
[ [ 1, 4, 8, 5 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  C2
Orientable, spherical
12 edges of "degree" 2,

Manifold 15 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 7, 8, 4 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
[ [ 5, 8, 7, 6 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  Z
Non orientable, other
4 edges of "degree" 1,  2 edges of "degree" 2,  2 edges of "degree" 8,

Manifold 16 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
[ [ 5, 8, 7, 6 ], [ 2, 3, 7, 6 ] ]
Fundamental group is:  Z
Orientable, other
4 edges of "degree" 1,  2 edges of "degree" 2,  2 edges of "degree" 8,

Manifold 17 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
[ [ 5, 8, 7, 6 ], [ 3, 7, 6, 2 ] ]
Fundamental group is:  C4
Orientable, spherical
2 edges of "degree" 1,  2 edges of "degree" 3,  2 edges of "degree" 8,
```

```
Manifold 18 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 8, 4, 1, 5 ] ]
[ [ 5, 8, 7, 6 ], [ 6, 2, 3, 7 ] ]
Fundamental group is:  C3 : C4
Orientable, spherical
2 edges of "degree" 2,  4 edges of "degree" 5,

Manifold 19 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 8, 4, 1, 5 ] ]
[ [ 5, 8, 7, 6 ], [ 3, 7, 6, 2 ] ]
Fundamental group is:  C12
Orientable, spherical
2 edges of "degree" 2,  2 edges of "degree" 3,  2 edges of "degree" 7,

Manifold 20 has face pairings:
[ [ 1, 5, 6, 2 ], [ 3, 4, 8, 7 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 8, 4, 1 ] ]
[ [ 5, 8, 7, 6 ], [ 3, 7, 6, 2 ] ]
Fundamental group is:  C8
Orientable, spherical
8 edges of "degree" 3,

Manifold 21 has face pairings:
[ [ 1, 5, 6, 2 ], [ 7, 3, 4, 8 ] ]
[ [ 1, 2, 3, 4 ], [ 8, 4, 1, 5 ] ]
[ [ 5, 8, 7, 6 ], [ 7, 6, 2, 3 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, euclidean
6 edges of "degree" 4,

Manifold 22 has face pairings:
[ [ 1, 5, 6, 2 ], [ 5, 6, 7, 8 ] ]
[ [ 3, 7, 8, 4 ], [ 7, 6, 2, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 8, 4, 1, 5 ] ]
Fundamental group is:  C14
Orientable, spherical
2 edges of "degree" 2,  4 edges of "degree" 5,

Manifold 23 has face pairings:
[ [ 1, 5, 6, 2 ], [ 5, 6, 7, 8 ] ]
[ [ 3, 7, 8, 4 ], [ 7, 6, 2, 3 ] ]
[ [ 1, 2, 3, 4 ], [ 5, 8, 4, 1 ] ]
Fundamental group is:  C6
Orientable, spherical
6 edges of "degree" 2,  2 edges of "degree" 6,

Manifold 24 has face pairings:
[ [ 1, 5, 6, 2 ], [ 7, 8, 5, 6 ] ]
[ [ 3, 7, 8, 4 ], [ 2, 3, 7, 6 ] ]
```

```
[ [ 1, 2, 3, 4 ], [ 4, 1, 5, 8 ] ]
Fundamental group is:  infinite non-cyclic
Orientable, euclidean
6 edges of "degree" 4,

Manifold 25 has face pairings:
[ [ 1, 5, 6, 2 ], [ 6, 7, 8, 5 ] ]
[ [ 3, 7, 8, 4 ], [ 3, 7, 6, 2 ] ]
[ [ 1, 2, 3, 4 ], [ 1, 5, 8, 4 ] ]
Fundamental group is:  1
Orientable, spherical
4 edges of "degree" 1,  4 edges of "degree" 5,
```

4.10.2 Platonic cube manifolds

A *platonic solid* is a convex, regular polyhedron in 3-dimensional euclidean $\mathbb{E}^3$ or spherical $\mathbb{S}^3$ or hyperbolic space $\mathbb{H}^3$. Being *regular* means that all edges are congruent, all faces are congruent, all angles between adjacent edges in a face are congruent, all dihedral angles between adjacent faces are congruent. A platonic cube in euclidean space has six congruent square faces with diherdral angle $\pi/2$. A platonic cube in spherical space has dihedral angles $2\pi/3$. A platonic cube in hyperbolic space has dihedral angles $2\pi/5$. This can alternatively be expressed by saying that in a tessellation of $\mathbb{E}^3$ by platonic cubes each edge is adjacent to 4 square faces. In a tessellation of $\mathbb{S}^3$ by platonic cubes each edge is adjacent to 3 square faces. In a tessellation of $\mathbb{H}^3$ by platonic cubes each edge is adjacent to 5 five square faces.

Any cube manifold M induces a cubical CW-decomposition of its universal cover $\widetilde{M}$. We say that M is a *platonic cube manifold* if every edge in $\widetilde{M}$ is adjacent to 4 faces in the euclidean case $\widetilde{M} = \mathbb{E}^3$, is adjacent to 3 faces in the spherical case $\widetilde{M} = \mathbb{S}^3$, is adjacent to 5 faces in the hyperbolic case $\widetilde{M} = \mathbb{H}^3$.

In the above list of 25 cube manifolds we see that the euclidean manifolds 3, 4, 5, 6, 11 are platonic and that the spherical manifolds 9, 20 are platonic.

4.11 There are at most 41 distinct cube manifolds

Using the Simpcomp package for GAP we can show that many of the 163 cube manifolds constructed above are homeomorphic. We do this by showing that barycentric subdivisions of many of the manifolds are combinatorially the same.

The following commands establish homeomorphisms (simplicial complex isomorphisms) between manifolds in each equivalence class D[i] above for $1 \leq i \leq 25$, and then discard all but one manifold in each homeomorphism class. We are left with 59 cube manifolds, some of which may be homeomorphic, representing at least 25 distinct homeomorphism classes. The 59 manifolds are stored in the list DD of length 25 each of whose terms is a list of cube manifolds.

Example

```
gap> LoadPackage("Simpcomp");;

gap> inv3:=function(m)
> local K;
> K:=BarycentricSubdivision(m);
> K:=MaximalSimplicesOfSimplicialComplex(K);
```

```
> K:=SC(K);
> if not SCIsStronglyConnected(K) then Print("WARNING!\n"); fi;
> return SCExportIsoSig( K );
> end;
function( m ) ... end

gap> DD:=[];;
gap> for x in D do
> y:=Classify(x,inv3);
> Add(DD,List(y,z->z[1]));
> od;

gap> List(DD,Size);
[ 9, 1, 3, 3, 3, 1, 1, 1, 1, 1, 2, 1, 2, 7, 4, 4, 3, 1, 1, 1, 1, 3, 1, 1, 3 ]
```

The function `PoincareCubeCWCompex()` applies cell simplifications in its construction of the quotient of a CW-complex. A variant `PoincareCubeCWCompexNS()` performs no cell simplifications and thus returns a bigger cell complex which we can attempt to use to establish further homeomorphisms. This is done in the following session and succeeds in showing that there are at most 51 distinct homeomorphism types of cube manifolds.

Example

```
gap> DD:=List(DD,x->List(x,y->PoincareCubeCWComplexNS(
> y!.cubeFacePairings[1],y!.cubeFacePairings[2],y!.cubeFacePairings[3])));;

gap> D:=[];;
gap> for x in DD do
> y:=Classify(x,inv3);
> Add(D,List(y,z->z[1]));
>od;;

gap> List(D,Size);
[ 8, 1, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1, 2, 4, 4, 4, 3, 1, 1, 1, 1, 1, 1, 1, 2 ]
```

Making further modifications to the cell structures of the manifolds that leave their homeomorphism types unchanged can help to identify further simplicial isomorphisms between barycentric subdivisions. For instance, the following commands succeed in establishing that there are at most 45 distinct homeomorphism types of cube manifolds.

Example

```
gap> DD:=[];;
gap> for x in D do
> if Length(x)>1 then
> Add(DD, List(x,y->BarycentricallySimplifiedComplex(y)));
> else Add(DD,x);
> fi;
> od;
gap> D:=[];;

gap> for x in DD do
> y:=Classify(x,inv3);
```

```
> Add(D,List(y,z->z[1]));
> od;

gap> List(D,Size);
[ 7, 1, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1, 2, 2, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 2 ]

gap> DD:=List(D,x->List(x,y->PoincareCubeCWComplexNS(
> y!.cubeFacePairings[1],y!.cubeFacePairings[2],y!.cubeFacePairings[3])));;

gap> D1:=[];;
gap> for x in DD do
> if Length(x)>1 then
> Add(D1, List(x,y->BarycentricallySimplifiedComplex(RegularCWComplex(BarycentricSubdivision(y)))
> else Add(D1,x);
> fi;
> od;

gap> DD:=[];;
gap> for x in D1 do
> y:=Classify(x,inv3);
> Add(DD,List(y,z->z[1]));
> od;;

gap> Print(List(DD,Size),"\n");
[ 6, 1, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1, 2, 2, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 2 ]
```

The two manifolds in DD[14] have fundamental group C_2 and are thus lens spaces. There is only one homeomorphism class of such lens spaces and so these two manifolds are homeomorphic. The three manifolds in DD[17] are lens spaces with fundamental group C_4. Again, there is only one homeomorphism class of such lens spaces and so these three manifolds are homeomorphic. The two manifolds in DD[25] have trivial fundamental group and are hence both homeomorphic to the 3-sphere. These observations mean that there are at most 41 closed manifolds arising from a cube by identifying the cube's faces pairwise.

These observations can be incorporated into our list DD of equivalence classes of manifolds as follows.

Example

```
gap> DD[14]:=DD[14]{[1]};;
gap> DD[17]:=DD[17]{[1]};;
gap> DD[25]:=DD[25]{[1]};;
gap> List(DD,Size);
[ 6, 1, 3, 3, 2, 1, 1, 1, 1, 1, 2, 1, 2, 1, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

4.12 There are precisely 18 orientable cube manifolds, of which 9 are spherical and 5 are euclidean

The following commands show that there are at least 18 and at most 21 orientable cube manifolds.

Example
```
gap> DDorient:=Filtered(DD,x->Homology(x[1],3)=[0]);;
gap> List(DDorient,Size);
[ 1, 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]
```

The next commands show that the fundamental groups of the two manifolds in DDorient[7] are isomorphic to $\mathbb{Z} \times \mathbb{Z} : \mathbb{Z}$, and that the fundamental groups of the three manifolds in DDorient[9] are isomorphic to $\mathbb{Z}$.

Example
```
gap> g1:=FundamentalGroup(DDorient[7][1]);;
gap> g2:=FundamentalGroup(DDorient[7][2]);;
gap> RelatorsOfFpGroup(g1);
[ f1^-1*f2*f1*f2^-1, f3^-1*f1*f3*f1, f3^-1*f2^-1*f3*f2^-1 ]
gap> RelatorsOfFpGroup(g2);
[ f1*f2*f1^-1*f2^-1, f1^-1*f3*f1^-1*f3^-1, f3*f2*f3^-1*f2 ]

gap> h1:=FundamentalGroup(DDorient[9][1]);;
gap> h2:=FundamentalGroup(DDorient[9][2]);;
gap> h3:=FundamentalGroup(DDorient[9][3]);;
gap> StructureDescription(h1);
"Z"
gap> StructureDescription(h2);
"Z"
gap> StructureDescription(h3);
"Z"
```

Since neither $\mathbb{Z} \times \mathbb{Z} : \mathbb{Z}$ nor $\mathbb{Z}$ is a free product of two non-trivial groups we conclude that the manifolds in DDorient[7] and DDorient[9] are prime. Since oriented prime 3-manifolds are determined up to homeomorphism by their fundamental groups we can conclude that there are precisely 18 orientable closed manifolds arising from a cube by identifying the cube's faces pairwise.

A compact 3-manifold M is *spherical* if it is of the form $M = S^3/\Gamma$ where Γ is a finite group acting freely as rotations on S^3. The fundamental group of M is then the finite group Γ. Perelmen showed that a compact 3-manifold is spherical if and only if its fundamental group is finite.

A compact 3-manifold is *euclidean* if it is of the form $M = \mathbb{R}^3/\Gamma$ where Γ is a group of affine transformations acting freely on $\mathbb{R}^3$. The fundamental group is then Γ and is called a *Bieberbach group* of dimension 3. It can be shown that a group Γ is isomorphic to a Bieberbach group of dimension n if and only if there is a short exact sequence $\mathbb{Z}^n \rightarrowtail \Gamma \twoheadrightarrow P$ with P a finite group.

The following command establishes that there are precisely 9 orientable spherical manifolds and 5 closed orientable euclidean manifolds arising from pairwise identifications of the faces of the cube.

Example
```
gap> List(OrientableManifolds,ManifoldType);
[ "euclidean", "other", "other", "spherical", "other", "euclidean",
  "euclidean", "spherical", "other", "spherical", "spherical", "spherical",
  "spherical", "euclidean", "spherical", "spherical", "euclidean", "spherical" ]
```

4.13 Cube manifolds with boundary

If a space Y obtained from identifying faces of the cube fails to be a manifold then it fails because one or more vertices of Y fail to have a spherical link. By using barycentric subdivision if necessary, we can ensure that the stars of any two non-manifold vertices of Y have trivial intersection. Removing the stars of the non-manifold vertices from Y yields a 3-manifold with boundary $\hat{Y}$.

The following commands show that there are 367 combinatorially different regular CW-complexes Y that arise by identifying faces of a cube in pairs and which fail to be manifolds. The commands also show that these spaces give rise to at least 180 non-homeomorphic manifolds $\hat{Y}$ with boundary.

Example

```
gap> A1:= [ [1,2], [3,4], [5,6] ];;
gap> A2:=[ [1,2], [3,5], [4,6] ];;
gap> A3:=[ [1,4], [2,6], [3,5] ];;
gap> D8:=DihedralGroup(IsPermGroup,8);;

gap> NonManifolds:=[];;
gap> for A in [A1,A2,A3] do
> for x in D8 do
> for y in D8 do
> for z in D8 do
> G:=[x,y,z];
> F:=PoincareCubeCWComplex(A,G);
> b:=IsClosedManifold(F);
> if b=false then Add(NonManifolds,F); fi;
> od;od;od;od;

gap> D:=Classify(NonManifolds,inv3); #See above for inv3
gap> D:=List(D,x->x[1]);;
gap> Size(D);
367

gap> M:=List(D,ThreeManifoldWithBoundary);;
gap> C:=Classify(M,invariant1);; #See above for invariant1
gap> List(C,Size);
[ 33, 13, 3, 18, 21, 7, 6, 13, 51, 2, 1, 15, 11, 11, 1, 35, 2, 2, 6, 15,
  17, 2, 3, 2, 14, 17, 3, 1, 25, 8, 4, 1, 4 ]

gap> inv5:=function(m)
> local B;
> B:=BoundaryOfPureRegularCWComplex(m);;
> return invariant1(B);
> end;;

gap> CC:=RefineClassification(C,inv5);;
gap> List(CC,Size);
[ 25, 5, 3, 5, 4, 4, 2, 1, 11, 3, 4, 7, 3, 6, 4, 1, 5, 1, 1, 5, 1, 13, 4,
  6, 40, 1, 2, 1, 11, 4, 5, 3, 1, 2, 7, 4, 1, 14, 11, 10, 2, 2, 6, 9, 3, 3,
  2, 15, 2, 3, 2, 14, 17, 2, 1, 1, 4, 7, 14, 8, 3, 1, 1, 4 ]

gap> CC:=RefineClassification(CC,invariant2);;
gap> List(CC,Size);
```

```
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 1, 3, 1, 1, 2, 1, 2, 1, 4, 2, 3, 2, 3,
  4, 3, 2, 1, 1, 3, 2, 4, 3, 1, 1, 5, 1, 1, 3, 1, 1, 1, 13, 3, 1, 4, 2, 1,
  2, 2, 3, 3, 3, 4, 4, 2, 4, 4, 4, 4, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 2, 3, 4, 3, 1, 2, 3, 2, 3, 4, 3, 3, 2, 2, 1, 1, 2, 1, 1, 2,
  1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 10, 5, 2, 3, 2, 14, 17, 1, 1, 1,
  1, 4, 5, 2, 9, 1, 4, 7, 1, 3, 1, 1, 4 ]
gap> Length(CC);
180
```

4.14 Octahedral manifolds

The above construction of 3-manifolds as quotients of a cube can be extended to other polytopes. A polytope of particular interest, and one that appears several times in the classic book on Three-Manifolds by William Thurston [Thu02], is the octahedron. The function `PoincareOctahahedronCWComplex()` can be used to construct any 3-dimensional CW-complex arising from an octahedron by identifying the eight faces pairwise; the vertices and faces of the octahedron are numbered as follows.

The following commands construct a spherical 3-manifold Y with fundamental group equal to the binary tetrahedral group G. The commands then use the universal cover of this manifold to construct the first four terms of a free periodic $\mathbb{Z}G$-resolution of $\mathbb{Z}$ of period 4. The resolution has one free generator in dimensions $4n$ and $4n+3$ for $n \geq 0$. It has two free generators in dimensions $4n+1$ and $4n+2$.

Example

```
gap> L:=[ [ 1, 4, 5 ], [ 2, 6, 3 ] ];;
gap> M:=[ [ 3, 4, 5 ], [ 6, 1, 2 ] ];;
gap> N:=[ [ 2, 3, 5 ], [ 6, 4, 1 ] ];;
gap> P:=[ [ 1, 2, 5 ], [ 6, 3, 4 ] ];;
gap> Y:=PoincareOctahedronCWComplex(L,M,N,P);;
gap> IsClosedManifold(Y);
true

gap> G:=FundamentalGroup(Y);;
gap> StructureDescription(G);
"SL(2,3)"

gap> R:=ChainComplexOfUniversalCover(Y);
Equivariant chain complex of dimension 3

gap> List([0..3],R!.dimension);
[ 1, 2, 2, 1 ]
```

4.15 Dodecahedral manifolds

Another polytope of interest, and one that can be used to construct the Poincare homology sphere, is the dodecahedron. The function `PoincareDodecahedronCWComplex()` can be used to construct

any 3-dimensional CW-complex arising from a dodecahedron by identifying the 12 pentagonal faces pairwise; the vertices of the prism are numbered as follows.

The following commands construct the Poincare homology 3-sphere (with fundamental group equal to the binary icosahedral group of order 120).

Example

```
gap> Y:=PoincareDodecahedronCWComplex(
> [[1,2,3,4,5],[6,7,8,9,10]],
> [[1,11,16,12,2],[19,9,8,18,14]],
> [[2,12,17,13,3],[20,10,9,19,15]],
> [[3,13,18,14,4],[16,6,10,20,11]],
> [[4,14,19,15,5],[17,7,6,16,12]],
> [[5,15,20,11,1],[18,8,7,17,13]]);
Regular CW-complex of dimension 3
gap> IsClosedManifold(Y);
true
gap> List([0..3],n->Homology(Y,n));
[ [ 0 ], [  ], [  ], [ 0 ] ]
gap> StructureDescription(FundamentalGroup(Y));
"SL(2,5)"
```

The following commands construct Seifert-Weber space, a rational homology sphere.

Example

```
gap> W:=PoincareDodecahedronCWComplex(
> [[1,2,3,4,5],[7,8,9,10,6]],
> [[1,11,16,12,2],[9,8,18,14,19]],
> [[2,12,17,13,3],[10,9,19,15,20]],
> [[3,13,18,14,4],[6,10,20,11,16]],
> [[4,14,19,15,5],[7,6,16,12,17]],
> [[5,15,20,11,1],[8,7,17,13,18]]);
Regular CW-complex of dimension 3
gap> IsClosedManifold(W);
true
gap> List([0..3],n->Homology(W,n));
[ [ 0 ], [ 5, 5, 5 ], [  ], [ 0 ] ]
```

4.16 Prism manifolds

Another polytope of interest is the prism constructed as the direct product $D_n \times [0,1]$ of an n-gonal disk D_n with the unit interval. The function `PoincarePrismCWComplex()` can be used to construct any 3-dimensional CW-complex arising from a prism with even $n \geq 4$ by identifying the $n+2$ faces pairwise; the vertices of the prism are numbered as follows.

The case $n = 4$ is that of a cube. The following commands construct a manifold Y arising from a hexagonal prism ($n = 6$) with fundamental group $\pi_1 Y = C_5 \times Q_{32}$ equal to the direct product of the cyclic group of order 5 and the quaternion group of order 32.

Example

```
gap> L:=[[1,2,3,4,5,6],[11,12,7,8,9,10]];;
gap> M:=[[1,7,8,2],[4,5,11,10]];;
gap> N:=[[2,8,9,3],[6,1,7,12]];;
```

```
gap> P:=[[3,9,10,4],[6,12,11,5]];;
gap> Y:=PoincarePrismCWComplex(L,M,N,P);;
gap> IsClosedManifold(Y);
true
gap> G:=FundamentalGroup(Y);;
gap> StructureDescription(G);
"C5 x Q32"
```

An exhaustive search through all manifolds constructed from a hexagonal prism by identify faces pairwise shows that the finite groups arising as fundamental groups are precisely: Q_8, Q_{16}, C_4, $C_3:C_4$, $C_5:C_4$, C_8, C_{16}, C_{12}, C_{20}, C_2, C_6, $C_3\times Q_8$, $C_3\times Q_{16}$, $C_5\times Q_{32}$. Each of these finite groups $G=\pi_1 Y$ is either cyclic (in which case the corresponding manifold is a lens space) or else has the propert that $G/Z(G)$ is dihedral (in which case the corresponding manifold is called a *prism manifold*). The majority of the manifolds arising from a hexagonal prism have infinite fundamental group.

Infinite families of spherical 3-maniolds can be constructed from the infinite family of prisms. For instance, a prism manifold which we denote by P_r can be obtained from a prism $D_{2r}\times[0,1]$ by identifying the left and right side under a twist of π/r, and identifying opposite square faces under a twist of $\pi/2$. Its fundamental group $\pi_1 P_r$ is the binary dihedral group of order $4r$. The following commands construct P_r for $r=3$.

Example

```
gap> L:=[[1,2,3,4,5,6],[8,9,10,11,12,7]];;
gap> M:=[[1,7,8,2],[11,10,4,5]];;
gap> N:=[[2,8,9,3],[12,11,5,6]];;
gap> P:=[[3,9,10,4],[7,12,6,1]];;
gap> Y:=PoincarePrismCWComplex(L,M,N,P);;
gap> IsClosedManifold(Y);
true
gap> StructureDescription(FundamentalGroup(Y));
"C3 : C4"
```

4.17 Bipyramid manifolds

Yet another polytope of interest is the bipyramid constructed as the suspension of an n-gonal disk D_n. The function `PoincareBipyramidCWComplex()` can be used to construct any 3-dimensional CW-complex arising from a bipyramid with $n\geq 3$ by identifying the $2n$ faces pairwise; the vertices of the prism are numbered as follows.

For $n=4$ the bipyramid is the octahedron.

4.18 Flat manifolds, orbifolds and crystallographic groups

Given an n-dimensional crystallographic group G the quotient $X=\mathbb{R}^n/G$ will be an orbifold. If G is Bieberbach (i.e. if it acts on $\mathbb{R}^n$ without fixed points) then the quotient X will be a flat manifold. The following example constructs a 3-dimensional flat manifold and computes its homology.

Example

```
gap> G:=SpaceGroup(3,142);;
gap> IsAlmostBieberbachGroup(Image(IsomorphismPcpGroup(G)));
true
gap> K:=CrystallographicComplex(G);;
gap> KK:=BarycentricSubdivision(K);;
gap> KK!.group:=G;;
gap> Y:=GComplexToRegularCWComplex(KK,G);
Regular CW-complex of dimension 3

gap> Homology(Y,1);
[ 3, 0 ]
gap> Homology(Y,2);
[ 0 ]
gap> Homology(Y,3);
[ 0 ]
```

The following example constructs a 4-dimensional orbifold arising as the quotient of the 4-torus by the permutation action of the cyclic group C_5.

Example

```
gap> G:=SpaceGroup(4,4499);;
gap> StructureDescription(PointGroup(G));
"C5"
gap> IsAlmostBieberbachGroup(Image(IsomorphismPcpGroup(G)));
false
gap> K:=CrystallographicComplex(G);;
KK:=BarycentricSubdivision(K);;
gap> KK!.group:=G;;
gap> Y:=GComplexToRegularCWComplex(KK,G);
Regular CW-complex of dimension 4

gap> Homology(Y,1);
[  ]
gap> Homology(Y,2);
[ 5, 5, 0, 0 ]
gap> Homology(Y,3);
[  ]
gap> Homology(Y,4);
[ 0 ]

gap> Y:=SimplifiedComplex(Y);;
gap> Size(Y); #number of cells in Y
6948
gap> Y!.nrCells(4); #number of 4-dimensional cells in Y
126
```

Chapter 5

Topological data analysis

5.1 Persistent homology

Pairwise distances between 74 points from some metric space have been recorded and stored in a 74×74 matrix D. The following commands load the matrix, construct a filtration of length 100 on the first two dimensions of the assotiated clique complex (also known as the *Vietoris-Rips Complex*), and display the resulting degree 0 persistent homology as a barcode. A single bar with label n denotes n bars with common starting point and common end point.

Example

```
gap> file:=HapFile("data253a.txt");;
gap> Read(file);

gap> G:=SymmetricMatrixToFilteredGraph(D,100);
Filtered graph on 74 vertices.

gap> K:=FilteredRegularCWComplex(CliqueComplex(G,2));
Filtered regular CW-complex of dimension 2

gap> P:=PersistentBettiNumbers(K,0);;
gap> BarCodeCompactDisplay(P);
```

The first 54 terms in the filtration each have 74 path components -- one for each point in the sample. During the next 9 filtration terms the number of path components reduces, meaning that sample points begin to coalesce due to the formation of edges in the simplicial complexes. Then, two path components persist over an interval of 18 filtration terms, before they eventually coalesce.

The next commands display the resulting degree 1 persistent homology as a barcode.

Example

```
gap> P:=PersistentBettiNumbers(K,1);;
gap> BarCodeCompactDisplay(P);
```

Interpreting short bars as noise, we see for instance that the 65th term in the filtration could be regarded as noiseless and belonging to a "stable interval" in the filtration with regards to first and second homology functors. The following command displays (up to homotopy) the 1 skeleton of the simplicial complex arizing as the 65-th term in the filtration on the clique complex.

Example

```
gap> Y:=FiltrationTerm(K,65);
Regular CW-complex of dimension 1

gap> Display(HomotopyGraph(Y));
```

These computations suggest that the dataset contains two persistent path components (or clusters), and that each path component is in some sense periodic. The final command displays one possible representation of the data as points on two circles.

5.1.1 Background to the data

Each point in the dataset was an image consisting of 732×761 pixels. This point was regarded as a vector in $\mathbb{R}^{557052} = \mathbb{R}^{732\times761}$ and the matrix D was constructed using the Euclidean metric. The images were the following:

5.2 Mapper clustering

The following example reads in a set S of vectors of rational numbers. It uses the Euclidean distance $d(u,v)$ between vectors. It fixes some vector $u_0 \in S$ and uses the associated function $f: D \to [0,b] \subset \mathbb{R}, v \mapsto d(u_0,v)$. In addition, it uses an open cover of the interval $[0,b]$ consisting of 100 uniformly distributed overlapping open subintervals of radius $r = 29$. It also uses a simple clustering algorithm implemented in the function `cluster`.

These ingredients are input into the Mapper clustering procedure to produce a simplicial complex M which is intended to be a representation of the data. The complex M is 1-dimensional and the final command uses GraphViz software to visualize the graph. The nodes of this simplicial complex are "buckets" containing data points. A data point may reside in several buckets. The number of points in the bucket determines the size of the node. Two nodes are connected by an edge when they contain common data points.

Example

```
gap> file:=HapFile("data134.txt");;
gap> Read(file);
gap> dx:=EuclideanApproximatedMetric;;
gap> dz:=EuclideanApproximatedMetric;;
gap> L:=List(S,x->Maximum(List(S,y->dx(x,y))));;
gap> n:=Position(L,Minimum(L));;
gap> f:=function(x); return [dx(S[n],x)]; end;;
gap> P:=30*[0..100];; P:=List(P, i->[i]);;
gap> r:=29;;
gap> epsilon:=75;;
gap>  cluster:=function(S)
>   local Y, P, C;
>   if Length(S)=0 then return S; fi;
>   Y:=VectorsToOneSkeleton(S,epsilon,dx);
>   P:=PiZero(Y);
>   C:=Classify([1..Length(S)],P[2]);
>   return List(C,x->S{x});
>  end;;
```

```
gap> M:=Mapper(S,dx,f,dz,P,r,cluster);
Simplicial complex of dimension 1.

gap> Display(GraphOfSimplicialComplex(M));
```

5.2.1 Background to the data

The datacloud S consists of the 400 points in the plane shown in the following picture.

5.3 Some tools for handling pure complexes

A CW-complex X is said to be *pure* if all of its top-dimensional cells have a common dimension. There are instances where such a space X provides a convenient ambient space whose subspaces can be used to model experimental data. For instance, the plane $X = \mathbb{R}^2$ admits a pure regular CW-structure whose 2-cells are open unit squares with integer coordinate vertices. An alternative, and sometimes preferrable, pure regular CW-structure on $\mathbb{R}^2$ is one where the 2-cells are all reguar hexagons with sides of unit length. Any digital image can be thresholded to produce a black-white image and this black-white image can naturally be regared as a finite pure cellular subcomplex of either of the two proposed CW-structures on $\mathbb{R}^2$. Analogously, thresholding can be used to represent 3-dimensional greyscale images as finite pure cellular subspaces of cubical or permutahedral CW-structures on $\mathbb{R}^3$, and to represent RGB colour photographs as analogous subcomplexes of $\mathbb{R}^5$.

In this section we list a few functions for performing basic operations on n-dimensional pure cubical and pure permutahedral finite subcomplexes M of $X = R^n$. We refer to M simply as a *pure complex*. In subsequent sections we demonstrate how these few functions on pure complexes allow for in-depth analysis of experimental data.

(ASIDE. The basic operations could equally well be implemented for other CW-decompositions of $X = \mathbb{R}^n$ such as the regular CW-decompositions arising as the tessellations by a fundamental domain of a Bieberbach group (=torsion free crytallographic group). Moreover, the basic operations could also be implemented for other manifolds such as an n-torus $X = S^1 \times S^1 \times \cdots \times S^1$ or n-sphere $X = S^n$ or for X the universal cover of some interesting hyperbolic 3-manifold. An example use of the ambient manifold $X = S^1 \times S^1 \times S^1$ could be for the construction of a cellular subspace recording the time of day, day of week and week of the year of crimes committed in a population.)

BASIC OPERATIONS RETURNING PURE COMPLEXES. (Function descriptions available here.)

- `PureCubicalComplex(binary array)`
- `PurePermutahedralComplex(binary array)`
- `ReadImageAsPureCubicalComplex(file,threshold)`
- `ReadImageSquenceAsPureCubicalComplex(file,threshold)`
- `PureComplexBoundary(M)`
- `PureComplexComplement(M)`
- `PureComplexRandomCell(M)`

- PureComplexThickened(M)
- ContractedComplex(M, optional subcomplex of M)
- ExpandedComplex(M, optional supercomplex of M)
- PureComplexUnion(M,N)
- PureComplexIntersection(M,N)
- PureComplexDifference(M,N)
- FiltrationTerm(F,n)

BASIC OPERATIONS RETURNING FILTERED PURE COMPLEXES.

- PureComplexThickeningFiltration(M,length)
- ReadImageAsFilteredPureCubicalComplex(file,length)

5.4 Digital image analysis and persistent homology

The following example reads in a digital image as a filtered pure cubical complexex. The filtration is obtained by thresholding at a sequence of uniformly spaced values on the greyscale range. The persistent homology of this filtered complex is calculated in degrees 0 and 1 and displayed as two barcodes.

Example

```
gap> file:=HapFile("image1.3.2.png");;
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,40);
Filtered pure cubical complex of dimension 2.
gap> P:=PersistentBettiNumbers(F,0);;
gap> BarCodeCompactDisplay(P);
```

Example

```
gap> P:=PersistentBettiNumbers(F,1);;
gap> BarCodeCompactDisplay(P);
```

The 20 persistent bars in the degree 0 barcode suggest that the image has 20 objects. The degree 1 barcode suggests that there are 14 (or possibly 17) holes in these 20 objects.

5.4.1 Naive example of image segmentation by automatic thresholding

Assuming that short bars and isolated points in the barcodes represent noise while long bars represent essential features, a "noiseless" representation of the image should correspond to a term in the filtration corresponding to a column in the barcode incident with all the long bars but incident with no short bars or isolated points. There is no noiseless term in the above filtration of length 40. However (in conjunction with the next subsection) the following commands confirm that the 64th term in the filtration of length 500 is such a term and display this term as a binary image.

Example
```
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,500);;
gap> Y:=FiltrationTerm(F,64);
Pure cubical complex of dimension 2.
gap> BettiNumber(Y,0);
20
gap> BettiNumber(Y,1);
14
gap> Display(Y);
```

5.4.2 Refining the filtration

The first filtration for the image has 40 terms. One may wish to investigate a filtration with more terms, say 500 terms, with a view to analysing, say, those 1-cycles that are born by term 25 of the filtration and that die between terms 50 and 60. The following commands produce the relevant barcode showing that there is precisely one such 1-cycle.

Example
```
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,500);;
gap> L:=[20,60,61,62,63,64,65,66,67,68,69,70];;
gap> T:=FiltrationTerms(F,L);;
gap> P0:=PersistentBettiNumbers(T,0);;
gap> BarCodeCompactDisplay(P0);
gap> P1:=PersistentBettiNumbers(T,1);;
gap> BarCodeCompactDisplay(P1);
```

β_0:
β_1:

5.4.3 Background to the data

The following image was used in the example.

5.5 A second example of digital image segmentation

In order to automatically count the number of coins in this picture

we can load the image as a filtered pure cubical complex F of filtration length 40 say, and observe the degree zero persistent Betti numbers to establish that the 28-th term or so of F seems to be 'noise free' in degree zero. We can then set M equal to the 28-th term of F and thicken M a couple of times say to remove any tiny holes it may have. We can then construct the complement C of M. Then we can construct a 'neighbourhood thickening' filtration T of C with say 50 consecutive thickenings. The degree one persistent barcode for T has 24 long bars, suggesting that the original picture consists of 24 coins.

Example
```
gap> F:=ReadImageAsFilteredPureCubicalComplex("my_coins.png",40);;
gap> M:=FiltrationTerm(F,24);;  #Chosen after viewing degree 0 barcode for F
gap> M:=PureComplexThickened(M);;
gap> M:=PureComplexThickened(M);;
```

```
gap> C:=PureComplexComplement(M);;
gap> T:=ThickeningFiltration(C,50);;
gap> P:=PersistentBettiNumbers(T,1);;
gap> BarCodeCompactDisplay(P);
```

The pure cubical complex `W:=PureComplexComplement(FiltrationTerm(T,25))` has the correct number of path components, namely 25, but its path components are very much subsets of the regions in the image corresponding to coins. The complex W can be thickened repeatedly, subject to no two path components being allowed to merge, in order to obtain a more realistic image segmentation with path components corresponding more closely to coins. This is done in the follow commands which use a makeshift function `Basins(L)` available here. The commands essentially implement a standard watershed segmentation algorithm but do so by using the language of filtered pure cubical complexes.

Example

```
gap> W:=PureComplexComplement(FiltrationTerm(T,25));;
gap> L:=[];;
gap> for i in [1..PathComponentOfPureComplex(W,0)] do
gap> P:=PathComponentOfPureComplex(W,i);;
gap> Q:=ThickeningFiltration(P,50,M);;
gap> Add(L,Q);;
gap> od;;

gap> B:=Basins(L);
gap> Display(B);
```

5.6 A third example of digital image segmentation

The following image is number 3096 in the BSDS500 database of images [MFTM01].

A common first step in segmenting such an image is to appropriately threshold the corresponding gradient image.

The following commands use the thresholded gradient image to produce an outline of the aeroplane. The outline is a pure cubical complex with one path component and with first Betti number equal to 1.

Example

```
gap> file:=Filename(DirectoriesPackageLibrary("HAP"),"../tutorial/images/3096b.jpg");;
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,30);;
gap> F:=ComplementOfFilteredPureCubicalComplex(F);;
gap> M:=FiltrationTerm(F,27);;  #Thickening chosen based on degree 0 barcode
gap> Display(M);;
gap> P:=List([1..BettiNumber(M,0)],n->PathComponentOfPureComplex(M,n));;
gap> P:=Filtered(P,m->Size(m)>10);;
gap> M:=P[1];;
gap> for m in P do
> M:=PureComplexUnion(M,m);;
> od;
gap> T:=ThickeningFiltration(M,50);;
gap> BettiNumber(FiltrationTerm(T,11),0);
```

```
1
gap> BettiNumber(FiltrationTerm(T,11),1);
1
gap> BettiNumber(FiltrationTerm(T,12),1);
0
gap> #Confirmation that 11-th filtration term has one hole and the 12-th term is contractible.
gap> C:=FiltrationTerm(T,11);;
gap> for n in Reversed([1..10]) do
> C:=ContractedComplex(C,FiltrationTerm(T,n));
> od;
gap> C:=PureComplexBoundary(PureComplexThickened(C));;
gap> H:=HomotopyEquivalentMinimalPureCubicalSubcomplex(FiltrationTerm(T,12),C);;
gap> B:=ContractedComplex(PureComplexBoundary(H));;
gap> Display(B);
```

5.7 Naive example of digital image contour extraction

The following greyscale image is available from the online appendix to the paper [CKL14].

The following commands produce a picture of contours from this image based on greyscale values. They also produce a picture of just the closed contours (the non-closed contours having been homotopy collapsed to points).

Example

```
gap> file:=Filename(DirectoriesPackageLibrary("HAP"),"../tutorial/images/circularGradient.png");;
gap> L:=[];;
gap> for n in [1..15] do
> M:=ReadImageAsPureCubicalComplex(file,n*30000);
> M:=PureComplexBoundary(M);;
> Add(L,M);
> od;;
gap> C:=L[1];;
gap> for n in [2..Length(L)] do C:=PureComplexUnion(C,L[n]); od;
gap> Display(C);
gap> Display(ContractedComplex(C));
```

Contours from the above greyscale image:

Closed contours from the above greyscale image:

Very similar results are obtained when applied to the file `circularGradientNoise.png`, containing noise, available from the online appendix to the paper [CKL14].

The number of distinct "light sources" in the image can be read from the countours. Alternatively, this number can be read directly from the barcode produced by the following commands.

Example

```
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,20);;
gap> P:=PersistentBettiNumbersAlt(F,1);;
gap> BarCodeCompactDisplay(P);
```

The seventeen bars in the barcode correspond to seventeen light sources. The length of a bar is a measure of the "persistence" of the corresponding light source. A long bar may initially represent a

cluster of several lights whose members may eventually be distinguished from each other as new bars (or persistent homology classes) are created.

Here the command `PersistentBettiNumbersAlt` has been used. This command is explained in the following section.

The follwowing commands use a watershed method to partition the digital image into regions, one region per light source. A makeshift function `Basins(L)`, available here, is called. (The efficiency of the example could be easily improved. For simplicity it uses generic commands which, in principle, can be applied to cubical or permutarhedral complexes of higher dimensions.)

Example

```
gap> file:=Filename(DirectoriesPackageLibrary("HAP"),"../tutorial/images/circularGradient.png");;
gap> F:=ReadImageAsFilteredPureCubicalComplex(file,20);;
gap> FF:=ComplementOfFilteredPureCubicalComplex(F);

gap> W:=(FiltrationTerm(FF,3));
gap> for n in [4..23] do
> L:=[];;
> for i in [1..PathComponentOfPureComplex(W,0)] do
>  P:=PathComponentOfPureComplex(W,i);;
>  Q:=ThickeningFiltration(P,150,FiltrationTerm(FF,n));;
>  Add(L,Q);;
> od;;
> W:=Basins(L);
> od;

gap> C:=PureComplexComplement(W);;
gap> T:=PureComplexThickened(C);; C:=ContractedComplex(T,C);;
gap> Display(C);
```

5.8 Alternative approaches to computing persistent homology

From any sequence $X_0 \subset X_1 \subset X_2 \subset \cdots \subset X_T$ of cellular spaces (such as pure cubical complexes, or cubical complexes, or simplicial complexes, or regular CW complexes) we can construct a filtered chain complex $C_*X_0 \subset C_*X_1 \subset C_*X_2 \subset \cdots C_*X_T$. The induced homology homomorphisms $H_n(C_*X_0,\mathbb{F}) \to H_n(C_*X_1,\mathbb{F}) \to H_n(C_*X_2,\mathbb{F}) \to \cdots \to H_n(C_*X_T,\mathbb{F})$ with coefficients in a field $\mathbb{F}$ can be computed by applying an appropriate sequence of elementary row operations to the boundary matrices in the chain complex $C_*X_T \otimes \mathbb{F}$; the boundary matrices are sparse and are best represented as such; the row operations need to be applied in a fashion that respects the filtration. This method is used in the above examples of persistent homology. The method is not practical when the number of cells in X_T is large.

An alternative approach is to construct an admissible discrete vector field on each term X_k in the filtration. For each vector field there is a non-regular CW-complex Y_k whose cells correspond to the critical cells in X_k and for which there is a homotopy equivalence $X_k \simeq Y_k$. For each k the composite homomorphism $H_n(C_*Y_k,\mathbb{F}) \xrightarrow{\cong} H_n(C_*X_k,\mathbb{F}) \to H_n(C_*X_{k+1},\mathbb{F}) \xrightarrow{\cong} H_n(C_*Y_{k+1},\mathbb{F})$ can be computed and the persistent homology can be derived from these homology homomorphisms. This method is implemented in the function `PersistentBettiNUmbersAlt(X,n,p)` where p is the characteristic of the field, n is the homology degree, and X can be a filtered pure cubical complex, or a filtered simplicial complex, or a filtered regular CW complex, or indeed a filtered chain complex (represented in sparse

form). This function incorporates the functions `ContractedFilteredPureCubicalComplex(X)` and `ContractedFilteredRegularComplex(X)` which respectively input a filtered pure cubical complex and filtered regular CW-complex and return a filtered complex of the same data type in which each term of the output filtration is a deformation retract of the corresponding term in the input filtration.

In this approach the vector fields on the various spaces X_k are completely independent and so the method lends itself to a degree of easy parallelism. This is not incorporated into the current implementation.

As an illustration we consider a synthetic data set S consisting of 3527 points sampled, with errors, from an 'unknown' manifold M in $\mathbb{R}^3$. From such a data set one can associate a 3-dimensional cubical complex X_0 consisting of one unit cube centred on each (suitably scaled) data point. A visualization of X_0 is shown below.

Given a pure cubical complex X_s we construct $X_{s+1} = X_s \cup \{\bar{e}^3_\lambda\}_{\lambda\in\Lambda}$ by adding to X_s each closed unit cube $\bar{e}^3_\lambda$ in $\mathbb{R}^3$ that intersects non-trivially with X_s. We construct the filtered cubical complex $X_* = \{X_i\}_{0\le i\le 19}$ and compute the persistence matrices β^{**}_d for $d = 0,1,2$ and for $\mathbb{Z}_2$ coefficients. The filtered complex X_* is quite large. In particular, the final space X_{19} in the filtration involves 1 092727 vertices, 3 246354 edges, 3 214836 faces of dimension 2 and 1 061208 faces of dimension 3. The usual matrix reduction approach to computing persistent Betti numbers would involve an appropriate row reduction of sparse matrices one of which has over 3 million rows and 3 million columns.

Example

```
gap> file:=HapFile("data247.txt");;
gap> Read(file);;
gap> F:=ThickeningFiltration(T,20);;
gap> P:=PersistentBettiNumbersAlt(F,[0,1,2]);;
gap> BarCodeCompactDisplay(P);
```

The barcodes suggest that the data points might have been sampled from a manifold with the homotopy type of a torus.

5.8.1 Non-trivial cup product

Of course, a wedge $S^2 \vee S^1 \vee S^1$ has the same homology as the torus $S^1 \times S^1$. By establishing that a 'noise free' model for our data points, say the 10-th term X_{10} in the filtration, has a non-trivial cup product $\cup: H^1(X_{10},\mathbb{Z}) \times H^1(X_{10},\mathbb{Z}) \to H^2(X_{10},\mathbb{Z})$ we can eliminate $S^2 \vee S^1 \vee S^1$ as a candidate from which the data was sampled.

Example

```
gap> X10:=RegularCWComplex(FiltrationTerm(F,10));;
gap> cup:=LowDimensionalCupProduct(X10);;
gap> cup([1,0],[0,1]);
[ 1 ]
```

5.8.2 Explicit homology generators

It could be desirable to obtain explicit representatives of the persistent homology generators that "persist" through a significant sequence of filtration terms. There are two such generators in degree 1 and one such generator in degree 2. The explicit representatives in degree n could consist of

an inclusion of pure cubical complexes $Y_n \subset X_{10}$ for which the incuced homology homomorphism $H_n(Y_n,\mathbb{Z}) \to H_n(X_{10},\mathbb{Z})$ is an isomorphism, and for which Y_n is minimal in the sense that its homotopy type changes if any one or more of its top dimensional cells are removed. Ideally the space Y_n should be "close to the original dataset" X_0. The following commands first construct an explicit degree 2 homology generator representative $Y_2 \subset X_{10}$ where Y_2 is homotopy equivalent to X_{10}. They then construct an explicit degree 1 homology generators representative $Y_1 \subset X_{10}$ where Y_1 is homotopy equivalent to a wedge of two circles. The final command displays the homology generators representative Y_1.

Example

```
gap> Y2:=FiltrationTerm(F,10);;
gap> for t in Reversed([1..9]) do
> Y2:=ContractedComplex(Y2,FiltrationTerm(F,t));
> od;
gap> Y2:=ContractedComplex(Y2);;

gap> Size(FiltrationTerm(F,10));
918881
gap> Size(Y2);
61618

gap> Y1:=PureComplexDifference(Y2,PureComplexRandomCell(Y2));;
gap> Y1:=ContractedComplex(Y1);;
gap> Size(Y1);
474
gap> Display(Y1);
```

5.9 Knotted proteins

The Protein Data Bank contains a wealth of data which can be investigated with respect to knottedness. Information on a particular protein can be downloaded as a .pdb file. Each protein consists of one or more chains of amino acids and the file gives 3-dimensional Euclidean coordinates of the atoms in amino acids. Each amino acid has a unique "alpha carbon" atom (labelled as "CA" in the pdb file). A simple 3-dimensional curve, the *protein backbone*, can be constructed through the sequence of alpha carbon atoms. Typically the ends of the protein backbone lie near the "surface" of the protein and can be joined by a path outside of the protein to obtain a simple closed curve in Euclidean 3-space.

The following command reads in the pdb file for the T.thermophilus 1V2X protein, which consists of a single chain of amino acids, and uses Asymptote software to produce an interactive visualization of its backbone. A path joining the end vertices of the backbone is displayed in blue.

Example

```
gap> file:=HapFile("data1V2X.pdb");;
gap> DisplayPDBfile(file);
```

The next command reads in the pdb file for the T.thermophilus 1V2X protein and represents it as a 3-dimensional pure cubical complex K. A resolution of $r = 5$ is chosen and this results in a representation as a subcomplex K of an ambient rectangular box of volume equal to $184 \times 186 \times 294$ unit cubes. The complex K should have the homotopy type of a circle and the protein backbone is a 1-dimenional curve that should lie in K. The final command displays K.

Example

```
gap> r:=5;;
gap> K:=ReadPDBfileAsPureCubicalComplex(file,r);;
gap> K:=ContractedComplex(K);;
gap> K!.properties;
[ [ "dimension", 3 ], [ "arraySize", [ 184, 186, 294 ] ] ]

gap> Display(K);
```

Next we create a filtered pure cubical complex by repeatedly thickening K. We perform 15 thickenings, each thickening being a term in the filtration. The β_1 barcode for the filtration is displayed. This barcode is a descriptor for the geometry of the protein. For current purposes it suffices to note that the first few terms of the filtration have first homology equal to that of a circle. This indicates that the Euclidean coordinates in the pdb file robustly determine some knot.

Example

```
gap> F:=ThickeningFiltration(K,15);;
gap> F:=FilteredPureCubicalComplexToCubicalComplex(F);;
gap> F:=FilteredCubicalComplexToFilteredRegularCWComplex(F);;
gap> P:=PersistentBettiNumbersAlt(F,1);;
gap> BarCodeCompactDisplay(P);
```

The next commands compute a presentation for the fundamental group $\pi_1(\mathbb{R}^3 \setminus K)$ and the Alexander polynomial for the knot. This is the same Alexander polynomial as for the trefoil knot. Also, Tietze transformations can be used to see that the fundamental group is the same as for the trefoil knot.

Example

```
gap> C:=PureComplexComplement(K);;
gap> C:=ContractedComplex(C);;
gap> G:=FundamentalGroup(C);;
gap> GeneratorsOfGroup(G);
[ f1, f2 ]
gap> RelatorsOfFpGroup(G);
[ f2*f1^-1*f2^-1*f1^-1*f2*f1 ]

gap> AlexanderPolynomial(G);
x_1^2-x_1+1
```

5.10 Random simplicial complexes

For a positive integer n and probability p we denote by $Y(n,p)$ the *Linial-Meshulam random simplicial 2-complex*. Its 1-skeleton is the complete graph on n vertices; each possible 2-simplex is included independently with probability p.

The following commands first compute the number h_i of non-trivial cyclic summands in $H_i(Y(100,p),\mathbb{Z})$ for a range of probabilities p and $i = 1,2$ and then produce a plot of h_i versus p. The plot for h_1 is red and the plot for h_2 is blue. A plot for the Euler characteristic $1-h_1+h_2$ is shown in green.

Example

```
gap> L:=[];;M:=[];;
gap> for p in [1..100] do
> K:=RegularCWComplex(RandomSimplicialTwoComplex(100,p/1000));;
> h1:=Length(Homology(K,1));;
> h2:=Length(Homology(K,2));;
> Add(L, [1.0*(p/1000),h1,"red"]);
> Add(L, [1.0*(p/1000),h2,"blue"]);
> Add(M, [1.0*(p/1000),1-h1+h2,"green"]);
> od;
gap> ScatterPlot(L);
gap> ScatterPlot(M);
```

From this plot it seems that there is a *phase change threshold* at around $p = 0.025$. An inspection of the first homology groups $H_1(Y(100,p),\mathbb{Z})$ shows that in most cases there is no torsion. However, around the threshold some of the complexes do have torsion in their first homology.

Similar commands for $Y(75,p)$ suggest a phase transition at around $p = 0.035$ in this case. The following commands compute $H_1(Y(75,p),\mathbb{Z})$ for 900 random 2-complexes with p in a small interval around 0.035 and, in each case where there is torsion, the torsion coefficients are stored in a list. The final command prints these lists -- all but one of which are of length 1. For example, there is one 2-dimensional simplicial complex on 75 vertices whose first homology contains the summand $\mathbb{Z}_{10787966187051680066516118257882312 8}$. The largest prime factor is 805552359079941450096902 63 occuring in the summand $\mathbb{Z}_{259837760616287294231081766978855}$.

Example

```
gap> torsion:=function(n,p)
> local H, Y;
> Y:=RegularCWComplex(RandomSimplicialTwoComplex(n,p));
> H:=Homology(Y,1);
> H:=Filtered(H,x->not x=0);
> return H;
> end;
function( n, p ) ... end


gap> L:=[];;for n in [73000..73900] do
> t:=torsion(75,n/2000000);
> if not t=[] then Add(L,t); fi;
> od;

gap> Display(L);
[ [                                      2 ],
  [                                     26 ],
  [     259837760616287294231081766978855 ],
  [                                      2 ],
  [                                      3 ],
  [                                      2 ],
  [          2761642698060127444812143568 ],
  [       26263552810109746637762733819 76 ],
  [                                      2 ],
  [                                      3 ],
```

```
[          331123827512648948194307853 50 ],
[                                    16 ],
[                                     4 ],
[                                     3 ],
[                                     2 ],
[                                     3 ],
[                                     2 ],
[      852349499991838889677631005909 77 ],
[                                     2 ],
[      246441961307858211078977186620 22 ],
[                                     2,                                    2 ],
[                                     2 ],
[           41664166288902564549298246 8 ],
[          41582773001875039168786970 816 ],
[                                     2 ],
[            75889883165411088431747730 ],
[          33523474091636554792305315165 ],
[  10787966187051680066516118257882312 8 ],
[           55882658144091195683417299 80 ],
[                                     2 ],
[           50014572492241158780150534 58 ],
[                                    10 ],
[                                    12 ],
[                                     2 ],
[                                     2 ],
[                                     3 ],
[           77578702434252469879717893 22 ],
[        81646488569932696734396134974 12 ],
[                                     2 ] ]
```

5.11 Computing homology of a clique complex (Vietoris-Rips complex)

Topological data analysis provides one motivation for wanting to compute the homology of a clique complex. Consider for instance the cloud of data points shown in Example 5.2.1. This data is a set S of 400 points in the plane. Let Γ be the graph with vertex set S and with two vertices joined by an edge if they lie within a Euclidean distance of 40 of each other. The clique complex $K = K(\Gamma)$ could be studied to see what it reveals about the data. The following commands construct K and show that it is a 23-dimensional simplicial complex consisting of a total of 36191976 simplices.

Example

```
gap> file:=HapFile("data134.txt");;
gap> Read(file);
gap> A:=VectorsToSymmetricMatrix(S,EuclideanApproximatedMetric);;
gap> threshold:=40;;
gap> grph:=SymmetricMatrixToGraph(A,threshold);;
gap> dimension_cap:=100;;
gap> K:=CliqueComplex(grph,dimension_cap);
Simplicial complex of dimension 23.
```

```
gap> Size(K);
36191976
```

The computation of the homology of this clique complex K is a challenge because of its size. If we are only interested in K up to homotopy then we could try to modify the graph Γ in such a way that the homotopy type of the clique complex is unchanged but the size of the clique complex is reduced. This is done in the following commands, producing a smaller 19-dimensional simplicial complex K with 4180652 simplices.

Example

```
gap> ContractGraph(grph);;

gap> dimension_cap:=100;;
gap> K:=CliqueComplex(grph,dimension_cap);
Simplicial complex of dimension 19.

gap> Size(K);
4180652
```

To compute the homology of K in degrees 0 to 5 say, we could represent K as a regular CW-complex Y and then compute the homology of Y as follows. The homology $H_n(K) = \mathbb{Z}$ for $n = 0, 1$ and $H_n(K) = 0$ for $n = 2, 3, 4, 5$ is consistent with the data having been sampled from a space with the homotopy type of a circle.

Example

```
gap> Y:=RegularCWComplex(K);
Regular CW-complex of dimension 19

gap> Homology(Y,0);
[ 0 ]
gap> Homology(Y,1);
[ 0 ]
gap> Homology(Y,2);
[  ]
gap> Homology(Y,3);
[  ]
gap> Homology(Y,4);
[  ]
gap> Homology(Y,5)
[  ]
```

Chapter 6

Group theoretic computations

6.1 Third homotopy group of a supsension of an Eilenberg-MacLane space

The following example uses the nonabelian tensor square of groups to compute the third homotopy group

$\pi_3(S(K(G,1))) = \mathbb{Z}^{30}$

of the suspension of the Eigenberg-MacLane space $K(G,1)$ for G the free nilpotent group of class 2 on four generators.

Example

```
gap> F:=FreeGroup(4);;G:=NilpotentQuotient(F,2);;
gap> ThirdHomotopyGroupOfSuspensionB(G);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0 ]
```

6.2 Representations of knot quandles

The following example constructs the finitely presented quandles associated to the granny knot and square knot, and then computes the number of quandle homomorphisms from these two finitely prresented quandles to the 17-th quandle in HAP's library of connected quandles of order 24. The number of homomorphisms differs between the two cases. The computation therefore establishes that the complement in $\mathbb{R}^3$ of the granny knot is not homeomorphic to the complement of the square knot.

Example

```
gap> Q:=ConnectedQuandle(24,17,"import");;
gap> K:=PureCubicalKnot(3,1);;
gap> L:=ReflectedCubicalKnot(K);;
gap> square:=KnotSum(K,L);;
gap> granny:=KnotSum(K,K);;
gap> gcsquare:=GaussCodeOfPureCubicalKnot(square);;
gap> gcgranny:=GaussCodeOfPureCubicalKnot(granny);;
gap> Qsquare:=PresentationKnotQuandle(gcsquare);;
gap> Qgranny:=PresentationKnotQuandle(gcgranny);;
gap> NumberOfHomomorphisms(Qsquare,Q);
408
```

```
gap> NumberOfHomomorphisms(Qgranny,Q);
24
```

The following commands compute a knot quandle directly from a pdf file containing the following hand-drawn image of the knot.

Example

```
gap>  gc:=ReadLinkImageAsGaussCode("myknot.pdf");
[ [ [ -2, 4, -1, 3, -3, 2, -4, 1 ] ], [ -1, -1, 1, -1 ] ]
gap> Q:=PresentationKnotQuandle(gc);
Quandle presentation of 4 generators and 4 relators.
```

6.3 Identifying knots

Low index subgrops of the knot group can be used to identify knots with few crossings. For instance, the following commands read in the following image of a knot and identify it as a sum of two trefoils. The commands determine the prime components only up to reflection, and so they don't distinguish between the granny and square knots.

Example

```
gap> gc:=ReadLinkImageAsGaussCode("myknot2.png");
[ [ [ -4, 7, -5, 4, -7, 5, -3, 6, -2, 3, 8, -8, -6, 2, 1, -1 ] ],
  [ 1, -1, -1, -1, -1, -1, -1, 1 ] ]
gap> IdentifyKnot(gc);;
PrimeKnot(3,1) + PrimeKnot(3,1)    modulo reflections of components.
```

6.4 Aspherical 2-complexes

The following example uses Polymake's linear programming routines to establish that the 2-complex associated to the group presentation $< x,y,z : xyx = yxy, yzy = zyz, xzx = zxz >$ is aspherical (that is, has contractible universal cover). The presentation is Tietze equivalent to the presentation used in the computer code, and the associated 2-complexes are thus homotopy equivalent.

Example

```
gap> F:=FreeGroup(6);;
gap> x:=F.1;;y:=F.2;;z:=F.3;;a:=F.4;;b:=F.5;;c:=F.6;;
gap> rels:=[a^-1*x*y, b^-1*y*z, c^-1*z*x, a*x*(y*a)^-1,
>    b*y*(z*b)^-1, c*z*(x*c)^-1];;
gap> Print(IsAspherical(F,rels),"\n");
Presentation is aspherical.

true
```

6.5 Group presentations and homotopical syzygies

Free resolutons for a group G are constructed in HAP as the cellular chain complex $R_* = C_*(\tilde{X})$ of the universal cover of some CW-complex $X = K(G,1)$. The 2-skeleton of X gives rise to a free

presentation for the group G. This presentation depends on a choice of maximal tree in the 1-skeleton of X in cases where X has more than one 0-cell. The attaching maps of 3-cells in X can be regarded as *homotopical syzygies* or van Kampen diagrams over the group presentation whose boundaries spell the trivial word.

The following example constructs four terms of a resolution for the free abelian group G on $n = 3$ generators, and then extracts the group presentation from the resolution as well as the unique homotopical syzygy. The syzygy is visualized in terms of its graph of edges, directed edges being coloured according to the corresponding group generator. (In this example the CW-complex $\tilde{X}$ is regular, but in cases where it is not the visualization may be a quotient of the 1-skeleton of the syzygy.)

Example

```
gap> n:=3;;c:=1;;
gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(n),c));;
gap> R:=ResolutionNilpotentGroup(G,4);;
gap> P:=PresentationOfResolution(R);;
gap> P.freeGroup;
<free group on the generators [ x, y, z ]>
gap> P.relators;
[ y^-1*x^-1*y*x, z^-1*x^-1*z*x, z^-1*y^-1*z*y ]
gap> IdentityAmongRelatorsDisplay(R,1);
```

This homotopical syzygy represents a relationship between the three relators $[x,y]$, $[x,z]$ and $[y,z]$ where $[x,y] = xyx^{-1}y^{-1}$. The syzygy can be thought of as a geometric relationship between commutators corresponding to the well-known Hall-Witt identity:

$$[\,[x,y],\,{}^{y}z\,]\;[\,[y,z],\,{}^{z}x\,]\;[\,[z,x],\,{}^{x}y\,] \;=\; 1\;.$$

The homotopical syzygy is special since in this example the edge directions and labels can be understood as specifying three homeomorphisms between pairs of faces. Viewing the syzygy as the boundary of the 3-ball, by using the homeomorphisms to identify the faces in each face pair we obtain a quotient CW-complex M involving one vertex, three edges, three 2-cells and one 3-cell. The cell structure on the quotient exists because, under the restrictions of homomorphisms to the edges, any cycle of edges retricts to the identity map on any given edge. The following result tells us that M is in fact a closed oriented compact 3-manifold.

THEOREM. [Seifert u. Threlfall, Topologie, p.208] *Let S^2 denote the boundary of the 3-ball B^3 and suppose that the sphere S^2 is given a regular CW-structure in which the faces are partitioned into a collection of face pairs. Suppose that for each face pair there is an orientation reversing homeomorphism between the two faces that sends edges to edges and vertices to vertices. Suppose that by using these homeomorphisms to identity face pairs we obtain a (not necessarily regular) CW-structure on the quotient M. Then M is a closed compact orientable manifold if and only if its Euler characteristic is $\chi(M) = 0$.*

The next commands construct a presentation and associated unique homotopical syzygy for the free nilpotent group of class $c = 2$ on $n = 2$ generators.

Example

```
gap> n:=2;;c:=2;;
gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(n),c));;
gap> R:=ResolutionNilpotentGroup(G,4);;
gap> P:=PresentationOfResolution(R);;
gap> P.freeGroup;
<free group on the generators [ x, y, z ]>
gap> P.relators;
```

```
[ z*x*y*x^-1*y^-1, z*x*z^-1*x^-1, z*y*z^-1*y^-1 ]
gap> IdentityAmongRelatorsDisplay(R,1);
```

The syzygy represents the following relationship between commutators (in a free group).
$[\,[x^{-1},y][x,y]\,,\,[y,x][y^{-1},x]y^{-1}\,]\,[\,[y,x][y^{-1},x]\,,\,x^{-1}\,] \;=\; 1$
Again, using the theorem of Seifert and Threlfall we see that the free nilpotent group of class two on two generators arises as the fundamental group of a closed compact orientable 3-manifold M.

6.6 Bogomolov multiplier

The Bogomolov multiplier of a group is an isoclinism invariant. Using this property, the following example shows that there are precisely three groups of order 243 with non-trivial Bogomolov multiplier. The groups in question are numbered 28, 29 and 30 in GAP's library of small groups of order 243.

Example

```
gap> L:=AllSmallGroups(3^5);;
gap> C:=IsoclinismClasses(L);;
gap> for c in C do
> if Length(BogomolovMultiplier(c[1]))>0 then
> Print(List(c,g->IdGroup(g)),"\n\n\n"); fi;
> od;
[ [ 243, 28 ], [ 243, 29 ], [ 243, 30 ] ]
```

6.7 Second group cohomology and group extensions

Any group extension $N \rightarrowtail E \twoheadrightarrow G$ gives rise to:

- an outer action $\alpha: G \to Out(N)$ of G on N.
- an action $G \to Aut(Z(N))$ of G on the centre of N, uniquely induced by the outer action α and the canonical action of $Out(N)$ on $Z(N)$.
- a "2-cocycle" $f: G \times G \to N$.

Any outer homomorphism $\alpha: G \to Out(N)$ gives rise to a cohomology class k in $H^3(G,Z(N))$. It was shown by Eilenberg and MacLane that the class k is trivial if and only if the outer action α arises from some group extension $N \rightarrowtail E \twoheadrightarrow G$. If k is trivial then there is a (non-canonical) bijection between the second cohomology group $H^2(G,Z(N))$ and Yoneda equivalence classes of extensions of G by N that are compatible with α.

FIRST EXAMPLE.

Consider the group $H = SmallGroup(64,134)$. Consider the normal subgroup $N = NormalSubgroups(G)[15]$ and quotient group $G = H/N$. We have $N = C_2 \times D_4$, $A = Z(N) = C_2 \times C_2$ and $G = C_2 \times C_2$.

Suppose we wish to classify all extensions $C_2 \times D_4 \rightarrowtail E \twoheadrightarrow C_2 \times C_2$ that induce the given outer action of G on N. The following commands show that, up to Yoneda equivalence, there are two such extensions.

Example

```
gap> H:=SmallGroup(64,134);;
gap> N:=NormalSubgroups(H)[15];;
gap> A:=Centre(GOuterGroup(H,N));;
gap> G:=ActingGroup(A);;
gap> R:=ResolutionFiniteGroup(G,3);;
gap> C:=HomToGModule(R,A);;
gap> Cohomology(C,2);
[ 2 ]
```

The following additional commands return a standard 2-cocycle $f : G \times G \to A = C_2 \times C_2$ corresponding to the non-trivial element in $H^2(G,A)$. The value $f(g,h)$ of the 2-cocycle is calculated for all 16 pairs $g,h \in G$.

Example

```
gap> CH:=CohomologyModule(C,2);;
gap> Elts:=Elements(ActedGroup(CH));
[ <identity> of ..., f1 ]

gap> x:=Elts[2];;
gap> c:=CH!.representativeCocycle(x);
Standard 2-cocycle

gap> f:=Mapping(c);;
gap> for g in G do for h in G do
> Print(f(g,h),"\n");
> od;
> od;
<identity> of ...
<identity> of ...
<identity> of ...
<identity> of ...
<identity> of ...
f6
<identity> of ...
f6
<identity> of ...
<identity> of ...
<identity> of ...
<identity> of ...
<identity> of ...
f6
<identity> of ...
f6
```

The following commands will then construct and identify all extensions of N by G corresponding to the given outer action of G on N.

Example

```
gap> H := SmallGroup(64,134);;
gap> N := NormalSubgroups(H)[15];;
gap> ON := GOuterGroup(H,N);;
```

```
gap> A := Centre(ON);;
gap> G:=ActingGroup(A);;
gap> R:=ResolutionFiniteGroup(G,3);;
gap> C:=HomToGModule(R,A);;
gap> CH:=CohomologyModule(C,2);;
gap> Elts:=Elements(ActedGroup(CH));;

gap> lst := List(Elts{[1..Length(Elts)]},x->CH!.representativeCocycle(x));;
gap> ccgrps := List(lst, x->CcGroup(ON, x));;
gap> #So ccgrps is a list of groups, each being an extension of G by N, corresponding
gap> #to the two elements in H^2(G,A).

gap> #The following command produces the GAP identification number for each group.
gap> L:=List(ccgrps,IdGroup);
[ [ 64, 134 ], [ 64, 135 ] ]
```

SECOND EXAMPLE

The following example illustrates how to construct a cohomology class k in $H^2(G,A)$ from a cocycle $f : G \times G \to A$, where $G = SL_2(\mathbb{Z}_4)$ and $A = \mathbb{Z}_8$ with trivial action.

Example

```
gap> #We'll construct G=SL(2,Z_4) as a permutation group.
gap> G:=SL(2,ZmodnZ(4));;
gap> G:=Image(IsomorphismPermGroup(G));;

gap> #We'll construct Z_8=Z/8Z as a G-outer group
gap> z_8:=Group((1,2,3,4,5,6,7,8));;
gap> Z_8:=TrivialGModuleAsGOuterGroup(G,z_8);;

gap> #We'll compute the group h=H^2(G,Z_8)
gap> R:=ResolutionFiniteGroup(G,3);;  #R is a free resolution
gap> C:=HomToGModule(R,Z_8);; # C is a chain complex
gap> H:=CohomologyModule(C,2);; #H is the second cohomology H^2(G,Z_8)
gap> h:=ActedGroup(H);; #h is the underlying group of H

gap> #We'll compute  cocycles c2, c5 for the second and fifth cohomology classs
gap> c2:=H!.representativeCocycle(Elements(h)[2]);
Standard 2-cocycle

gap> c5:=H!.representativeCocycle(Elements(h)[5]);
Standard 2-cocycle

gap> #Now we'll construct the cohomology classes C2, C5 in the group h corresponding to the cocyc
gap> C2:=CohomologyClass(H,c2);;
gap> C5:=CohomologyClass(H,c5);;

gap> #Finally, we'll show that C2, C5 are distinct cohomology classes, both of order 4.
gap> C2=C5;
false
gap> Order(C2);
4
gap> Order(C5);
```

```
4
```

6.8 Cocyclic groups: a convenient way of representing certain groups

GAP offers a number of data types for representing groups, including those of fp-groups (*finitely presented groups*), pc-groups (*power-conjugate presentated groups* for finite polycyclic groups), pcp-groups (*polycyclically presented groups* for finite and infinite polycyclic groups), permutation groups (for finite groups), and matrix groups over a field or ring. Each data type has its advantages and limitations.

Based on the definitions and examples in Section 6.7 the additional data type of a cc-group (*cocyclic group*) is provided in HAP. This can be used for a group E arising as a group extension $N \rightarrowtail E \twoheadrightarrow G$ and is a component object involving:

- `E!.Base` consisting of some representation of a group G.
- `E!.Fibre` consisting of some representation of a group N.
- `E!.OuterGroup` consisting of an outer action $\alpha: G \to Out(N)$ of G on N.
- `E!.Cocycle` consisting of a "2-cocycle" $f: G \times G \to N$.

The first example in Section 6.7 illustrates the construction of cc-groups for which both the base G and fibre N are finite pc-groups. That example extends to any scenario in which the base G is a group for which:

1. we can construct the first 3 degrees of a free $\mathbb{Z}G$-resolution C_*X.
2. we can construct the first 2 terms of a contracting homotopy $h_i: C_nX \to C_{n+1}X$ for $i = 0, 1$.
3. N is a group in which we can multiply elements effectively and for which we can determine the centre $Z(N)$ and outer automorphism group $Out(N)$.

As an illustration where the base group is a non-solvable finite group and the fibre is the infinite cyclic group, with base group acting trivially on the fibre, the following commands list up to Yoneda equivalence all central extensions $\mathbb{Z} \rightarrowtail E \twoheadrightarrow G$ for $G = A_5 : C_{16}$. The base group is a non-solvable semi-direct product of order 960 and thus none of the 16 extensions are polycyclic. The commands classify the extensions according to their integral homology in degrees ≤ 2, showing that there are precisely 5 such equivalence classes of extensions. Thus, there are at least 5 distinct isomorphism types among the 16 extensions. A presentation is constructed for the group corresponding to the sixteenth extension. The final command lists the orders of the 16 cohomology group elements corresponding to the 16 extensions. The 16th element has order 1, meaning that the sixteenth extension is the direct product $C_\infty \times A_5 : C_{16}$.

Example

```
gap> G:=SmallGroup(960,637);;
gap> StructureDescription(G);
"A5 : C16"

gap> N:=AbelianPcpGroup([0]);;
gap> N:=TrivialGModuleAsGOuterGroup(G,N);;
```

```
gap> R:=ResolutionFiniteGroup(G,3);;
gap> C:=HomToGModule(R,N);;
gap> CH:=CohomologyModule(C,2);;
gap> Elts:=Elements(ActedGroup(CH));;
gap> lst := List(Elts{[1..Length(Elts)]},x->CH!.representativeCocycle(x));;
gap> ccgrps := List(lst, x->CcGroup(N, x));;

gap> inv:=function(gg)
> local T;
> T:=ResolutionInfiniteCcGroup(gg,3);
> return List([1..2],i->Homology(TensorWithIntegers(T),i));
> end;;

gap> EquivClasses:=Classify(ccgrps,inv);
 [ <Cc-group of Size infinity>, <Cc-group of Size infinity>,
      <Cc-group of Size infinity>, <Cc-group of Size infinity>,
      <Cc-group of Size infinity>, <Cc-group of Size infinity>,
      <Cc-group of Size infinity>, <Cc-group of Size infinity> ],
  [ <Cc-group of Size infinity>, <Cc-group of Size infinity>,
      <Cc-group of Size infinity>, <Cc-group of Size infinity> ],
  [ <Cc-group of Size infinity>, <Cc-group of Size infinity> ],
  [ <Cc-group of Size infinity> ], [ <Cc-group of Size infinity> ] ]
gap> List(EquivClasses,Size);
[ 8, 4, 2, 1, 1 ]

gap> F16:=Image(IsomorphismFpGroup(ccgrps[16]));
<fp group on the generators [ x, y, z, w, v ]>
gap> RelatorsOfFpGroup(F16);
[ (x^2*y*z*w*z*y)^3*x^2*(y*x*w*y^2*z*x*y*z*y)^3*y*x*w*y^2*z*x*y^2*z*w*y^2*z*y,
  x*y^-2*w^-1*z^-1*y^-1*x^-1*y, x*z^-1*y^-1*z^-1*w^-1*z^-1*y^-1*x^-1*z,
  x*y^-2*z^-1*w^-1*z^-1*y^-1*x^-1*w, z^-2, w^-2, y^-3, w*y^-1*w^-1*y^-1,
  w*z*w^-1*z^-1*w^-1*z, z*y^2*(z^-1*y^-1)^2, v^-1*x^-1*v*x, v*y*v^-1*y^-1,
  v*z*v^-1*z^-1, v*w*v^-1*w^-1 ]

gap> List(Elts,Order);
[ 16, 16, 16, 16, 16, 16, 16, 16, 8, 8, 8, 8, 4, 4, 2, 1 ]
```

6.9 Effective group presentations

For any free $\mathbb{Z}G$-resolution $R_* = C_*X$ arising as the cellular chain complex of a contractible CW-complex, the terms in degrees ≤ 2 correspond to a free presentation for the group G. The following example accesses this presentation for the group $PGL_3(\mathbb{Z}[\sqrt{-1}])$.

Example

```
gap> K:=ContractibleGcomplex("PGL(3,Z[i])");;
gap> R:=FreeGResolution(K,2);;
gap> P:=PresentationOfResolution(R);;
gap> G:=P.freeGroup/P.relators;
<fp group on the generators [ v, w, x, y, z ]>
gap> P.relators;
[ v^2, w^-1*v*w*v^-1, w^-1*v^-1*w^-1, (x^-1*w)^3, (y^-1*w)^3, (z^-1*w)^4,
```

```
  y^-1*v^-1*z*y^-1*x, y^-1*v*x*v^-1*x*v, v^-1*z*v^-1*x*y, v^-1*x*v*y*v*x*v*y,
  x^3, x*z*y, y^-1*v^-1*y^2*v*y^-1, (v*y)^4, z^-1*y*v*z^-1, (v*y*z)^2,
  v^-1*(z*v)^2*z ]
```

The homomorphism $h_0: R_0 \to R_1$ of a contracting homotopy provides a unique expression for each element of G as a word in the free generators. To illustrate this, we consider the Sylow 2-subgroup $H = Syl_2(M_{24})$ of the Mathieu group M_{24}. We obtain a resolution R_* for H by recursively applying perturbation techniques to a composition series for H. Such a resolution will yield a "kind of" power-conjugate presentation for H.

Example

```
gap> H:=SylowSubgroup(MathieuGroup(24),2);
<permutation group of size 1024 with 10 generators>
gap> Order(H);
1024
gap> C:=CompositionSeries(H);;
gap> R:=ResolutionSubnormalSeries(C,2);;
gap> P:=PresentationOfResolution(R);;
gap> P.freeGroup/P.relators;
<fp group on the generators [ q, r, s, t, u, v, w, x, y, z ]>
gap> P.relators;
[ q^-2*z*y*x*w*v, q*r^-1*q^-1*y*u*r, s*q*s^-1*q^-1, t*q*t^-1*q^-1,
  q*u^-1*q^-1*y*v*u, y*q*v^-1*q^-1, q*w^-1*q^-1*z*x, w*q*x^-1*q^-1,
  q*y^-1*q^-1*z*v, z*q*z^-1*q^-1, r^-2, t*r*s^-1*r^-1, s*r*t^-1*r^-1,
  u*r*u^-1*r^-1, v*r*v^-1*r^-1, r*w^-1*r^-1*y*w*u, r*x^-1*r^-1*y*x*u,
  y*r*y^-1*r^-1, z*r*z^-1*r^-1, s^-2, t*s*t^-1*s^-1, x*s*u^-1*s^-1,
  s*v^-1*s^-1*z*y*w*u, s*w^-1*s^-1*y*v*u, u*s*x^-1*s^-1, s*y^-1*s^-1*y*x*u,
  z*s*z^-1*s^-1, t^-2, t*u^-1*t^-1*y*x*u, t*v^-1*t^-1*z*w, t*w^-1*t^-1*z*v,
  y*t*x^-1*t^-1, x*t*y^-1*t^-1, z*t*z^-1*t^-1, u^-2, v*u*v^-1*u^-1,
  u*w^-1*u^-1*z*w, x*u*x^-1*u^-1, y*u*y^-1*u^-1, z*u*z^-1*u^-1, v^-2,
  w*v*w^-1*v^-1, v*x^-1*v^-1*z*x, y*v*y^-1*v^-1, z*v*z^-1*v^-1, w^-2,
  x*w*x^-1*w^-1, w*y^-1*w^-1*z*y, z*w*z^-1*w^-1, x^-2, y*x*y^-1*x^-1,
  z*x*z^-1*x^-1, y^-2, z*y*z^-1*y^-1, z^-2 ]
```

The following additional commands use the contracting homotopy homomorphism $h_0: R_0 \to R_1$ to express some random elements of H as words in the free generators.

Example

```
gap> g:=Random(H);
(1,6)(2,3)(4,9)(5,16)(7,10)(8,21)(11,18)(12,17)(13,19)(14,20)(15,22)(23,24)
gap> P.wordInFreeGenerators(g);
q^-1*t^-1*x^-1*y^-1
gap>
gap> g:=Random(H);
(1,6)(2,23,10,18)(3,22,19,24)(4,11,15,9)(7,8,21,13)(12,14)
gap> P.wordInFreeGenerators(g);
q^-1*u^-1*w^-1*x^-1*z^-1
gap>
gap> g:=Random(H);
(1,14,5,17)(2,7,9,19)(3,11,4,22)(6,12,16,20)(8,18,24,15)(10,23,13,21)
gap> P.wordInFreeGenerators(g);
q^-1*r^-1*t^-1*v^-1*x^-1*z^-1
```

```
gap>
gap> g:=Random(H);
(1,14,5,17)(2,21)(3,9)(4,24)(6,12,16,20)(7,11,15,13)(8,23)(10,18,22,19)
gap> P.wordInFreeGenerators(g);
q^-1*r^-1*t^-1*v^-1*w^-1*z^-1
```

Because the resolution R_* was obtained from a composition series, the unique word associated to an element $g \in H$ always has the form $q^{\varepsilon_1} r^{\varepsilon_2} s^{\varepsilon_3} t^{\varepsilon_4} u^{\varepsilon_5} v^{\varepsilon_6} w^{\varepsilon_7} x^{\varepsilon_8} y^{\varepsilon_9} z^{\varepsilon_{10}}$ determined by the exponent vector $(\varepsilon_1, \cdots, \varepsilon_{10}) \in (\mathbb{Z}_2)^{10}$.

6.10 Second group cohomology and cocyclic Hadamard matrices

An *Hadamard matrix* is a square $n \times n$ matrix H whose entries are either $+1$ or -1 and whose rows are mutually orthogonal, that is $HH^t = nI_n$ where H^t denotes the transpose and I_n denotes the $n \times n$ identity matrix.

Given a group $G = \{g_1, g_2, \ldots, g_n\}$ of order n and the abelian group $A = \{1, -1\}$ of square roots of unity, any 2-cocycle $f : G \times G \to A$ corresponds to an $n \times n$ matrix $F = (f(g_i, g_j))_{1 \le i,j \le n}$ whose entries are ± 1. If F is Hadamard it is called a *cocyclic Hadamard matrix* corresponding to G.

The following commands compute all 192 of the cocyclic Hadamard matrices for the abelian group $G = \mathbb{Z}_4 \oplus \mathbb{Z}_4$ of order $n = 16$.

Example

```
gap> G:=AbelianGroup([4,4]);;
gap> F:=CocyclicHadamardMatrices(G);;
gap> Length(F);
192
```

6.11 Third group cohomology and homotopy 2-types

HOMOTOPY 2-TYPES

The third cohomology $H^3(G,A)$ of a group G with coefficients in a G-module A, together with the corresponding 3-cocycles, can be used to classify homotopy 2-types. A *homotopy 2-type* is a CW-complex whose homotopy groups are trivial in dimensions $n = 0$ and $n > 2$. There is an equivalence between the two categories

1. (Homotopy category of connected CW-complexes X with trivial homotopy groups $\pi_n(X)$ for $n > 2$)

2. (Localization of the category of simplicial groups with Moore complex of length 1, where localization is with respect to homomorphisms inducing isomorphisms on homotopy groups)

which reduces the homotopy theory of 2-types to a 'computable' algebraic theory. Furthermore, a simplicial group with Moore complex of length 1 can be represented by a group H endowed with two endomorphisms $s : H \to H$ and $t : H \to H$ satisfying the axioms

- $ss = s$, $ts = s$,
- $tt = t$, $st = t$,

- $[\ker s, \ker t] = 1$.

Ths triple (H,s,t) was termed a *cat*^{1}*-group* by J.-L. Loday since it can be regarded as a group H endowed with one compatible category structure.

The *homotopy groups* of a cat^1-group H are defined as: $\pi_1(H) = \text{image}(s)/t(\ker(s))$; $\pi_2(H) = \ker(s) \cap \ker(t)$; $\pi_n(H) = 0$ for $n > 2$ or $n = 0$. Note that $\pi_2(H)$ is a $\pi_1(H)$-module where the action is induced by conjugation in H.

A homotopy 2-type X can be represented by a cat^1-group H or by the homotopy groups $\pi_1 X = \pi_1 H$, $\pi_2 X = \pi_2 H$ and a cohomology class $k \in H^3(\pi_1 X, \pi_2 X)$. This class k is the *Postnikov invariant*.

RELATION TO GROUP THEORY

A number of standard group-theoretic constructions can be viewed naturally as a cat^1-group.

1. A $\mathbb{Z}G$-module A can be viewed as a cat^1-group (H,s,t) where H is the semi-direct product $A \rtimes G$ and $s(a,g) = (1,g)$, $t(a,g) = (1,g)$. Here $\pi_1(H) = G$ and $\pi_2(H) = A$.

2. A group G with normal subgroup N can be viewed as a cat^1-group (H,s,t) where H is the semi-direct product $N \rtimes G$ and $s(n,g) = (1,g)$, $t(n,g) = (1,ng)$. Here $\pi_1(H) = G/N$ and $\pi_2(H) = 0$.

3. The homomorphism $\iota : G \to Aut(G)$ which sends elements of a group G to the corresponding inner automorphism can be viewed as a cat^1-group (H,s,t) where H is the semi-direct product $G \rtimes Aut(G)$ and $s(g,a) = (1,a)$, $t(g,a) = (1,\iota(g)a)$. Here $\pi_1(H) = Out(G)$ is the outer automorphism group of G and $\pi_2(H) = Z(G)$ is the centre of G.

These three constructions are implemented in HAP.

EXAMPLE

The following commands begin by constructing the cat^1-group H of Construction 3 for the group $G = SmallGroup(64,134)$. They then construct the fundamental group of H and the second homotopy group of as a π_1-module. These homotopy groups have orders 8 and 2 respectively.

Example

```
gap> G:=SmallGroup(64,134);;
gap> H:=AutomorphismGroupAsCatOneGroup(G);;
gap> pi_1:=HomotopyGroup(H,1);;
gap> pi_2:=HomotopyModule(H,2);;
gap> Order(pi_1);
8
gap> Order(ActedGroup(pi_2));
2
```

The following additional commands show that there are 1024 Yoneda equivalence classes of cat^1-groups with fundamental group π_1 and π_1- module equal to π_2 in our example.

Example

```
gap> R:=ResolutionFiniteGroup(pi_1,4);;
gap> C:=HomToGModule(R,pi_2);;
gap> CH:=CohomologyModule(C,3);;
gap> AbelianInvariants(ActedGroup(CH));
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

A 3-cocycle $f:\pi_1\times\pi_1\times\pi_1\to\pi_2$ corresponding to a random cohomology class $k\in H^3(\pi_1,\pi_2)$ can be produced using the following command.

Example

```
gap> x:=Random(Elements(ActedGroup(CH)));;
gap> f:=CH!.representativeCocycle(x);
Standard 3-cocycle
```

The 3-cocycle corresponding to the Postnikov invariant of H itself can be easily constructed directly from its definition in terms of a set-theoretic 'section' of the crossed module corresponding to H.

Chapter 7

Cohomology of groups (and Lie Algebras)

7.1 Finite groups

7.1.1 Naive homology computation for a very small group

It is possible to compute the low degree (co)homology of a finite group or monoid of small order directly from the bar resolution. The following commands take this approach to computing the fifth integral homology

$H_5(Q_4,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$

of the quaternion group $G = Q_4$ of order 8.

Example

```
gap> Q:=QuaternionGroup(8);;
gap> B:=BarComplexOfMonoid(Q,6);;
gap> C:=ContractedComplex(B);;
gap> Homology(C,5);
[ 2, 2 ]


gap> List([0..6],B!.dimension);
[ 1, 7, 49, 343, 2401, 16807, 117649 ]
gap> List([0..6],C!.dimension);
[ 1, 2, 2, 1, 2, 4, 102945 ]
```

However, this approach is of limited applicability since the bar resolution involves $|G|^k$ free generators in degree k. A range of techniques, tailored to specific classes of groups, can be used to compute the (co)homology of larger finite groups.

This naive approach does have the merit of being applicable to arbitrary small monoids. The following calculates the homology in degrees ≤ 7 of a monoid of order 8, the monoid being specified by its multiplication table.

Example

```
gap> T:=[ [  1,  1,  1,  4,  4,  4,  4,  1 ],
>         [  1,  1,  1,  4,  4,  4,  4,  2 ],
>         [  1,  1,  1,  4,  4,  4,  4,  3 ],
>         [  4,  4,  4,  1,  1,  1,  1,  4 ],
>         [  4,  4,  4,  1,  1,  1,  1,  5 ],
>         [  4,  4,  4,  1,  1,  1,  1,  6 ],
```

```
>         [  4,  4,  4,  1,  1,  1,  1,  7 ],
>         [  1,  2,  3,  4,  5,  6,  7,  8 ] ];;
gap> M:=MonoidByMultiplicationTable(T);
<monoid of size 8, with 8 generators>
gap> B:=BarComplexOfMonoid(M,8);;
gap> C:=ContractedComplex(B);;
gap> List([0..7],i->Homology(C,i));
[ [ 0 ], [ 2 ], [  ], [ 2 ], [  ], [ 2 ], [  ], [ 2 ] ]


gap> List([0..8],B!.dimension);
[ 1, 7, 49, 343, 2401, 16807, 117649, 823543, 5764801 ]
gap> List([0..8],C!.dimension);
[ 1, 1, 1, 1, 1, 1, 1, 1, 5044201 ]
```

7.1.2 A more efficient homology computation

The following example computes the seventh integral homology
$H_7(M_{23},\mathbb{Z}) = \mathbb{Z}_{16} \oplus \mathbb{Z}_{15}$
and fourth integral cohomomogy
$H^4(M_{24},\mathbb{Z}) = \mathbb{Z}_{12}$
and fifth integral homology
$H_5(M_{24},\mathbb{Z}) = \mathbb{Z}_{14}$
of the Mathieu groups M_{23} and M_{24}. (Warning: the computation of $H_7(M_{23},\mathbb{Z})$ takes a couple of hours to run and the computation for $H_5(M_{24},\mathbb{Z})$ takes an hour to run.)

Example

```
gap> GroupHomology(MathieuGroup(23),7);
[ 16, 3, 5 ]

gap> GroupCohomology(MathieuGroup(24),4);
[ 4, 3 ]

gap> GroupHomology(MathieuGroup(24),5);
[ 2, 7 ]
```

7.1.3 Computation of an induced homology homomorphism

The following example computes the cokernel
$\mathrm{coker}(H_3(A_7,\mathbb{Z}) \to H_3(S_{10},\mathbb{Z})) \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2$
of the degree-3 integral homomogy homomorphism induced by the canonical inclusion $A_7 \to S_{10}$ of the alternating group on 7 letters into the symmetric group on 10 letters. The analogous cokernel with $\mathbb{Z}_2$ homology coefficients is also computed.

Example

```
gap> G:=SymmetricGroup(10);;
gap> H:=AlternatingGroup(7);;
gap> f:=GroupHomomorphismByFunction(H,G,x->x);;
gap> F:=GroupHomology(f,3);
MappingByFunction( Pcp-group with orders [ 4, 3 ], Pcp-group with orders
```

```
[ 2, 2, 4, 3 ], function( x ) ... end )
gap> AbelianInvariants(Range(F)/Image(F));
[ 2, 2 ]

gap> Fmod2:=GroupHomology(f,3,2);;
gap> AbelianInvariants(Range(Fmod2)/Image(Fmod2));
[ 2, 2 ]
```

7.1.4 Some other finite group homology computations

The following example computes the third integral homology of the Weyl group $W = Weyl(E_8)$, a group of order 696729600.

$H_3(Weyl(E_8),\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_{12}$

Example

```
gap> L:=SimpleLieAlgebra("E",8,Rationals);;
gap> W:=WeylGroup(RootSystem(L));;
gap> Order(W);
696729600
gap> GroupHomology(W,3);
[ 2, 2, 4, 3 ]
```

The preceding calculation could be achieved more quickly by noting that $W = Weyl(E_8)$ is a Coxeter group, and by using the associated Coxeter polytope. The following example uses this approach to compute the fourth integral homology of W. It begins by displaying the Coxeter diagram of W, and then computes

$H_4(Weyl(E_8),\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus Z_2 \oplus \mathbb{Z}_2$.

Example

```
gap> D:=[[1,[2,3]],[2,[3,3]],[3,[4,3],[5,3]],[5,[6,3]],[6,[7,3]],[7,[8,3]]];;
gap> CoxeterDiagramDisplay(D);
```

Example

```
gap> polytope:=CoxeterComplex_alt(D,5);;
gap> R:=FreeGResolution(polytope,5);
Resolution of length 5 in characteristic 0 for <matrix group with
8 generators> .
No contracting homotopy available.

gap> C:=TensorWithIntegers(R);
Chain complex of length 5 in characteristic 0 .

gap> Homology(C,4);
[ 2, 2, 2, 2 ]
```

The following example computes the sixth mod-2 homology of the Sylow 2-subgroup $Syl_2(M_{24})$ of the Mathieu group M_{24}.

$H_6(Syl_2(M_{24}),\mathbb{Z}_2) = \mathbb{Z}_2^{143}$

Example

```
gap> GroupHomology(SylowSubgroup(MathieuGroup(24),2),6,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

The following example computes the sixth mod-2 homology of the Unitary group $U_3(4)$ of order 312000.

$H_6(U_3(4),\mathbb{Z}_2) = \mathbb{Z}_2^4$

Example

```
gap> G:=GU(3,4);;
gap> Order(G);
312000
gap> GroupHomology(G,6,2);
[ 2, 2, 2, 2 ]
```

The following example constructs the Poincare series

$p(x) = \frac{1}{-x^3+3*x^2-3*x+1}$

for the cohomology $H^*(Syl_2(M_{12},\mathbb{F}_2)$. The coefficient of x^n in the expansion of $p(x)$ is equal to the dimension of the vector space $H^n(Syl_2(M_{12},\mathbb{F}_2)$. The computation involves SINGULAR's Groebner basis algorithms and the Lyndon-Hochschild-Serre spectral sequence.

Example

```
gap> G:=SylowSubgroup(MathieuGroup(12),2);;
gap> P:=PoincareSeriesLHS(G);
(1)/(-x_1^3+3*x_1^2-3*x_1+1)
```

The additional following command uses the Poincare series

Example

```
gap> RankHomologyPGroup(G,P,1000);
251000
```

to determine that $H_{1000}(Syl_2(M_{12},\mathbb{Z})$ is a direct sum of 251000 non-trivial cyclic 2-groups.

The following example constructs the series

$p(x) = \frac{x^4-x^3+x^2-x+1}{x^6-x^5+x^4-2*x^3+x^2-x+1}$

whose coefficient of x^n is equal to the dimension of the vector space $H^n(M_{11},\mathbb{F}_2)$ for all n in the range $0 \leq n \leq 14$. The coefficient is not guaranteed correct for $n \geq 15$.

Example

```
gap> PoincareSeriesPrimePart(MathieuGroup(11),2,14);
(x_1^4-x_1^3+x_1^2-x_1+1)/(x_1^6-x_1^5+x_1^4-2*x_1^3+x_1^2-x_1+1)
```

7.2 Nilpotent groups

The following example computes
$H_4(N,\mathbb{Z}) = (Z_3)^4 \oplus \mathbb{Z}^{84}$
for the free nilpotent group N of class 2 on four generators.

Example

```
gap> F:=FreeGroup(4);; N:=NilpotentQuotient(F,2);;
gap> GroupHomology(N,4);
[ 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

7.3 Crystallographic and Almost Crystallographic groups

The following example computes
$H_5(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$
for the 3-dimensional crystallographic space group G with Hermann-Mauguin symbol "P62"

Example

```
gap> GroupHomology(SpaceGroupBBNWZ("P62"),5);
[ 2, 2 ]
```

The following example computes
$H^5(G,\mathbb{Z}) = \mathbb{Z}$
for an almost crystallographic group.

Example

```
gap> G:=AlmostCrystallographicPcpGroup( 4, 50, [ 1, -4, 1, 2 ] );;
gap> GroupCohomology(G,4);
[ 0 ]
```

7.4 Arithmetic groups

The following example computes
$H_6(SL_2(\mathcal{O}),\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_{12}$
for $\mathcal{O}$ the ring of integers of the number field $\mathbb{Q}(\sqrt{-2})$.

Example

```
gap> C:=ContractibleGcomplex("SL(2,O-2)");;
gap> R:=FreeGResolution(C,7);;
gap> Homology(TensorWithIntegers(R),6);
[ 2, 12 ]
```

The following example computes

$$H_n(SL_4(\mathbb{Z}),\mathbb{Z}) = \begin{cases} 0, & n=1 \\ \mathbb{Z}_2 \oplus \mathbb{Z}_2, & n=2 \\ \mathbb{Z} \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_{12} \oplus \mathbb{Z}_{240}, & n=3 \\ \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_{10}, & n=4 \\ (\mathbb{Z}_2)^7, & n=5\,. \end{cases}$$

Example

```
gap> K:=ContractibleGcomplex("SL4Z_d");;
gap> R:=FreeGResolution(K,6);;
gap> Homology(TensorWithIntegers(R),1);
[  ]
gap> Homology(TensorWithIntegers(R),2);
[ 2, 2 ]
gap> Homology(TensorWithIntegers(R),3);
[ 2, 12, 240, 0 ]
gap> Homology(TensorWithIntegers(R),4);
[ 2, 2, 10 ]
gap> Homology(TensorWithIntegers(R),5);
[ 2, 2, 2, 2, 2, 2, 2 ]
```

For the congruence subgroup $\Gamma_0(101) < SL_3(\mathbb{Z})$ the following example computes

$$H_n(\Gamma_0(101),\mathbb{Z}) = \begin{cases} \mathbb{Z}_{100}, & n=1 \\ (\mathbb{Z}_2)^{15} \oplus (\mathbb{Z}_4)^4, & n=2 \\ (\mathbb{Z}_4)^2 \oplus (\mathbb{Z}_{12})^2 \oplus \mathbb{Z}^{16}, & n=3\,. \end{cases}$$

Example

```
gap> G:=HAPCongruenceSubgroupGamma0(3,101);;
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> R:=FreeGResolution(K,4);;
gap> S:=ResolutionFiniteSubgroup(R,G);;
gap> C:=TensorWithIntegers(S);;
gap> D:=ContractedComplex(C);;
gap> Homology(D,1);
[ 100 ]
gap> Homology(D,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4 ]
gap> Homology(D,3);
[ 4, 4, 12, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

In the preceding code K is a representation of a 3-dimensional contractible regular CW-complex (a 'well-rounded retract') on which $SL_3(\mathbb{Z})$ acts by permuting cells and for which every cell stabilizer is a finite group. The following code shows that there is just one orbit of 3-cells and displays the 1-skeleton of the closure of a 3-cell. (Further investigation of K shows that the 2-cells are either hexagonal or triangular. Each hexagonal 2-cell is incident with three 3-cells; each triangular 2-cell is incident with four 3-cells.)

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> List([0..4],K!.dimension);
[ 1, 1, 2, 1, 0 ]
```

```
gap> Y:=GComplexToFiniteRegularCWRegion(K,0);;
gap> Display(GraphOfRegularCWComplex(Y));
```

The command `Y:=GComplexToFiniteRegularCWRegion(K,r)` returns a 'ball of radius r' in the contractible CW-complex represented by K for any integer $r \geq 0$. The ball is returned as a finite regular CW-complex. For instance, the following code constructs a ball of radius $r = 7$ and builds a contracting disctete vector field on this ball. The component `Y!.G_Cells[n+1][k]` returns a pair $[e,g]$ with integer e specifying a representative of an orbit of n-cells in K and g an element in $SL(3,\mathbb{Z})$ such that the kth n-cell of Y corresponds to $g.e$.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> Y:=GComplexToFiniteRegularCWRegion(K,7);;
gap> Size(Y); #number of cells in Y
90537003
gap> Y!.nrCells(3); #number of 3-cells in Y
4729285
gap> CriticalCells(Y);
[ [ 0, 18456062 ] ]
```

Further details on computing with Arithmetic groups can be found in Chapters 13, 14 .

7.4.1 Rational cohomology

To compute the rational homology of an arithmetic group such as $G = SL_n(\mathbb{Z})$ one can work directly with the well-rounded retract K on which G acts with finite stabilizers. The following computes the rational cohomology

$$H^n(SL_4(\mathbb{Z}),\mathbb{Q}) = \begin{cases} \mathbb{Q} & n = 0,3 \\ 0 & \text{otherwise.} \end{cases}$$

Example

```
gap> K:=ContractibleGcomplex("SL(4,Z)");;
gap> C:=HomToRationals(K);;
gap> List([0..6],i->Cohomology(C,i));
[ 1, 0, 0, 1, 0, 0, 0 ]
```

For ν the virtual cohomological dimension of $G = SL_n(\mathbb{Z})$ there is an isomorphism

$H^{\nu-k}(SL_n(\mathbb{Z}),\mathbb{Z}) \cong H_k(SL_n(\mathbb{Z}),St_n \otimes_{\mathbb{Z}} \mathbb{Z})$

where St_n is the Steinberg module for G. This means that computations on the cohomology of G can be performed using a resolution V_* of the $\mathbb{Z}G$-module St_n.

Recall that in general a group G is said to be a duality group, of dimension $n \geq 0$, with respect to a right G-module S if there is an element $e \in H_n(G,S)$ such that the cap-product with e induces isomorphisms $H^k(G,A) \cong H_{n-k}(G,S \otimes A)$ for all integers k and all left G-modules A. A group G has a dualizing module S if $\mathbb{Z}$ admits a finite length projective $\mathbb{Z}G$- resolution R_* and two sufficient conditions hold: (a) $H^k(G,\mathbb{Z}G) = 0$ for $k \neq n$; (b) $H^n(G,\mathbb{Z}G)$ is torsion free. Then $S \cong H^n(G,\mathbb{Z}G)$ as G-modules. A resolution for S is given by $V_* = Hom_{\mathbb{Z}G}(R_*,\mathbb{Z}G)$.

If one is interested in computing only the rational cohomology of finite index subgoups of $SL_n(\mathbb{Z})$ then it suffices to use the non free chain complex $V_* = \mathrm{Hom}^c_{\mathbb{Z}}(C_*(K),\mathbb{Z})$ where K is a well-rounded retract and $C_*(K)$ its cellular chain complex and $\mathrm{Hom}^c_{\mathbb{Z}}(C_n(K),\mathbb{Z})$ is the group of those $\mathbb{Z}$-linear homomorphisms $C_n(K) \to \mathbb{Z}$ having finite support (i.e. trivial on all but finitely many basis elements).

For $G = SL_4(\mathbb{Z})$ we have $\nu = 6$. The following commands construct a non free V_* and compute $H_k(V_* \otimes_{\mathbb{Z}G} \mathbb{Q})$.

Example

```
gap> K:=ContractibleGcomplex("SL(4,Z)");;
gap> V:=DualComplex(K);;
gap> D:=TensorWithRationals(V);;
gap> List([0..6],i->Homology(D,i));
[ 0, 0, 0, 1, 0, 0, 1 ]
```

For $G = SL_3(\mathbb{Z})$ we have $\nu = 3$. The following commands use a resolution of the Steinberg module to compute the top rational cohomology

$H^3(\Gamma_0(501),\mathbb{Q}) \cong \mathbb{Q}^{140}$

for the congruence subgroup $\Gamma_0(501) < SL_3(\mathbb{Z})$. Furthermore, since any torsion element in $SL_3(\mathbb{Z})$ has order dividing 24, these commands also establish that the p-primary part of the third integral cohomology is

$H^3(\Gamma_0(501),\mathbb{Z})_{(7)} \cong \mathbb{Z}_7 \oplus \mathbb{Z}_7 \oplus \mathbb{Z}_7 \oplus \mathbb{Z}_7 \oplus \mathbb{Z}_7 \oplus \mathbb{Z}_7$,

$H^3(\Gamma_0(501),\mathbb{Z})_{(p)} \cong \mathbb{Z}_p \oplus \mathbb{Z}_p$ for primes $p = 7, 773, 191249, 81127067460894029326373198387$.

The p-part is trivial for all other primes $p > 3$. The computation does not determine the 2-part or 3-part.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> V:=DualComplex(K);;
gap> R:=FreeGResolution(V,1);; #first two terms of a free SL(3,Z)-resolution
gap> G:=HAPCongruenceSubgroupGamma0(3,501);;
gap> S:=ResolutionFiniteSubgroup(R,G);; #first two terms of a free G-resolution
gap> C:=TensorWithIntegersSparse(S);;
gap> for r in [1..200] do   #careful step-by-step reduction of the complex C
> C:=ContractedComplexViaChild(C,r);;
> od;

gap> Homology(C,0);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 14, 14, 14, 14,
  503725266059082802220487559469426575158,
  503725266059082802220487559469426575158, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0 ]

gap> Factors(503725266059082802220487559469426575158);
```

```
[ 2, 3, 7, 773, 191249, 81127067460894029326373198387 ]
```

The group $\Gamma_0(501)$ is of index 364741 in $SL_3(\mathbb{Z})$. The above computation uses `ContractedComplexViaChild(C,r)` which creates a child process, calls `ContractibleComplex(C,r)` in the child, and then kills the child. This provides a simple method for avoiding the build-up of unnecessarily retained data in memory caused by the current naive implementation of the function `ContractibleComplex(C,r)`. The initial version of the chain complex $C = C_*$ has $C_0 = \mathbb{Z}^{364741}$ while the quasi-isomorphic version of this complex after $r = 200$ reduction attempts has $C_0 = \mathbb{Z}^{741}$.

A similar computation gives:
$H^3(\Gamma_0(601),\mathbb{Q}) \cong \mathbb{Q}^{98}$,
$H^3(\Gamma_0(601),\mathbb{Z})_{(p)} \cong \mathbb{Z}_p \oplus \mathbb{Z}_p$ for primes $p = 37, 175829261, 101175716337972869988 5049667755467$,
$H^3(\Gamma_0(601),\mathbb{Z})_{(p)} \cong 0$ for all other primes $p > 3$.
Also
$H^3(\Gamma_0(701),\mathbb{Q}) \cong \mathbb{Q}^{116}$,
$H^3(\Gamma_0(701),\mathbb{Z})_{(p)} \cong \mathbb{Z}_p \oplus \mathbb{Z}_p$ for primes $p = 5, 7, 13, 113, 541, 23867527870278363301, 406535606513108747882$
$H^3(\Gamma_0(701),\mathbb{Z})_{(p)} \cong 0$ for all other primes $p > 3$.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> V:=DualComplex(K);;
gap> R:=FreeGResolution(V,1);; #first two terms of a free SL(3,Z)-resolution
gap> G:=HAPCongruenceSubgroupGamma0(3,601);;
gap> S:=ResolutionFiniteSubgroup(R,G);; #first two terms of a free G-resolution
gap> C:=TensorWithIntegersSparse(S);;
gap> for r in [1..200] do   #careful step-by-step reduction of the complex C
> C:=ContractedComplexViaChild(C,r);;
> od;

gap> Homology(C,0);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 8, 24,
  157972104741480396196975812410408490692059656,
  157972104741480396196975812410408490692059656, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

gap> Factors(157972104741480396196975812410408490692059656);
[ 2, 2, 2, 3, 37, 175829261, 101175716337972869988 5049667755467 ]
```

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> V:=DualComplex(K);;
```

```
gap> R:=FreeGResolution(V,1);; #first two terms of a free SL(3,Z)-resolution
gap> G:=HAPCongruenceSubgroupGamma0(3,701);;
gap> S:=ResolutionFiniteSubgroup(R,G);; #first two terms of a free G-resolution
gap> C:=TensorWithIntegersSparse(S);;
gap> for r in [1..200] do   #careful step-by-step reduction of the complex C
> C:=ContractedComplexViaChild(C,r);;
> od;

Homology(C,0);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6, 6, 6, 6, 6, 18, 18, 18, 18,
  145742727483325625191266021324259868511377454732522306398125478265 90,
  145742727483325625191266021324259868511377454732522306398125478265 90, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

gap> Factors(145742727483325625191266021324259868511377454732522306398125478265 90);
[ 2, 3, 3, 3, 5, 7, 13, 113, 541, 23867527870278363301,
  406535606513108747882997061430663240539 ]
```

7.4.2 Mod-q cohomology

For a congruence subgroup G of an arithmetic group such as $SL_n(\mathbb{Z})$ and positive integer q coprime to the order of any torsion element in G one can again work directly with the well-rounded retract K in order to compute $H^*(G,\mathbb{Z}/q\mathbb{Z})$.

For instance, for $G = \Gamma_0(11) < SL_3(\mathbb{Z})$ the following computes the cohomology ring $H^*(\Gamma_0(11),\mathbb{Z}/5\mathbb{Z})$ and shows it to be a 5-dimensional algebra with non-trivial product.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)s");;
gap> G:=HAPCongruenceSubgroupGamma0(3,11);;
gap> Y:=GComplexToRegularCWComplex(K,G);;
gap> Y:=SimplifiedComplex(Y);;
gap> R:=CohomologyRing(Y,5);
<algebra of dimension 5 over GF(5)>
gap> B:=CanonicalBasis(R);;
gap> List(B,R!.degree); #degrees of basis elements
[ 0, 1, 2, 3, 3 ]
gap> A:=NullMat(5,5);;
gap> for i in [1..5] do
>    for j in [1..5] do
>       A[i][j]:=B[i]*B[j];
>    od;
>    od;
gap> Display(A);
[ [    v.1,    v.2,    v.3,    v.4,    v.5 ],
  [    v.2,  0*v.1,    v.4,  0*v.1,  0*v.1 ],
```

```
[    v.3,    v.4,  0*v.1,  0*v.1,  0*v.1 ],
[    v.4,  0*v.1,  0*v.1,  0*v.1,  0*v.1 ],
[    v.5,  0*v.1,  0*v.1,  0*v.1,  0*v.1 ] ]
```

7.5 Artin groups

The following example computes

$$H_n(G,\mathbb{Z}) = \begin{cases} \mathbb{Z} & n = 0,1,7,8 \\ \mathbb{Z}_2, & n = 2,3 \\ \mathbb{Z}_2 \oplus \mathbb{Z}_6, & n = 4,6 \\ \mathbb{Z}_3 \oplus \mathbb{Z}_6, & n = 5 \\ 0, & n > 8 \end{cases}$$

for G the Artin group of type E_8. (Similar commands can be used to compute a resolution and homology of arbitrary Artin monoids and, in thoses cases such as the spherical cases where the $K(\pi,1)$-conjecture is known to hold, the homology is equal to that of the corresponding Artin group.)

Example

```
gap> D:=[[1,[2,3]],[2,[3,3]],[3,[4,3],[5,3]],[5,[6,3]],[6,[7,3]],[7,[8,3]]];;
gap> CoxeterDiagramDisplay(D);;
```

Example

```
gap> R:=ResolutionArtinGroup(D,9);;
gap> C:=TensorWithIntegers(R);;
gap> List([0..8],n->Homology(C,n));
[ [ 0 ], [ 0 ], [ 2 ], [ 2 ], [ 2, 6 ], [ 3, 6 ], [ 2, 6 ], [ 0 ], [ 0 ] ]
```

The Artin group G projects onto the Coxeter group W of type E_8. The group W has a natural representation as a group of 8×8 integer matrices. This projection gives rise to a representation $\rho: G \to GL_8(\mathbb{Z})$. The following command computes the cohomology group $H^6(G,\rho) = (\mathbb{Z}_2)^6$.

Example

```
gap> G:=R!.group;;
gap> gensG:=GeneratorsOfGroup(G);;
gap> W:=CoxeterDiagramMatCoxeterGroup(D);;
gap> gensW:=GeneratorsOfGroup(W);;
gap> rho:=GroupHomomorphismByImages(G,W,gensG,gensW);;
gap> C:=HomToIntegralModule(R,rho);;
gap> Cohomology(C,6);
[ 2, 2, 2, 2, 2, 2 ]
```

7.6 Graphs of groups

The following example computes

$H_5(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus Z_2 \oplus Z_2 \oplus Z_2 \oplus Z_2$

for G the graph of groups corresponding to the amalgamated product $G = S_5 *_{S_3} S_4$ of the symmetric groups S_5 and S_4 over the canonical subgroup S_3.

Example

```
gap> S5:=SymmetricGroup(5);SetName(S5,"S5");
gap> S4:=SymmetricGroup(4);SetName(S4,"S4");
gap> A:=SymmetricGroup(3);SetName(A,"S3");
gap> AS5:=GroupHomomorphismByFunction(A,S5,x->x);
gap> AS4:=GroupHomomorphismByFunction(A,S4,x->x);
gap> D:=[S5,S4,[AS5,AS4]];
gap> GraphOfGroupsDisplay(D);
```

Example

```
gap> R:=ResolutionGraphOfGroups(D,6);;
gap> Homology(TensorWithIntegers(R),5);
[ 2, 2, 2, 2, 2 ]
```

The next example computes the integral homology of $SL_2(\mathbb{Z})$ in degrees ≤ 5 using the isomorphism $SL_2(\mathbb{Z}) \cong C_4 *_{C_2} C_6$.

Example

```
gap> x:=(1,2,3,4,5,6,7,8,9,10,11,12);;
gap> C6:=Group(x^2);;SetName(C6,"C6");;
gap> C4:=Group(x^3);;SetName(C4,"C4");;
gap> C2:=Group(x^6);;SetName(C2,"C2");;
gap> C2C6:=GroupHomomorphismByFunction(C2,C6,x->x);;
gap> C2C4:=GroupHomomorphismByFunction(C2,C4,x->x);;
gap> D:=[C6,C4,[C2C6,C2C4]];;
gap> GraphOfGroupsDisplay(D);
```

Example

```
gap> R:=ResolutionGraphOfGroups(D,6);;
gap> List([0..5],n->Homology(TensorWithIntegers(R),n));
[ [ 0 ], [ 12 ], [  ], [ 12 ], [  ], [ 12 ] ]
```

The next commands compute the integral homology in degree 7 of a graph of groups corresponding to the HNN extension $G = H*_A$ where H is the symmetric group $H = S_5$ and A is the subgroup $A = S_3$ and $f : A \to H$ is the homomorphism $(1,2) \to (3,4)$, $(1,2,3) \to (3,4,5)$. The HNN extension G is obtained from H by adding a generator t subject to the relations $t^{-1}at = f(a)$ for $a \in A$.

Example

```
gap> S5:=SymmetricGroup(5);SetName(S5,"S5");;
gap> A:=SymmetricGroup(3);SetName(A,"S3");;
gap> f:=GroupHomomorphismByImages(A,S5,[(1,2),(1,2,3)],[(3,4),(3,4,5)]);;
gap> g:=GroupHomomorphismByFunction(A,S5,x->x);;
gap> D:=[S5,[f,g]];;
gap> GraphOfGroupsDisplay(D);;
```

Example

```
gap> R:=ResolutionGraphOfGroups(D,8);;
gap> Homology(TensorWithIntegers(R),7);
[ 2, 2, 60 ]
```

7.7 Lie algebra homology and free nilpotent groups

One method of producting a Lie algebra L from a group G is by forming the direct sum $L(G) = G/\gamma_2 G \oplus \gamma_2 G/\gamma_3 G \oplus \gamma_3 G/\gamma_4 G \oplus \cdots$ of the quotients of the lower central series $\gamma_1 G = G$, $\gamma_{n+1} G = [\gamma_n G, G]$. Commutation in G induces a Lie bracket $L(G) \times L(G) \to L(G)$.

The homology $H_n(L)$ of a Lie algebra (with trivial coefficients) can be calculated as the homology of the Chevalley-Eilenberg chain complex $C_*(L)$. This chain complex is implemented in HAP in the cases where the underlying additive group of L is either finitely generated torsion free or finitely generated of prime exponent p. In these two cases the ground ring for the Lie algebra/ Chevalley-Eilenberg complex is taken to be $\mathbb{Z}$ and $\mathbb{Z}_p$ respectively.

For example, consider the quotient $G = F/\gamma_8 F$ of the free group $F = F(x,y)$ on two generators by eighth term of its lower central series. So G is the *free nilpotent group of class 7 on two generators*. The following commands compute $H_4(L(G)) = \mathbb{Z}_2^{77} \oplus \mathbb{Z}_6^8 \oplus \mathbb{Z}_{12}^{51} \oplus \mathbb{Z}_{132}^{11} \oplus \mathbb{Z}^{2024}$ and show that the fourth homology in this case contains 2-, 3- and 11-torsion. (The commands take an hour or so to complete.)

Example

```
gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(2),7));;
gap> L:=LowerCentralSeriesLieAlgebra(G);;
gap> h:=LieAlgebraHomology(L,4);;
gap> Collected(h);
[ [ 0, 2024 ], [ 2, 77 ], [ 6, 8 ], [ 12, 51 ], [ 132, 11 ] ]
```

For a free nilpotent group G the additive homology $H_n(L(G))$ of the Lie algebra can be computed more quickly in HAP than the integral group homology $H_n(G,\mathbb{Z})$. Clearly there are isomorphisms$H_1(G) \cong H_1(L(G)) \cong G_{ab}$ of abelian groups in homological degree $n = 1$. Hopf's formula can be used to establish an isomorphism $H_2(G) \cong H_2(L(G))$ also in degree $n = 2$. The following two theorems provide further isomorphisms that allow for the homology of a free nilpotent group to be calculated more efficiently as the homology of the associated Lie algebra.

THEOREM 1. [KS98] *Let G be a finitely generated free nilpotent group of class 2. Then the integral group homology $H_n(G,\mathbb{Z})$ is isomorphic to the integral Lie algebra homology $H_n(L(G),\mathbb{Z})$ in each degree $n \geq 0$.*

THEOREM 2. [IO01] *Let G be a finitely generated free nilpotent group (of any class). Then the integral group homology $H_n(G,\mathbb{Z})$ is isomorphic to the integral Lie algebra homology $H_n(L(G),\mathbb{Z})$ in degrees $n = 0,1,2,3$.*

We should remark that experimentation on free nilpotent groups of class ≥ 4 has not yielded a group for which the isomorphism $H_n(G,\mathbb{Z}) \cong H_n(L(G),\mathbb{G})$ fails. For instance, the isomorphism holds in degree $n = 4$ for the free nilpotent group of class 5 on two generators, and for the free nilpotent group of class 2 on four generators:

Example

```
gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(2),5));;
gap> L:=LowerCentralSeriesLieAlgebra(G);;
gap> Collected( LieAlgebraHomology(L,4) );
[ [ 0, 85 ], [ 7, 1 ] ]
gap> Collected( GroupHomology(G,4) );
[ [ 0, 85 ], [ 7, 1 ] ]

gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(4),2));;
gap> L:=LowerCentralSeriesLieAlgebra(G);;
gap> Collected( LieAlgebraHomology(L,4) );
```

```
[ [ 0, 84 ], [ 3, 4 ] ]
gap> Collected( GroupHomology(G,4) );
[ [ 0, 84 ], [ 3, 4 ] ]
```

7.8 Cohomology with coefficients in a module

There are various ways to represent a $\mathbb{Z}G$-module A with action $G \times A \to A, (g,a) \mapsto \alpha(g,a)$.

One possibility is to use the data type of a *G-Outer Group* which involves three components: an *ActedGroup* A; an *ActingGroup* G; a *Mapping* $(g,a) \mapsto \alpha(g,a)$. The following example uses this data type to compute the cohomology $H^4(G,A) = \mathbb{Z}_5 \oplus \mathbb{Z}_{10}$ of the symmetric group $G = S_6$ with coefficients in the integers $A = \mathbb{Z}$ where odd permutations act non-trivially on A.

Example

```
gap> G:=SymmetricGroup(6);;

gap> A:=AbelianPcpGroup([0]);;
gap> alpha:=function(g,a); return a^SignPerm(g); end;;
gap> A:=GModuleAsGOuterGroup(G,A,alpha);
ZG-module with abelian invariants [ 0 ] and G= SymmetricGroup( [ 1 .. 6 ] )

gap> R:=ResolutionFiniteGroup(G,5);;
gap> C:=HomToGModule(R,A);
G-cocomplex of length 5 .

gap> Cohomology(C,4);
[ 2, 2, 5 ]
```

If $A = \mathbb{Z}^n$ and G acts as

$G \times A \to A, (g,v) \mapsto \rho(g)\,v$

where $\rho : G \to Gl_n(\mathbb{Z})$ is a (not necessarily faithful) matrix representation of degree n then we can avoid the use of G-outer groups and use just the homomorphism ρ instead. The following example uses this data type to compute the cohomology

$H^6(G,A) = \mathbb{Z}_2$

and the homology

$H_6(G,A) = 0$

of the alternating group $G = A_5$ with coefficients in $A = \mathbb{Z}^5$ where elements of G act on $\mathbb{Z}^5$ via an irreducible representation.

Example

```
gap> G:=AlternatingGroup(5);;
gap> rho:=IrreducibleRepresentations(G)[5];
[ (1,2,3,4,5), (3,4,5) ] ->
[
  [ [ 0, 0, 1, 0, 0 ], [ -1, -1, 0, 0, 1 ], [ 0, 1, 1, 1, 0 ],
      [ 1, 0, -1, 0, -1 ], [ -1, -1, 0, -1, 0 ] ],
  [ [ -1, -1, 0, 0, 1 ], [ 1, 0, -1, 0, -1 ], [ 0, 0, 0, 0, 1 ],
      [ 0, 0, 1, 0, 0 ], [ 0, 0, 0, 1, 0 ] ] ]
gap> R:=ResolutionFiniteGroup(G,7);;
gap> C:=HomToIntegralModule(R,rho);;
```

```
gap> Cohomology(C,6);
[ 2 ]
gap> D:=TensorWithIntegralModule(R,rho);
Chain complex of length 7 in characteristic 0 .

gap> Homology(D,6);
[  ]
```

If $V = K^d$ is a vetor space of dimension d over the field $K = GF(p)$ with p a prime and G acts on V via a homomorphism $\rho: G \to GL_d(K)$ then the homology $H^n(G,V)$ can again be computed without the use of G-outer groups. As an example, the following commands compute

$H^4(GL(3,2),V) = K^2$

where $K = GF(2)$ and $GL(3,2)$ acts with its natural action on $V = K^3$.

Example

```
gap> G:=GL(3,2);;
gap> rho:=GroupHomomorphismByFunction(G,G,x->x);;
gap> R:=ResolutionFiniteGroup(G,5);;
gap> C:=HomToModPModule(R,rho);;
gap> Cohomology(C,4);
2
```

The following example constructs the Artin braid group $G = B_3$ on three strings, the surjection $G \twoheadrightarrow S_3$, a complex representation $S_3 \to GL_2(\mathbb{C})$ and the composite representation $\rho: G \to GL_2(\mathbb{C})$. The group G acts on the vector space $V = \mathbb{C} \oplus \mathbb{C}$ via ρ. Then the dimensions of the cohomology groups $H^n(G,V)$ are computed.

Example

```
gap> D:=[[1,[2,3]],[2,[3,3]]];;
gap> R:=ResolutionArtinGroup(D,4);;
gap> G:=R!.group;; #Braid group
gap> r:=[G.1^2,G.2^2,G.3^2];;Q:=G/r;;
gap> q:=GroupHomomorphismByImages(G,Q,GeneratorsOfGroup(G),GeneratorsOfGroup(Q));;
gap> h:=IrreducibleRepresentations(Q)[3];
Pcgs([ f1^-1*f2^-1*f3^-2*f1^-1, f3^-1*f2^-1*f1^-1*f2^-1, f2*f3^-1*f1^-1*f2^-1,
  f3^-1*f1^-1 ]) -> [ [ [ 0, 1 ], [ 1, 0 ] ], [ [ E(3), 0 ], [ 0, E(3)^2 ] ],
  [ [ 1, 0 ], [ 0, 1 ] ], [ [ 1, 0 ], [ 0, 1 ] ] ]
gap> rho:=GroupHomomorphismByFunction(G,Image(h),x->Image(h,Image(q,x)));;
gap> C:=HomToModuleOverCyclotomicField(R,rho);
Cochain complex of length 4 over field of characteristic 0.

gap> Cohomology(C,1);
1
gap> Cohomology(C,2);
1
```

It can be computationally difficult to compute resolutions for large finite groups. But the p-primary part of the homology can be computed using resolutions of Sylow p-subgroups. This approach is used in the following example that computes the 2-primary part

$H_2(G,\mathbb{Z})_{(2)} = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$

of the degree 2 integral homology of the Rubik's cube group G. This group has order 43252003274489856000.

Example

```
gap> gens:= [
>    ( 1, 3, 8, 6)( 2, 5, 7, 4)( 9,33,25,17)(10,34,26,18)(11,35,27,19),
>    ( 9,11,16,14)(10,13,15,12)( 1,17,41,40)( 4,20,44,37)( 6,22,46,35),
>    (17,19,24,22)(18,21,23,20)( 6,25,43,16)( 7,28,42,13)( 8,30,41,11),
>    (25,27,32,30)(26,29,31,28)( 3,38,43,19)( 5,36,45,21)( 8,33,48,24),
>    (33,35,40,38)(34,37,39,36)( 3, 9,46,32)( 2,12,47,29)( 1,14,48,27),
>    (41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40)
>  ];; G:=Group(gens);;P:=SylowSubgroup(G,2);;
gap> R:=ResolutionNormalSeries(BigStepUCS(P,6),3);;
gap> PrimePartDerivedFunctorViaSubgroupChain(G,R,TensorWithIntegers,2);
[ 2, 2, 2 ]
```

The same approach is used in the following example that computes the 2-primary part

$H_{11}(A_7,A)_{(2)} = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$

of the degree 11 homology of the alternating group A_7 of degree 7 with coefficients in the module $A = \mathbb{Z}^7$ on which A_7 acts by permuting basis vectors.

Example

```
gap> G:=AlternatingGroup(7);;
gap> rho:=PermToMatrixGroup(G);;
gap> R:=ResolutionFiniteGroup(SylowSubgroup(G,2),12);;
gap> F:=function(X); return TensorWithIntegralModule(X,rho); end;;
gap> PrimePartDerivedFunctorViaSubgroupChain(G,R,F,11);
[ 2, 2, 4 ]
```

Similar commands compute

$H_3(A_{10},A)_{(2)} = \mathbb{Z}_4$

with coefficient module $A = \mathbb{Z}^{10}$ on which A_{10} acts by permuting basis vectors.

Example

```
gap> G:=AlternatingGroup(10);;
gap> rho:=PermToMatrixGroup(G);;
gap> R:=ResolutionFiniteGroup(SylowSubgroup(G,2),4);;
gap> F:=function(X); return TensorWithIntegralModule(X,rho); end;;
gap> PrimePartDerivedFunctorViaSubgroupChain(G,R,F,3);
[ 4 ]
```

The following commands compute

$H_{100}(GL(3,2),V) = K^{34}$

where V is the vector space of dimension 3 over $K = GF(2)$ acting via some irreducible representation $\rho: GL(3,2) \to GL(V)$.

Example

```
gap> G:=GL(3,2);;
gap> rho:=IrreducibleRepresentations(G,GF(2))[3];
CompositionMapping( [ (5,7)(6,8), (2,3,5)(4,7,6) ] ->
[ <an immutable 3x3 matrix over GF2>, <an immutable 3x3 matrix over GF2> ],
```

```
 <action isomorphism> )
gap> F:=function(X); return TensorWithModPModule(X,rho); end;;
gap> S:=ResolutionPrimePowerGroup(SylowSubgroup(G,2),101);;
gap> PrimePartDerivedFunctorViaSubgroupChain(G,S,F,100);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2 ]
```

7.9 Cohomology as a functor of the first variable

Suppose given a group homomorphism $f:G_1 \to G_2$ and a G_2-module A. Then A is naturally a G_1-module with action via f, and there is an induced cohomology homomorphism $H^n(f,A):H^n(G_2,A) \to H^n(G_1,A)$.

The following example computes this cohomology homomorphism in degree $n = 6$ for the inclusion $f:A_5 \to S_5$ and $A = \mathbb{Z}^5$ with action that permutes the canonical basis. The final commands determine that the kernel of the homomorphism $H^6(f,A)$ is the Klein group of order 4 and that the cokernel is cyclic of order 6.

Example

```
gap> G1:=AlternatingGroup(5);;
gap> G2:=SymmetricGroup(5);;
gap> f:=GroupHomomorphismByFunction(G1,G2,x->x);;
gap> pi:=PermToMatrixGroup(G2,5);;
gap> R1:=ResolutionFiniteGroup(G1,7);;
gap> R2:=ResolutionFiniteGroup(G2,7);;
gap> F:=EquivariantChainMap(R1,R2,f);;
gap> C:=HomToIntegralModule(F,pi);;
gap> c:=Cohomology(C,6);
[ g1, g2, g3 ] -> [ id, id, g3 ]

gap> AbelianInvariants(Kernel(c));
[ 2, 2 ]
gap> AbelianInvariants(Range(c)/Image(c));
[ 2, 3 ]
```

7.10 Cohomology as a functor of the second variable and the long exact coefficient sequence

A short exact sequence of $\mathbb{Z}G$-modules $A \rightarrowtail B \twoheadrightarrow C$ induces a long exact sequence of cohomology groups

$\to H^n(G,A) \to H^n(G,B) \to H^n(G,C) \to H^{n+1}(G,A) \to .$

Consider the symmetric group $G = S_4$ and the sequence $\mathbb{Z}_4 \rightarrowtail \mathbb{Z}_8 \twoheadrightarrow \mathbb{Z}_2$ of trivial $\mathbb{Z}G$-modules. The following commands compute the induced cohomology homomorphism

$f:H^3(S_4,\mathbb{Z}_4) \to H^3(S_4,\mathbb{Z}_8)$

and determine that the image of this induced homomorphism has order 8 and that its kernel has order 2.

Example

```
gap> G:=SymmetricGroup(4);;
gap> x:=(1,2,3,4,5,6,7,8);;
gap> a:=Group(x^2);;
gap> b:=Group(x);;
gap> ahomb:=GroupHomomorphismByFunction(a,b,y->y);;
gap> A:=TrivialGModuleAsGOuterGroup(G,a);;
gap> B:=TrivialGModuleAsGOuterGroup(G,b);;
gap> phi:=GOuterGroupHomomorphism();;
gap> phi!.Source:=A;;
gap> phi!.Target:=B;;
gap> phi!.Mapping:=ahomb;;

gap> Hphi:=CohomologyHomomorphism(phi,3);;

gap> Size(ImageOfGOuterGroupHomomorphism(Hphi));
8

gap> Size(KernelOfGOuterGroupHomomorphism(Hphi));
2
```

The following commands then compute the homomorphism
$H^3(S_4,\mathbb{Z}_8) \to H^3(S_4,\mathbb{Z}_2)$
induced by $\mathbb{Z}_4 \rightarrowtail \mathbb{Z}_8 \twoheadrightarrow \mathbb{Z}_2$, and determine that the kernel of this homomorphsim has order 8.

Example

```
gap> bhomc:=NaturalHomomorphismByNormalSubgroup(b,a);
gap> B:=TrivialGModuleAsGOuterGroup(G,b);
gap> C:=TrivialGModuleAsGOuterGroup(G,Image(bhomc));
gap> psi:=GOuterGroupHomomorphism();
gap> psi!.Source:=B;
gap> psi!.Target:=C;
gap> psi!.Mapping:=bhomc;

gap> Hpsi:=CohomologyHomomorphism(psi,3);

gap> Size(KernelOfGOuterGroupHomomorphism(Hpsi));
8
```

The following commands then compute the connecting homomorphism
$H^2(S_4,\mathbb{Z}_2) \to H^3(S_4,\mathbb{Z}_4)$
and determine that the image of this homomorphism has order 2.

Example

```
gap> delta:=ConnectingCohomologyHomomorphism(psi,2);;
gap> Size(ImageOfGOuterGroupHomomorphism(delta));
```

Note that the various orders are consistent with exactness of the sequence
$H^2(S_4,\mathbb{Z}_2) \to H^3(S_4,\mathbb{Z}_4) \to H^3(S_4,\mathbb{Z}_8) \to H^3(S_4,\mathbb{Z}_2)$.

7.11 Transfer Homomorphism

Consider the action of the symmetric group $G = S_5$ on $A = \mathbb{Z}^5$ which permutes the canonical basis. The action restricts to the sylow 2-subgroup $P = Syl_2(G)$. The following commands compute the cohomology transfer homomorphism $t^4: H^4(P,A) \to H^4(S_5,A)$ and determine its kernel and image. The integral homology transfer $t_4: H_4(S_5,\mathbb{Z}) \to H_5(P,\mathbb{Z})$ is also computed.

Example

```
gap> G:=SymmetricGroup(5);;
gap> P:=SylowSubgroup(G,2);;
gap> R:=ResolutionFiniteGroup(G,5);;
gap> A:=PermToMatrixGroup(G);;
gap> tr:=TransferCochainMap(R,P,A);
Cochain Map between complexes of length 5 .

gap> t4:=Cohomology(tr,4);
[ g1, g2, g3, g4 ] -> [ id, g1, g2, g4 ]
gap> StructureDescription(Kernel(t4));
"C2 x C2"
gap> StructureDescription(Image(t4));
"C4 x C2"

gap> tr:=TransferChainMap(R,P);
Chain Map between complexes of length 5 .

gap> Homology(tr,4);
[ g1 ] -> [ g1 ]
```

7.12 Cohomology rings of finite fundamental groups of 3-manifolds

A *spherical 3-manifold* is a 3-manifold arising as the quotient S^3/Γ of the 3-sphere S^3 by a finite subgroup Γ of $SO(4)$ acting freely as rotations. The geometrization conjecture, proved by Grigori Perelman, implies that every closed connected 3-manifold with a finite fundamental group is homeomorphic to a spherical 3-manifold.

A spherical 3-manifold S^3/Γ has finite fundamental group isomorphic to Γ. This fundamental group is one of:

- $\Gamma = C_m = \langle x \mid x^m \rangle$ (CYCLIC FUNDAMENTAL GROUP)
- $\Gamma = C_m \times \langle x,y \mid xyx^{-1} = y^{-1}, x^{2^k} = y^n \rangle$ for integers $k,m \geq 1, n \geq 2$ and m coprime to $2n$ (PRISM MANIFOLD CASE)
- $\Gamma = C_m \times \langle x,y,z \mid (xy)^2 = x^2 = y^2, zxz^{-1} = y, zyz^{-1} = xy, z^{3^k} = 1 \rangle$ for integers $k,m \geq 1$ and m coprime to 6 (TETRAHEDRAL CASE)
- $\Gamma = C_m \times \langle x,y \mid (xy)^2 = x^3 = y^4 \rangle$ for $m \geq 1$ coprime to 6 (OCTAHEDRAL CASE)
- $\Gamma = C_m \times \langle x,y \mid (xy)^2 = x^3 = y^5 \rangle$ for $m \geq 1$ coprime to 30 (ICOSAHEDRAL CASE).

This list of cases is taken from the Wikipedia pages. The group Γ has periodic cohomology since it acts on a sphere. The cyclic group has period 2 and in the other four cases it has period 4. (Recall that

in general a finite group G has *periodic cohomology of period n* if there is an element $u \in H^n(G,\mathbb{Z})$ such that the cup product $- \cup u : H^k(G,\mathbb{Z}) \to H^{k+n}(G,\mathbb{Z})$ is an isomorphism for all $k \geq 1$. It can be shown that G has periodic cohomology of period n if and only if $H^n(G,\mathbb{Z}) = \mathbb{Z}_{|G|}$.)

The cohomology of the cyclic group is well-known, and the cohomology of a direct product can be obtained from that of the factors using the Kunneth formula.

In the icosahedral case with $m = 1$ the following commands yield $$H^\ast(\Gamma,\mathbb Z)=Z[t]/(120t=0)$$ with generator t of degree 4. The final command demonstrates that a periodic resolution is used in the computation.

Example

```
gap> F:=FreeGroup(2);;x:=F.1;;y:=F.2;;
gap> G:=F/[(x*y)^2*x^-3, x^3*y^-5];;
gap> Order(G);
120
gap> R:=ResolutionSmallGroup(G,5);;
gap> n:=0;;Cohomology(HomToIntegers(R),n);
[ 0 ]
gap> n:=1;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=2;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=3;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=4;;Cohomology(HomToIntegers(R),n);
[ 120 ]

gap> List([0..5],k->R!.dimension(k));
[ 1, 2, 2, 1, 1, 2 ]
```

In the octahedral case with $m = 1$ we obtain $$H^\ast(\Gamma,\mathbb Z) = \mathbb Z[s,t]/(s^2=24t, 2s=0, 48t=0)$$ where s has degree 2 and t has degree 4, from the following commands.

Example

```
gap> F:=FreeGroup(2);;x:=F.1;;y:=F.2;;
gap> G:=F/[(x*y)^2*x^-3, x^3*y^-4];;
gap> Order(G);
48
gap> R:=ResolutionFiniteGroup(G,5);;
gap> n:=0;;Cohomology(HomToIntegers(R),n);
[ 0 ]
gap> n:=1;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=2;;Cohomology(HomToIntegers(R),n);
[ 2 ]
gap> n:=3;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=4;;Cohomology(HomToIntegers(R),n);
[ 48 ]
gap> IntegralCupProduct(R,[1],[1],2,2);
[ 24 ]
```

In the tetrahedral case with $m = 1$ we obtain $$H^\ast(\Gamma,\mathbb Z) = \mathbb Z[s,t]/(s^2=16t, 3s=0, 24t=0)$$ where s has degree 2 and t has degree 4, from the following commands.

Example

```
gap> F:=FreeGroup(3);;x:=F.1;;y:=F.2;;z:=F.3;;
gap> G:=F/[(x*y)^2*x^-2, x^2*y^-2, z*x*z^-1*y^-1, z*y*z^-1*y^-1*x^-1,z^3];;
gap> Order(G);
24
gap> R:=ResolutionFiniteGroup(G,5);;
gap> n:=1;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=2;;Cohomology(HomToIntegers(R),n);
[ 3 ]
gap> n:=3;;Cohomology(HomToIntegers(R),n);
[  ]
gap> n:=4;;Cohomology(HomToIntegers(R),n);
[ 24 ]
gap> IntegralCupProduct(R,[1],[1],2,2);
[ 16 ]
```

A theoretical calculation of the integral and mod-p cohomology rings of all of these fundamental groups of spherical 3-manifolds is given in [TZ08].

7.13 Explicit cocycles

Given a $\mathbb{Z}G$-resolution R_* and a $\mathbb{Z}G$-module A, one defines an *n-cocycle* to be a $\mathbb{Z}G$-homomorphism $f: R_n \to A$ for which the composite homomorphism $fd_{n+1}: R_{n+1} \to A$ is zero. If R_* happens to be the standard bar resolution (i.e. the cellular chain complex of the nerve of the group G considered as a one object category) then the free $\mathbb{Z}G$-generators of R_n are indexed by n-tuples $(g_1|g_2|\dots|g_n)$ of elements g_i in G. In this case we say that the n-cocycle is a *standard n-cocycle* and we think of it as a set-theoretic function

$f: G \times G \times \cdots \times G \longrightarrow A$

satisfying a certain algebraic cocycle condition. Bearing in mind that a standard n-cocycle really just assigns an element $f(g_1,\dots,g_n) \in A$ to an n-simplex in the nerve of G , the cocycle condition is a very natural one which states that *f must vanish on the boundary of a certain $(n+1)$-simplex*. For $n = 2$ the condition is that a 2-cocycle $f(g_1,g_2)$ must satisfy

$g.f(h,k) + f(g,hk) = f(gh,k) + f(g,h)$

for all $g,h,k \in G$. This equation is explained by the following picture.

The definition of a cocycle clearly depends on the choice of $\mathbb{Z}G$-resolution R_*. However, the cohomology group $H^n(G,A)$, which is a group of equivalence classes of n-cocycles, is independent of the choice of R_*.

There are some occasions when one needs explicit examples of standard cocycles. For instance:

- Let G be a finite group and k a field of characteristic 0. The group algebra $k(G)$, and the algebra $F(G)$ of functions $d_g: G \to k, h \to d_{g,h}$, are both Hopf algebras. The tensor product $F(G) \otimes k(G)$ also admits a Hopf algebra structure known as the quantum double $D(G)$. A twisted quantum double $D_f(G)$ was introduced by R. Dijkraaf, V. Pasquier & P. Roche [DPR91]. The twisted double is a quasi-Hopf algebra depending on a 3-cocycle $f: G \times G \times G \to k$. The

multiplication is given by $(d_g \otimes x)(d_h \otimes y) = d_{gx,xh}\beta_g(x,y)(d_g \otimes xy)$ where β_a is defined by $\beta_a(h,g) = f(a,h,g)f(h,h^{-1}ah,g)^{-1}f(h,g,(hg)^{-1}ahg)$. Although the algebraic structure of $D_f(G)$ depends very much on the particular 3-cocycle f, representation-theoretic properties of $D_f(G)$ depend only on the cohomology class of f.

- An explicit 2-cocycle $f: G \times G \to A$ is needed to construct the multiplication $(a,g)(a',g') = (a + g \cdot a' + f(g,g'), gg')$ in the extension a group G by a $\mathbb{Z}G$-module A determined by the cohomology class of f in $H^2(G,A)$. See 6.7.

- In work on coding theory and Hadamard matrices a number of papers have investigated square matrices (a_{ij}) whose entries $a_{ij} = f(g_i, g_j)$ are the values of a 2-cocycle $f: G \times G \to \mathbb{Z}_2$ where G is a finite group acting trivially on $\mathbb{Z}_2$. See for instance [Hor00] and 6.10.

Given a $\mathbb{Z}G$-resolution R_* (with contracting homotopy) and a $\mathbb{Z}G$-module A one can use HAP commands to compute explicit standard n-cocycles $f: G^n \to A$. With the twisted quantum double in mind, we illustrate the computation for $n = 3$, $G = S_3$, and $A = U(1)$ the group of complex numbers of modulus 1 with trivial G-action.

We first compute a $\mathbb{Z}G$-resolution R_*. The Universal Coefficient Theorem gives an isomorphism $H_3(G,U(1)) = Hom_{\mathbb{Z}}(H_3(G,\mathbb{Z}),U(1))$, The multiplicative group $U(1)$ can thus be viewed as $\mathbb{Z}_m$ where m is a multiple of the exponent of $H_3(G,\mathbb{Z})$.

Example

```
gap> G:=SymmetricGroup(3);;
gap> R:=ResolutionFiniteGroup(G,4);;
gap> TR:=TensorWithIntegers(R);;
gap> Homology(TR,3);
[ 6 ]
gap> R!.dimension(3);
4
gap> R!.dimension(4);
5
```

We thus replace the very infinite group U(1) by the finite cyclic group $\mathbb{Z}_6$. Since the resolution R_* has 4 generators in degree 3, a homomorphism $f: R^3 \to U(1)$ can be represented by a list $f = [f_1, f_2, f_3, f_4]$ with f_i the image in $\mathbb{Z}_6$ of the ith generator. The cocycle condition on f can be expressed as a matrix equation

$Mf^t = 0 \bmod 6$.

where the matrix M is obtained from the following command and f^t denotes the transpose.

Example

```
gap> M:=CocycleCondition(R,3);;
```

A particular cocycle $f = [f_1, f_2, f_3, f_4]$ can be obtained by choosing a solution to the equation Mf^t=0.

Example

```
gap> SolutionsMod2:=NullspaceModQ(TransposedMat(M),2);
[ [ 0, 0, 0, 0 ], [ 0, 0, 1, 1 ], [ 1, 1, 0, 0 ], [ 1, 1, 1, 1 ] ]

gap> SolutionsMod3:=NullspaceModQ(TransposedMat(M),3);
[ [ 0, 0, 0, 0 ], [ 0, 0, 0, 1 ], [ 0, 0, 0, 2 ], [ 0, 0, 1, 0 ],
  [ 0, 0, 1, 1 ], [ 0, 0, 1, 2 ], [ 0, 0, 2, 0 ], [ 0, 0, 2, 1 ],
```

```
    [ 0, 0, 2, 2 ] ]
```

A non-standard 3-cocycle f can be converted to a standard one using the command `StandardCocycle(R,f,n,q)` . This command inputs R_*, integers n and q, and an n-cocycle f for the resolution R_*. It returns a standard cocycle $G^n \to \mathbb{Z}_q$.

Example

```
gap> f:=3*SolutionsMod2[3] - SolutionsMod3[5];   #An example solution to Mf=0 mod 6.
[ 3, 3, -1, -1 ]

gap> Standard_f:=StandardCocycle(R,f,3,6);;

gap> g:=Random(G); h:=Random(G); k:=Random(G);
(1,2)
(1,3,2)
(1,3)

gap> Standard_f(g,h,k);
3
```

A function $f: G \times G \times G \to A$ is a standard 3-cocycle if and only if

$$g \cdot f(h,k,l) - f(gh,k,l) + f(g,hk,l) - f(g,h,kl) + f(g,h,k) = 0$$

for all $g,h,k,l \in G$. In the above example the group $G = S_3$ acts trivially on $A = Z_6$. The following commands show that the standard 3-cocycle produced in the example really does satisfy this 3-cocycle condition.

Example

```
gap> sf:=Standard_f;;

gap> Test:=function(g,h,k,l);
> return sf(h,k,l) - sf(g*h,k,l) + sf(g,h*k,l) - sf(g,h,k*l) + sf(g,h,k);
> end;
function( g, h, k, l ) ... end

gap> for g in G do for h in G do for k in G do for l in G do
> Print(Test(g,h,k,l),",");
> od;od;od;od;
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,6,6,0,0,6,
0,0,0,0,0,6,6,6,0,6,0,12,12,6,12,6,0,12,6,0,6,6,0,0,0,0,0,0,0,12,12,6,6,6,0,
6,6,0,6,6,0,0,-6,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,0,0,0,0,0,0,0,6,0,0,6,6,0,6,6,
0,6,0,0,6,6,6,0,0,0,0,0,0,0,-6,0,0,-6,0,-6,0,0,0,0,0,0,0,0,6,6,0,6,0,0,6,0,0,
0,0,0,6,6,6,0,0,0,6,6,6,0,0,0,0,-6,0,6,6,0,0,0,0,0,0,0,12,6,6,0,6,0,0,0,0,12,
6,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,6,0,0,6,0,0,0,0,0,6,6,
6,0,0,0,6,12,6,6,0,0,0,-6,0,0,6,0,0,0,0,0,0,0,12,12,6,6,6,0,0,0,0,6,6,0,0,0,
```

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,6,0,6,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,0,
6,6,0,6,6,0,12,12,6,12,12,0,0,0,0,0,0,0,6,6,0,0,0,0,0,6,6,6,12,12,0,-6,-6,0,0,
0,0,6,6,0,0,0,6,0,0,6,0,6,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,6,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,0,6,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,6,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,6,6,0,6,6,0,6,0,0,6,6,6,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,6,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,6,6,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,-6,0,6,0,6,0,6,0,0,0,0,0,0,0,12,12,6,12,12,0,6,6,0,6,6,0,
0,0,0,0,0,0,12,12,6,12,12,0,6,6,0,6,6,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,0,
0,0,0,0,0,0,6,0,0,6,6,0,6,6,0,6,0,0,6,6,6,0,0,0,-6,0,0,0,-6,0,0,-6,0,-6,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,0,6,6,0,0,0,0,0,0,6,6,0,0,0,
0,0,0,0,6,6,0,-6,0,0,-6,0,0,12,6,0,-6,-6,0,0,0,0,6,6,0,0,6,0,0,6,0,6,6,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,-6,0,0,0,0,0,0,0,0,0,0,6,6,6,6,6,0,6,12,0,6,0,0,6,0,0,0,6,0,0,0,0,0,0,
0,6,12,0,0,0,0,0,0,0,6,6,0,-6,-6,0,0,0,0,0,0,0,0,6,0,0,6,0,6,6,0,0,0,0,0,0,0,0,
6,0,0,0,6,0,0,6,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6,
0,0,0,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,6,6,0,6,6,0,6,6,6,12,12,0,0,0,0,0,0,0,6,6,0,
6,6,0,6,6,6,12,12,0,0,0,0,0,0,0,6,6,0,0,6,0,0,6,0,6,6,
```

7.14 Quillen's complex and the p-part of homology

Let G be a finite group with order divisible by prime p. Let $\mathscr{A} = \mathscr{A}_p(G)$ denote Quillen's simplicial complex arising as the order complex of the poset of non-trivial elementary abelian p-subgroups of G. The group G acts on $\mathscr{A}$. Denote the orbit of a k-simplex e^k by $[e^k]$, and the stabilizer of e^k by $Stab(e^k) \leq G$. For a finite abelian group H let H_p denote the Sylow p-subgroup or the "p-part". In Theorem 3.3 of [Web87] P.J. Webb proved the following.

THEOREM.[Web87] For any G-module M there is a (non natural) isomomorphism

$H_n(G,M)_p \oplus \bigoplus_{[e^k]:k \text{ odd}} H_n(Stab(e^k),M)_p \cong \bigoplus_{[e^k]:k \text{ even}} H_n(Stab(e^k),M)_p$

for $n \geq 1$. The isomorphism can also be expressed as

$H_n(G,M)_p \cong \bigoplus_{[e^k]:k \text{ even}} H_n(Stab(e^k),M)_p \; - \; \bigoplus_{[e^k]:k \text{ odd}} H_n(Stab(e^k),M)_p$

where terms can often be cancelled.

Thus the additive structure of the p-part of the homology of G is determined by that of the stabilizer groups. The result also holds with homology replaced by cohomology.

ILLUSTRATION 1

As an illustration of the theorem, the following commands calculate

$H_n(SL_3(\mathbb{Z}_2),\mathbb{Z}) \cong H_n(S_4,\mathbb{Z})_2 \oplus H_n(S_4,\mathbb{Z})_2 \ominus H_n(D_8,\mathbb{Z})_2 \oplus H_n(S_3,\mathbb{Z})_3 \oplus H_n(C_7 : C_3,\mathbb{Z})_7$

where $n \geq 1$, S_k denotes the symmetric group on n letters, D_8 the dihedral group of order 8 and $C_7 : C_3$ a nonabelian semi-direct product of cyclic groups. Furthermore, for $n \geq 1$

$$H_n(C_7 : C_3,\mathbb{Z})_7 = \begin{cases} \mathbb{Z}_7, & n \equiv 5 \bmod 6 \\ 0, & \text{otherwise} \end{cases}$$

and

$$H_n(S_3,\mathbb{Z})_3 = \begin{cases} \mathbb{Z}_3, & n \equiv 3 \bmod 4 \\ 0, & n \text{ otherwise.} \end{cases}$$

Formulas for $H_n(S_4,\mathbb{Z})$ and $H_n(D_8,\mathbb{Z})$ can be found in the literature. Alternatively, they can be computed using GAP for a given value of n. For $n=27$ we find

$H_{27}(S_4,\mathbb{Z})_2 \oplus H_{27}(S_4,\mathbb{Z})_2 \ominus H_{27}(D_8,\mathbb{Z})_2 \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4$

and

$H_{27}(SL_3(\mathbb{Z}_2),\mathbb{Z}) \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_3$.

Example

```
gap> G:=SL(3,2);;Factors(Order(G));
[ 2, 2, 2, 3, 7 ]
gap> D2:=HomologicalGroupDecomposition(G,2);;
gap> D3:=HomologicalGroupDecomposition(G,3);;
gap> D7:=HomologicalGroupDecomposition(G,7);;
gap> List(D2[1],StructureDescription);
[ "S4", "S4" ]
gap> List(D2[2],StructureDescription);
[ "D8" ]
gap> List(D3[1],StructureDescription);
[ "S3" ]
gap> List(D3[2],StructureDescription);
[  ]
gap> List(D7[1],StructureDescription);
[ "C7 : C3" ]
gap> List(D7[2],StructureDescription);
[  ]

gap> CohomologicalPeriod(D7[1][1]);
6
gap> List([1..6],n->GroupHomology(D7[1][1],n));
[ [ 3 ], [  ], [ 3 ], [  ], [ 3, 7 ], [  ] ]

gap> CohomologicalPeriod(D3[1][1]);
4
gap> List([1..4],n->GroupHomology(D3[1][1],n));
[ [ 2 ], [  ], [ 6 ], [  ] ]

gap> R_S4:=ResolutionFiniteGroup(Group([(1,2),(2,3),(3,4)]),28);;
gap> R_D8:=ResolutionFiniteGroup(Group([(1,2),(1,3)(2,4)]),28);;

gap> Homology(TensorWithIntegers(R_S4),27);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 12 ]
gap> Homology(TensorWithIntegers(R_D8),27);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4 ]
```

ILLUSTRATION 2

As a further illustration of the theorem, the following commands calculate

$H_n(M_{12},M)_3 \cong \bigoplus_{1\le i\le 3} H_n(Stab_i,M)_3 - \bigoplus_{4\le i\le 5} H_n(Stab_i,M)_3$

for the Mathieu simple group M_{12} of order 95040, where

$Stab_1 \cong Stab_3 = (((C_3 \times C_3) : Q_8) : C_3) : C_2$

$Stab_2 = A_4 \times S_3$

$Stab_4 = C_3 \times S_3$

$Stab_5 = ((C_3 \times C_3) : C_3) : (C_2 \times C_2)$.

Example

```
gap> G:=MathieuGroup(12);;
gap> D:=HomologicalGroupDecomposition(G,3);;
gap> List(D[1],StructureDescription);
[ "(((C3 x C3) : Q8) : C3) : C2", "A4 x S3", "(((C3 x C3) : Q8) : C3) : C2" ]
gap> List(D[2],StructureDescription);
[ "C3 x S3", "((C3 x C3) : C3) : (C2 x C2)" ]
```

ILLUSTRATION 3

As a third illustration, the following commands show that $H_n(M_{23},M)_p$ is periodic for primes $p=5,7,11,23$ of periods dividing $8,6,10,22$ respectively. They also show that $H_n(M_{23},\mathbb{Z})_{(3)}=0$ for $n=8,9,12,13$ and $H_n(M_{23},\mathbb{Z})_{(3)}=\mathbb{Z}_3$ for $n=10,11$.

Example

```
gap> G:=MathieuGroup(23);;
gap> Factors(Order(G));
[ 2, 2, 2, 2, 2, 2, 2, 3, 3, 5, 7, 11, 23 ]

gap> sd:=StructureDescription;;

gap> D:=HomologicalGroupDecomposition(G,5);;
gap> List(D[1],sd);List(D[2],sd);
[ "C15 : C4" ]
[  ]
gap> IsPeriodic(D[1][1]);
true
gap> CohomologicalPeriod(D[1][1]);
8

gap> D:=HomologicalGroupDecomposition(G,7);;
gap> List(D[1],sd);List(D[2],sd);
[ "C2 x (C7 : C3)" ]
[  ]
gap> IsPeriodic(D[1][1]);
true
gap> CohomologicalPeriod(D[1][1]);
6

gap> D:=HomologicalGroupDecomposition(G,11);;
gap> List(D[1],sd);List(D[2],sd);
[ "C11 : C5" ]
[  ]
gap> IsPeriodic(D[1][1]);
true
gap> CohomologicalPeriod(D[1][1]);
10

gap> D:=HomologicalGroupDecomposition(G,23);;
gap> List(D[1],sd);List(D[2],sd);
[ "C23 : C11" ]
[  ]
gap> IsPeriodic(D[1][1]);
```

```
true
gap> CohomologicalPeriod(D[1][1]);
22
```

Example

```
gap> G:=MathieuGroup(23);;
gap> D:=HomologicalGroupDecomposition(G,3);;
gap> List(D[1],StructureDescription);List(D[2],StructureDescription);
[ "(C3 x C3) : QD16", "A5 : S3" ]
[ "S3 x S3" ]
gap> for n in [8..13] do
> Print(List(D[1],g->GroupHomology(g,n))," , ",List(D[2],g->GroupHomology(g,n)),"\n\n");
> od;
[ [ 2, 2 ], [ 2, 2, 2 ] ] , [ [ 2, 2, 2, 2 ] ]

[ [ 2, 2, 2, 2 ], [ 2, 2, 2, 2 ] ] , [ [ 2, 2, 2, 2, 2, 2 ] ]

[ [ 2, 2, 3 ], [ 2, 2, 2, 3, 3 ] ] , [ [ 2, 2, 2, 2, 2, 3, 3 ] ]

[ [ 2, 2, 2, 8, 3 ], [ 2, 2, 2, 2, 4, 3, 3, 3, 3 ] ] ,
[ [ 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3 ] ]

[ [ 2, 2, 2 ], [ 2, 2, 2, 2 ] ] , [ [ 2, 2, 2, 2, 2, 2 ] ]

[ [ 2, 2, 2, 2, 2 ], [ 2, 2, 2, 2, 2 ] ] , [ [ 2, 2, 2, 2, 2, 2, 2, 2 ] ]
```

The order $|M_{23}| = 10200960$ is divisible by primes $p = 2,3,5,7,11,23$. For $p = 3$ the following commands establish that the Poincare series

$(x^{16} - 2x^{15} + 3x^{14} - 4x^{13} + 4x^{12} - 4x^{11} + 4x^{10} - 3x^9 + 3x^8 - 3x^7 + 4x^6 - 4x^5 + 4x^4 - 4x^3 + 3x^2 - 2x + 1)/(x^{18} - 2x^{17} + 3x^{16} - 4x^{15} + 4x^{14} - 4x^{13} + 4x^{12} - 4x^{11} + 4x^{10} - 4x^9 + 4x^8 - 4x^7 + 4x^6 - 4x^5 + 4x^4 - 4x^3 + 3x^2 - 2x + 1)$

describes the dimension of the vector space $H^n(M_{23}, \mathbb{Z}_3)$ up to at least degree $n = 40$. To prove that it describes the dimension in all degrees one would need to verify "completion criteria".

Example

```
gap> G:=MathieuGroup(23);;
gap> D:=HomologicalGroupDecomposition(G,3);;
gap> List(D[1],StructureDescription);
[ "(C3 x C3) : QD16", "A5 : S3" ]
gap> List(D[2],StructureDescription);
[ "S3 x S3" ]

gap> P1:=PoincareSeriesPrimePart(D[1][1],3,40);
(x_1^16-2*x_1^15+3*x_1^14-4*x_1^13+4*x_1^12-4*x_1^11+4*x_1^10-3*x_1^9+3*x_1^8-3*x_1^7+4*x_1^6-4*x
4*x_1^4-4*x_1^3+3*x_1^2-2*x_1+1)/(x_1^18-2*x_1^17+3*x_1^16-4*x_1^15+4*x_1^14-4*x_1^13+4*x_1^12-4*
1+4*x_1^10-4*x_1^9+4*x_1^8-4*x_1^7+4*x_1^6-4*x_1^5+4*x_1^4-4*x_1^3+3*x_1^2-2*x_1+1)

gap> P2:=PoincareSeriesPrimePart(D[1][2],3,40);
(x_1^4-2*x_1^3+3*x_1^2-2*x_1+1)/(x_1^6-2*x_1^5+3*x_1^4-4*x_1^3+3*x_1^2-2*x_1+1)

gap> P3:=PoincareSeriesPrimePart(D[2][1],3,40);
```

```
(x_1^4-2*x_1^3+3*x_1^2-2*x_1+1)/(x_1^6-2*x_1^5+3*x_1^4-4*x_1^3+3*x_1^2-2*x_1+1)
```

7.15 Homology of a Lie algebra

Let A be the Lie algebra constructed from the associative algebra $M^{4\times 4}(\mathbb{Q})$ of all 4×4 rational matrices. Let V be its adjoint module (with underlying vector space of dimension 16 and equal to that of A). The following commands compute $H_4(A,V) = \mathbb{Q}$.

Example

```
gap> M:=FullMatrixAlgebra(Rationals,4);;
gap> A:=LieAlgebra(M);;
gap> V:=AdjointModule(A);;
gap> C:=ChevalleyEilenbergComplex(V,17);;
gap> List([0..17],C!.dimension);
[ 16, 256, 1920, 8960, 29120, 69888, 128128, 183040, 205920, 183040, 128128,
  69888, 29120, 8960, 1920, 256, 16, 0 ]
gap> Homology(C,4);
1
```

Note that the eighth term $C_8(V)$ in the Chevalley-Eilenberg complex $C_*(V)$ is a vector space of dimension 205920 and so it will take longer to compute the homology in degree 8.

As a second example, let B be the classical Lie ring of type B_3 over the ring of integers. The following commands compute $H_3(B,\mathbb{Z}) = \mathbb{Z}\oplus\mathbb{Z}_2^{105}$.

Example

```
gap> A:=SimpleLieAlgebra("B",7,Integers);
<Lie algebra of dimension 105 over Integers>
gap> C:=ChevalleyEilenbergComplex(A,4,"sparse");
Sparse chain complex of length 4 in characteristic 0 .

gap> D:=ContractedComplex(C);
Sparse chain complex of length 4 in characteristic 0 .

gap> Collected(Homology(D,3));
[ [ 0, 1 ], [ 2, 105 ] ]
```

7.16 Covers of Lie algebras

A short exact sequence of Lie algebras

$M \rightarrowtail C \twoheadrightarrow L$

(over a field k) is said to be a *stem extension* of L if M lies both in the centre $Z(C)$ and in the derived subalgeba C^2. If, in addition, the rank of the vector space M is equal to the rank of the second Chevalley-Eilenberg homology $H_2(L,k)$ then the Lie algebra C is said to be a *cover* of L.

Each finite dimensional Lie algebra L admits a cover C, and this cover can be shown to be unique up to Lie isomorphism.

The cover can be used to determine whether there exists a Lie algebra E whose central quotient $E/Z(E)$ is isomorphic to L. The image in L of the centre of C is called the *Lie Epicentre* of L, and this image is trivial if and only if such an E exists.

The cover can also be used to determine the stem extensions of L. It can be shown that each stem extension is a quotient of the cover by an ideal in the Lie multiplier $H_2(L,k)$.

7.16.1 Computing a cover

The following commands compute the cover C of the solvable but non-nilpotent 13-dimensional Lie algebra L (over $k=\mathbb{Q}$) that was introduced by M. Wuestner [Wue92]. They also show that: the second homology of C is trivial and compute the ranks of the homology groups in other dimensions; the Lie algebra L is not isomorphic to any central quotient $E/Z(E)$.

Example

```
gap> SCTL:=EmptySCTable(13,0,"antisymmetric");;
gap> SetEntrySCTable( SCTL, 1, 6, [ 1, 7 ] );;
gap> SetEntrySCTable( SCTL, 1, 8, [ 1, 9 ] );;
gap> SetEntrySCTable( SCTL, 1, 10, [ 1, 11 ] );;
gap> SetEntrySCTable( SCTL, 1, 12, [ 1, 13 ] );;
gap> SetEntrySCTable( SCTL, 1, 7, [ -1, 6 ] );;
gap> SetEntrySCTable( SCTL, 1, 9, [ -1, 8 ] );;
gap> SetEntrySCTable( SCTL, 1, 11, [ -1, 10 ] );;
gap> SetEntrySCTable( SCTL, 1, 13, [ -1, 12 ] );;
gap> SetEntrySCTable( SCTL, 6, 7, [ 1, 2 ] );;
gap> SetEntrySCTable( SCTL, 8, 9, [ 1, 3 ] );;
gap> SetEntrySCTable( SCTL, 6, 9, [ -1, 5 ] );;
gap> SetEntrySCTable( SCTL, 7, 8, [ 1, 5 ] );;
gap> SetEntrySCTable( SCTL, 2, 8, [ 1, 12 ] );;
gap> SetEntrySCTable( SCTL, 2, 9, [ 1, 13 ] );;
gap> SetEntrySCTable( SCTL, 3, 6, [ 1, 10 ] );;
gap> SetEntrySCTable( SCTL, 3, 7, [ 1, 11 ] );;
gap> SetEntrySCTable( SCTL, 2, 3, [ 1, 4 ] );;
gap> SetEntrySCTable( SCTL, 5, 6, [ -1, 12 ] );;
gap> SetEntrySCTable( SCTL, 5, 7, [ -1, 13 ] );;
gap> SetEntrySCTable( SCTL, 5, 8, [ -1, 10 ] );;
gap> SetEntrySCTable( SCTL, 5, 9, [ -1, 11 ] );;
gap> SetEntrySCTable( SCTL, 6, 11, [ -1/2, 4 ] );;
gap> SetEntrySCTable( SCTL, 7, 10, [ 1/2, 4 ] );;
gap> SetEntrySCTable( SCTL, 8, 13, [ 1/2, 4 ] );;
gap> SetEntrySCTable( SCTL, 9, 12, [ -1/2, 4 ] );;
gap> L:=LieAlgebraByStructureConstants(Rationals,SCTL);;

gap> C:=Source(LieCoveringHomomorphism(L));
<Lie algebra of dimension 15 over Rationals>

gap> Dimension(LieEpiCentre(L));
1

gap> ch:=ChevalleyEilenbergComplex(C,17);;
gap> List([0..16],n->Homology(ch,n));
[ 1, 1, 0, 9, 23, 27, 47, 88, 88, 47, 27, 23, 9, 0, 1, 1, 0 ]
```

Chapter 8

Cohomology rings and Steenrod operations for groups

8.1 Mod-p cohomology rings of finite groups

For a finite group G, prime p and positive integer deg the function `ModPCohomologyRing(G,p,deg)` computes a finite dimensional graded ring equal to the cohomology ring $H^{\leq deg}(G,\mathbb{Z}_p) := H^*(G,\mathbb{Z}_p)/\{x = 0 \ : \ \text{degree}(x) > deg\}$.

The following example computes the first 14 degrees of the cohomology ring $H^*(M_{11},\mathbb{Z}_2)$ where M_{11} is the Mathieu group of order 7920. The ring is seen to be generated by three elements a_3, a_4, a_6 in degrees $3,4,5$.

Example

```
gap> G:=MathieuGroup(11);;
gap> p:=2;;deg:=14;;
gap> A:=ModPCohomologyRing(G,p,deg);
<algebra over GF(2), with 20 generators>

gap> gns:=ModPRingGenerators(A);
[ v.1, v.6, v.8+v.10, v.13 ]
gap> List(gns,A!.degree);
[ 0, 3, 4, 5 ]
```

The following additional command produces a rational function $f(x)$ whose series expansion $f(x) = \sum_{i=0}^{\infty} f_i x^i$ has coefficients f_i which are guaranteed to satisfy $f_i = \dim H^i(G,\mathbb{Z}_p)$ in the range $0 \leq i \leq deg$. We refer to $f(x)$ as the *Poincare series* for the group at the prime $p = 2$.

Example

```
gap> f:=PoincareSeries(A);
(x_1^4-x_1^3+x_1^2-x_1+1)/(x_1^6-x_1^5+x_1^4-2*x_1^3+x_1^2-x_1+1)


gap> Let's use f to list the first few cohomology dimensions
gap> ExpansionOfRationalFunction(f,deg);
[ 1, 0, 0, 1, 1, 1, 1, 1, 2, 2, 1, 2, 3, 2, 2 ]
```

An alternative command for computing the Poincare series is the following. In this alternative we choose to ensure correctness in degrees ≤ 100.

Example

```
gap> PoincareSeriesPrimePart(MathieuGroup(11),2,100);
The series is guaranteed correct for group cohomology in degrees < 101
(x_1^4-x_1^3+x_1^2-x_1+1)/(x_1^6-x_1^5+x_1^4-2*x_1^3+x_1^2-x_1+1)
```

If one needs to verify that the Poincare series is valid in all degrees then more work is required. One readily implemented (but computationally non-optimal) approach is to use Peter Symmonds result [Sym10] that: if a non-cyclic finite group G has a faithful complex representation equal to a sum of irreducibles of dimensions n_i then the cohomology ring $H^*(G,\mathbb{Z}_p)$ is generated by elements of degree at most $\sum n_i^2$; a degree bound for the relations is $2\sum n_i^2$. The following commands use this bound, in conjunction with Webb's result 7.14 on the Quillen complex, to obtained a Poincare series that is guaranteed correct in all degree.

Example

```
gap> G:=MathieuGroup(11);;
gap> h:=HomologicalGroupDecomposition(G,2);;

gap> ModPCohomologyPresentationBounds(h[1][1]);
rec( generators_degree_bound := 4, relators_degree_bound := 8 )
gap> A:=ModPCohomologyRing(h[1][1],2,9);;F:=PresentationOfGradedStructureConstantAlgebra(A);;f11:
(x_1^2-x_1+1)/(x_1^4-2*x_1^3+2*x_1^2-2*x_1+1)

gap> ModPCohomologyPresentationBounds(h[1][2]);
rec( generators_degree_bound := 9, relators_degree_bound := 18 )
gap> A:=ModPCohomologyRing(h[1][2],2,19);;F:=PresentationOfGradedStructureConstantAlgebra(A);;f12
(x_1^2+1)/(x_1^4-x_1^3-x_1+1)

gap> ModPCohomologyPresentationBounds(h[2][1]);
rec( generators_degree_bound := 4, relators_degree_bound := 8 )
gap> A:=ModPCohomologyRing(h[2][1],2,9);;F:=PresentationOfGradedStructureConstantAlgebra(A);;f21:
(1)/(x_1^2-2*x_1+1)

gap> f11+f12-f21;
(x_1^4-x_1^3+x_1^2-x_1+1)/(x_1^6-x_1^5+x_1^4-2*x_1^3+x_1^2-x_1+1)
```

8.1.1 Ring presentations (for the commutative $p = 2$ case)

The cohomology ring $H^*(G,\mathbb{Z}_p)$ is graded commutative which, in the case $p = 2$, implies strictly commutative. The following additional commands can be applied in the $p = 2$ setting to determine a presentation for a graded commutative ring F that is guaranteed to be isomorphic to the cohomology ring $H^*(G,\mathbb{Z}_p)$ in degrees $i \leq deg$. If deg is chosen "sufficiently large" then F will be isomorphic to the cohomology ring.

Example

```
gap> F:=PresentationOfGradedStructureConstantAlgebra(A);
Graded algebra GF(2)[ x_1, x_2, x_3 ] / [ x_1^2*x_2+x_3^2
 ] with indeterminate degrees [ 3, 4, 5 ]
```

The additional command

Example

```
gap> p:=HilbertPoincareSeries(F);
(x_1^4-x_1^3+x_1^2-x_1+1)/(x_1^6-x_1^5+x_1^4-2*x_1^3+x_1^2-x_1+1)
```

invokes a call to SINGULAR in order to calculate the Poincare series of the graded algebra F.

8.2 Poincare Series for Mod-p cohomology

For a finite p-group G the command `PoincarePolynomial(G)` returns a rational function $f(x) = p(x)/q(x)$ whose series expansion $f(x) = \sum_{i=0}^{\infty} f_i x^i$ has coefficients f_i that are guaranteed to satisfy $f_i = \dim H^i(G, \mathbb{Z}_p)$ in the range $0 \leq i < 1 + deg$ for some displayed value of deg. Furthermore, the coefficients f_i are guaranteed to be integers for all $0 \leq i \leq 1000$ and the order of the pole of $f(x)$ at $x = 1$ is guaranteed to equal the p-rank of G.

Example

```
gap> G:=SmallGroup(3^4,10);;
gap> StructureDescription(G);
"C3 . ((C3 x C3) : C3) = (C3 x C3) . (C3 x C3)"
gap> f:=PoincareSeries(G);
The series is guaranteed correct for group cohomology in degrees < 14
(-x_1^3+x_1^2+1)/(x_1^6-2*x_1^5+2*x_1^4-2*x_1^3+2*x_1^2-2*x_1+1)
```

If a higher value of deg is required then this can be entered as an optional second argument. For instance, the following increases the value to $deg = 100$.

Example

```
gap> f:=PoincareSeries(G,100);
The series is guaranteed correct for group cohomology in degrees < 101
(-x_1^3+x_1^2+1)/(x_1^6-2*x_1^5+2*x_1^4-2*x_1^3+2*x_1^2-2*x_1+1)
```

As mentioned above, one approach to verifying that the Poincare series is valid in all degrees is to use Peter Symmonds result [Sym10] that: if a non-cyclic finite group G has a faithful complex representation equal to a sum of irreducibles of dimensions n_i then the cohomology ring $H^*(G, \mathbb{Z}_p)$ is generated by elements of degree at most $\sum n_i^2$; a degree bound for the relations is $2\sum n_i^2$. Thus, if we use at least $\sum n_i^2$ degrees of a resolution to construct a presentation for the cohomology ring then the presented ring maps surjectively onto the actual cohomology ring. Furthermore, if this surjection is a bijection in the first $2\sum n_i^2$ degrees then it is necessarily an isomorphism in all degrees.

The following commands use this approach to obtain a guaranteed presentation and Poincare series for the Sylow 2-subgroup of the Mathieu group M_{12}.

Example

```
gap> G:=SylowSubgroup(MathieuGroup(12),2);;
gap> ModPCohomologyPresentationBounds(G);
rec( generators_degree_bound := 16, relators_degree_bound := 32 )
gap> A:=ModPCohomologyRing(G,16);;
gap> F:=PresentationOfGradedStructureConstantAlgebra(A);
Graded algebra GF(2)[ x_1, x_2, x_3, x_4, x_5, x_6, x_7 ] /
[ x_1*x_3, x_1*x_2, x_1*x_4, x_2*x_3^2+x_3^3+x_3*x_4+x_3*x_5,
```

```
  x_2*x_6+x_3*x_6+x_4*x_5, x_2*x_3*x_4+x_3^2*x_4+x_3*x_6,
  x_2^2*x_4+x_3^2*x_4+x_3*x_6+x_4^2, x_3^2*x_6+x_3*x_4^2+x_3*x_4*x_5,
  x_2*x_4*x_5+x_4*x_6, x_3^2*x_4*x_5+x_3*x_4*x_6+x_3*x_5*x_6,
  x_1^3*x_6+x_1^2*x_7+x_1*x_5*x_6+x_3*x_5*x_6+x_4*x_5^2+x_6^2,
  x_3*x_4^2*x_5+x_3*x_6^2 ] with indeterminate degrees [ 1, 1, 1, 2, 2, 3, 4 ]
gap> f:=HilbertPoincareSeries(F);
(1)/(-x_1^3+3*x_1^2-3*x_1+1)

gap> ff:=PoincareSeries(G,32);
The series is guaranteed correct for group cohomology in degrees < 33
(1)/(-x_1^3+3*x_1^2-3*x_1+1)
```

An alternative approach to obtaining a guaranteed presentation is to implement Len even's spectral sequence proof of the finite generation of cohomology rings of finite groups. The following example determines a guaranteed presentation in this way for the cohomology ring $H^*(Syl_2(M_{12}),\mathbb{Z}_2)$. The Lyndon-Hochschild-Serre spectral sequence, and Groebner basis routines from SINGULAR (for commutative rings), are used to determine how much of a resolution is needed to compute the guaranteed correct presentation.

Example

```
gap> G:=SylowSubgroup(MathieuGroup(12),2);;
gap> F:=Mod2CohomologyRingPresentation(G);
Alpha version of completion test code will be used. This needs further work.
Graded algebra GF(2)[ x_1, x_2, x_3, x_4, x_5, x_6, x_7 ] /
[ x_2*x_3, x_1*x_2, x_2*x_4, x_1^3+x_1^2*x_3+x_1*x_5,
  x_1*x_3*x_4+x_1*x_3*x_5+x_3^2*x_4+x_1*x_6+x_3*x_6+x_4*x_5,
  x_1^2*x_4+x_1^2*x_5+x_1*x_3*x_5+x_3^2*x_4+x_1*x_6+x_4^2,
  x_1^2*x_3^2+x_1^2*x_5+x_1*x_3*x_5+x_1*x_6+x_3*x_6+x_4^2+x_4*x_5,
  x_1^2*x_6+x_1*x_3*x_6+x_1*x_4*x_5+x_3^2*x_6+x_3*x_4^2+x_3*x_4*x_5,
  x_1*x_3^2*x_5+x_3^3*x_4+x_1*x_3*x_6+x_1*x_4^2+x_3^2*x_6+x_3*x_4^2+x_4*x_6,
  x_1^2*x_3*x_5+x_1*x_3*x_6+x_1*x_4^2+x_1*x_5^2,
  x_3^3*x_6+x_3^2*x_4^2+x_3^2*x_4*x_5+x_3*x_4*x_6+x_3*x_5*x_6+x_4^3+x_4*x_5^2,
  x_1*x_3^2*x_6+x_1*x_4*x_6+x_2^2*x_7+x_2*x_5*x_6+x_3*x_4*x_6+x_3*x_5*x_6+x_6^2,
  x_1^2*x_5^2+x_1*x_3*x_5^2+x_3^2*x_4^2+x_3^2*x_4*x_5+x_2^2*x_7+x_2*x_5*x_6+x_3*x_5*x_6+x_6^2 ]
with indeterminate degrees [ 1, 1, 1, 2, 2, 3, 4 ]

gap> f:=HilbertPoincareSeries(F);
(1)/(-x_1^3+3*x_1^2-3*x_1+1)
```

8.3 Functorial ring homomorphisms in Mod-p cohomology

The following example constructs the ring homomorphism

$F: H^{\leq deg}(G,\mathbb{Z}_p) \to H^{\leq deg}(H,\mathbb{Z}_p)$

induced by the group homomorphism $f: H \to G$ with $H = A_5$, $G = S_5$, f the canonical inclusion of the alternating group into the symmetric group, $p = 2$ and $deg = 7$.

Example

```
gap> G:=SymmetricGroup(5);;H:=AlternatingGroup(5);;
gap> f:=GroupHomomorphismByFunction(H,G,x->x);;
gap> p:=2;; deg:=7;;
```

```
gap> F:=ModPCohomologyRing(f,p,deg);
[ v.1, v.2, v.4+v.6, v.5, v.7, v.8, v.9, v.12+v.15, v.13, v.14, v.16+v.17,
  v.18, v.19, v.20, v.22+v.24+v.28, v.23, v.25, v.26, v.27 ] ->
[ v.1, 0*v.1, v.4+v.5+v.6, 0*v.1, v.7+v.8, 0*v.1, 0*v.1, v.14+v.15, 0*v.1,
  0*v.1, v.16+v.17+v.19, 0*v.1, 0*v.1, 0*v.1, v.22+v.23+v.26+v.27+v.28,
  v.25, 0*v.1, 0*v.1, 0*v.1 ]
```

8.3.1 Testing homomorphism properties

The following commands are consistent with F being a ring homomorphism.

Example

```
gap> x:=Random(Source(F));
v.4+v.6+v.8+v.9+v.12+v.13+v.14+v.15+v.18+v.20+v.22+v.24+v.25+v.28+v.32+v.35
gap> y:=Random(Source(F));
v.1+v.2+v.7+v.9+v.13+v.23+v.26+v.27+v.32+v.33+v.34+v.35
gap> Image(F,x)+Image(F,y)=Image(F,x+y);
true
gap> Image(F,x)*Image(F,y)=Image(F,x*y);
true
```

8.3.2 Testing functorial properties

The following example takes two "random" automorphisms $f,g:K \to K$ of the group K of order 24 arising as the direct product $K = C_3 \times Q_8$ and constructs the three ring isomorphisms $F,G,FG:H^{\leq 5}(K,\mathbb{Z}_2) \to H^{\leq 5}(K,\mathbb{Z}_2)$ induced by f,g and the composite $f \circ g$. It tests that FG is indeed the composite $G \circ F$. Note that when we create the ring $H^{\leq 5}(K,\mathbb{Z}_2)$ twice in GAP we obtain two canonically isomorphic but distinct implimentations of the ring. Thus the canocial isomorphism between these distinct implementations needs to be incorporated into the test. Note also that GAP defines $g * f = f \circ g$.

Example

```
gap> K:=SmallGroup(24,11);;
gap> aut:=AutomorphismGroup(K);;
gap> f:=Elements(aut)[5];;
gap> g:=Elements(aut)[8];;
gap> fg:=g*f;;
gap> F:=ModPCohomologyRing(f,2,5);
[ v.1, v.2, v.3, v.4, v.5, v.6, v.7 ] -> [ v.1, v.2+v.3, v.3, v.4+v.5, v.5,
  v.6, v.7 ]
gap> G:=ModPCohomologyRing(g,2,5);
[ v.1, v.2, v.3, v.4, v.5, v.6, v.7 ] -> [ v.1, v.2+v.3, v.2, v.5, v.4+v.5,
  v.6, v.7 ]
gap> FG:=ModPCohomologyRing(fg,2,5);
[ v.1, v.2, v.3, v.4, v.5, v.6, v.7 ] -> [ v.1, v.3, v.2, v.4, v.4+v.5, v.6,
  v.7 ]

gap> sF:=Source(F);;tF:=Target(F);;
gap> sG:=Source(G);;
gap> tGsF:=AlgebraHomomorphismByImages(tF,sG,Basis(tF),Basis(sG));;
```

```
gap> List(GeneratorsOfAlgebra(sF),x->Image(G,Image(tGsF,Image(F,x))));
[ v.1, v.3, v.2, v.4, v.4+v.5, v.6, v.7 ]
```

8.3.3 Computing with larger groups

Mod-p cohomology rings of finite groups are constructed as the rings of stable elements in the cohomology of a (non-functorially) chosen Sylow p-subgroup and thus require the construction of a free resolution only for the Sylow subgroup. However, to ensure the functoriality of induced cohomology homomorphisms the above computations construct free resolutions for the entire groups G, H. This is a more expensive computation than finding resolutions just for Sylow subgroups.

The default algorithm used by the function `ModPCohomologyRing()` for constructing resolutions of a finite group G is `ResolutionFiniteGroup()` or `ResolutionPrimePowerGroup()` in the case when G happens to be a group of prime-power order. If the user is able to construct the first *deg* terms of free resolutions RG, RH for the groups G, H then the pair `[RG,RH]` can be entered as the third input variable of `ModPCohomologyRing()`.

For instance, the following example constructs the ring homomorphism
$F: H^{\leq 7}(A_6, \mathbb{Z}_2) \to H^{\leq 7}(S_6, \mathbb{Z}_2)$
induced by the the canonical inclusion of the alternating group A_6 into the symmetric group S_6.

Example

```
gap> G:=SymmetricGroup(6);;
gap> H:=AlternatingGroup(6);;
gap> f:=GroupHomomorphismByFunction(H,G,x->x);;
gap> RG:=ResolutionFiniteGroup(G,7);;
gap> RH:=ResolutionFiniteSubgroup(RG,H);;
gap> F:=ModPCohomologyRing(f,2,[RG,RH]);
[ v.1, v.2+v.3, v.6+v.8+v.10, v.7+v.9, v.11+v.12, v.13+v.15+v.16+v.18+v.19,
  v.14+v.16+v.19, v.17, v.22, v.23+v.28+v.32+v.35,
  v.24+v.26+v.27+v.29+v.32+v.33+v.35, v.25+v.26+v.27+v.29+v.32+v.33+v.35,
  v.30+v.32+v.33+v.34+v.35, v.36+v.39+v.43+v.45+v.47+v.49+v.50+v.55,
  v.38+v.45+v.47+v.49+v.50+v.55, v.40,
  v.41+v.43+v.45+v.47+v.48+v.49+v.50+v.53+v.55,
  v.42+v.43+v.45+v.46+v.47+v.49+v.53+v.54, v.44+v.45+v.46+v.47+v.49+v.53+v.54,
  v.51+v.52, v.58+v.60, v.59+v.68+v.73+v.77+v.81+v.83,
  v.62+v.68+v.74+v.77+v.78+v.80+v.81+v.83+v.84,
  v.63+v.69+v.73+v.74+v.78+v.80+v.84, v.64+v.68+v.73+v.77+v.81+v.83, v.65,
  v.66+v.75+v.81, v.67+v.68+v.69+v.70+v.73+v.74+v.78+v.80+v.84,
  v.71+v.72+v.73+v.76+v.77+v.78+v.80+v.82+v.83+v.84, v.79 ] ->
[ v.1, 0*v.1, v.4+v.5+v.6, 0*v.1, v.8, v.8, 0*v.1, v.7, 0*v.1,
  v.12+v.13+v.14+v.15, v.12+v.13+v.14+v.15, v.12+v.13+v.14+v.15,
  v.12+v.13+v.14+v.15, v.18+v.19, 0*v.1, 0*v.1, v.18+v.19, v.18+v.19,
  v.18+v.19, v.16+v.17, 0*v.1, v.25, v.22+v.24+v.25+v.26+v.27+v.28,
  v.22+v.24+v.25+v.26+v.27+v.28, 0*v.1, 0*v.1, v.25, v.22+v.24+v.26+v.27+v.28,
  v.22+v.24+v.26+v.27+v.28, v.23 ]
```

8.4 Steenrod operations for finite 2-groups

The command `CohomologicalData(G,n)` prints complete information for the cohomology ring $H^*(G,Z_2)$ and steenrod operations for a 2-group G provided that the integer n is at least the maximal degree of a generator or relator in a minimal set of generatoirs and relators for the ring.

The following example produces complete information on the Steenrod algebra of group number 8 in GAP's library of groups of order 32. Groebner basis routines (for commutative rings) from SINGULAR are called in the example. (This example take over 2 hours to run. Most other groups of order 32 run significantly quicker.)

Example

```
gap> CohomologicalData(SmallGroup(32,8),12);

Integer argument is large enough to ensure completeness of cohomology ring presentation.

Group number: 8
Group description: C2 . ((C4 x C2) : C2) = (C2 x C2) . (C4 x C2)

Cohomology generators
Degree 1: a, b
Degree 2: c, d
Degree 3: e
Degree 5: f, g
Degree 6: h
Degree 8: p

Cohomology relations
1: f^2
2: c*h+e*f
3: c*f
4: b*h+c*g
5: b*e+c*d
6: a*h
7: a*g
8: a*f+b*f
9: a*e+c^2
10: a*c
11: a*b
12: a^2
13: d*e*h+e^2*g+f*h
14: d^2*h+d*e*f+d*e*g+f*g
15: c^2*d+b*f
16: b*c*g+e*f
17: b*c*d+c*e
18: b^2*g+d*f
19: b^2*c+c^2
20: b^3+a*d
21: c*d^2*e+c*d*g+d^2*f+e*h
22: c*d^3+d*e^2+d*h+e*f+e*g
23: b^2*d^2+c*d^2+b*f+e^2
24: b^3*d
25: d^3*e^2+d^2*e*f+c^2*p+h^2
26: d^4*e+b*c*p+e^2*g+g*h
```

```
27: d^5+b*d^2*g+b^2*p+f*g+g^2

Poincare series
(x^5+x^2+1)/(x^8-2*x^7+2*x^6-2*x^5+2*x^4-2*x^3+2*x^2-2*x+1)

Steenrod squares
Sq^1(c)=0
Sq^1(d)=b*b*b+d*b
Sq^1(e)=c*b*b
Sq^2(e)=e*d+f
Sq^1(f)=c*d*b*b+d*d*b*b
Sq^2(f)=g*b*b
Sq^4(f)=p*a
Sq^1(g)=d*d*d+g*b
Sq^2(g)=0
Sq^4(g)=c*d*d*d*b+g*d*b*b+g*d*d+p*a+p*b
Sq^1(h)=c*d*d*b+e*d*d
Sq^2(h)=d*d*d*b*b+c*d*d*d+g*c*b
Sq^4(h)=d*d*d*d*b*b+g*e*d+p*c
Sq^1(p)=c*d*d*d*b
Sq^2(p)=d*d*d*d*b*b+c*d*d*d*d
Sq^4(p)=d*d*d*d*d*b*b+d*d*d*d*d*d*d+g*d*d*d*b+g*g*d+p*d*d
```

8.5 Steenrod operations on the classifying space of a finite p-group

The following example constructs the first eight degrees of the mod-3 cohomology ring $H^*(G,\mathbb{Z}_3)$ for the group G number 4 in GAP's library of groups of order 81. It determines a minimal set of ring generators lying in degree ≤ 8 and it evaluates the Bockstein operator on these generators. Steenrod powers for $p \geq 3$ are not implemented as no efficient method of implementation is known.

Example

```
gap> G:=SmallGroup(81,4);;
gap> A:=ModPSteenrodAlgebra(G,8);;
gap> List(ModPRingGenerators(A),x->Bockstein(A,x));
[ 0*v.1, 0*v.1, v.5, 0*v.1, (Z(3))*v.7+v.8+(Z(3))*v.9 ]
```

8.6 Mod-p cohomology rings of crystallographic groups

Mod p cohomology ring computations can be attempted for any group G for which we can compute sufficiently many terms of a free ZG-resolution with contracting homotopy. The contracting homotopy is not needed if only the dimensions of the cohomology in each degree are sought. Crystallographic groups are one class of infinite groups where such computations can be attempted.

8.6.1 Poincare series for crystallographic groups

Consider the space group $G = SpaceGroupOnRightIT(3,226,'1')$. The following computation computes the infinite series

$(-2x^4+2x^2+1)/(-x^5+2x^4-x^3+x^2-2x+1)$

in which the coefficient of the monomial x^n is guaranteed to equal the dimension of the vector space $H^n(G,\mathbb{Z}_2)$ in degrees $n \leq 14$. One would need to involve a theoretical argument to establish that this equality in fact holds in every degree $n \geq 0$.

Example

```
gap> G:=SpaceGroupIT(3,226);
SpaceGroupOnRightIT(3,226,'1')
gap> R:=ResolutionSpaceGroup(G,15);
Resolution of length 15 in characteristic 0 for <matrix group with
8 generators> .
No contracting homotopy available.

gap> D:=List([0..14],n->Cohomology(HomToIntegersModP(R,2),n));
[ 1, 2, 5, 9, 11, 15, 20, 23, 28, 34, 38, 44, 51, 56, 63 ]

gap> PoincareSeries(D,14);
(-2*x_1^4+2*x_1^2+1)/(-x_1^5+2*x_1^4-x_1^3+x_1^2-2*x_1+1)
```

Consider the space group $SpaceGroupOnRightIT(3,103,'1')$. The following computation uses a different construction of a free resolution to compute the infinite series

$(x^3+2x^2+2x+1)/(-x+1)$

in which the coefficient of the monomial x^n is guaranteed to equal the dimension of the vector space $H^n(G,\mathbb{Z}_2)$ in degrees $n \leq 99$. The final commands show that G acts on a (cubical) cellular decomposition of $\mathbb{R}^3$ with cell ctabilizers being either trivial or cyclic of order 2 or 4. From this extra calculation it follows that the cohomology is periodic in degrees greater than 3 and that the Poincare series is correct in every degree $n \geq 0$.

Example

```
gap> G:=SpaceGroupIT(3,103);
SpaceGroupOnRightIT(3,103,'1')
gap> R:=ResolutionCubicalCrystGroup(G,100);
Resolution of length 100 in characteristic 0 for <matrix group with 6 generators> .

gap> D:=List([0..99],n->Cohomology(HomToIntegersModP(R,2),n));;
gap> PoincareSeries(D,99);
(x_1^3+2*x_1^2+2*x_1+1)/(-x_1+1)


#Torsion subgroups are cyclic
gap> B:=CrystGFullBasis(G);;
gap> C:=CrystGcomplex(GeneratorsOfGroup(G),B,1);;
gap> for n in [0..3] do
> for k in [1..C!.dimension(n)] do
> Print(StructureDescription(C!.stabilizer(n,k)),"  ");
> od;od;
C4  C2  C4  1  1  C4  C2  C4  1  1  1  1
```

8.6.2 Mod 2 cohomology rings of 3-dimensional crystallographic groups

Computations in the *integral* cohomology of a crystallographic group are illustrated in Section 1.19. The commands underlying that illustration could be further developed and adapted to mod p cohomology. Indeed, the authors of the paper [LY24a] have developed commands for accessing the mod 2 cohomology of 3-dimensional crystallographic groups with the aim of establishing a connection between these rings and the lattice structure of crystals with space group symmetry. Their code is available at the github repository [LY24b]. In particular, their code contains the command

- `SpaceGroupCohomologyRingGapInterface(ITC)`

that inputs an integer in the range $1 \leq ITC \leq 230$ corresponding to the numbering of a 3-dimensional space group G in the International Table for Crystallography. This command returns

- a presentation for the mod 2 cohomology ring $H^*(G,\mathbb{Z}_2)$. The presentation is guaranteed to be correct for low degree cohomology. In cases where the cohomology is periodic in degrees ≥ 5 (which can be tested using `IsPeriodicSpaceGroup(G)`) the presentation is guaranteed correct in all degrees. In non-periodic cases some additional mathematical argument needs to be provided to be mathematically sure that the presentation is correct in all degrees.

- the Lieb-Schultz-Mattis anomaly (degree-3 cocycles) associated with the Irreducible Wyckoff Position (see the paper [LY24a] for a definition).

The command `SpaceGroupCohomologyRingGapInterface(ITC)` is fast for most groups (a few seconds to a few minutes) but can be very slow for certain space groups (e.g. ITC $= 228$ and ITC $= 142$). The following illustration assumes that two relevant files have been downloaded from [LY24b] and illustrates the command for ITC $= 30$ and ITC $= 216$.

Example

```
gap> Read("SpaceGroupCohomologyData.gi");         #These two files must be
gap> Read("SpaceGroupCohomologyFunctions.gi");    #downloaded from
gap>       #https://github.com/liuchx1993/Space-Group-Cohomology-and-LSM/

gap> IsPeriodicSpaceGroup(SpaceGroupIT(3,30));
true

gap> SpaceGroupCohomologyRingGapInterface(30);
==========================================
Mod-2 Cohomology Ring of Group No. 30:
Z2[Ac,Am,Ax,Bb]/<R2,R3,R4>
R2:  Ac.Am  Am^2  Ax^2+Ac.Ax
R3:  Am.Bb
R4:  Bb^2
==========================================
LSM:
2a Ac.Bb+Ax.Bb
2b Ax.Bb
true


gap> IsPeriodicSpaceGroup(SpaceGroupIT(3,216));
false
```

```
gap> SpaceGroupCohomologyRingGapInterface(216);
==========================================
Mod-2 Cohomology Ring of Group No. 216:
Z2[Am,Ba,Bb,Bxyxzyz,Ca,Cb,Cc,Cxyz]/<R4,R5,R6>
R4:  Am.Ca  Am.Cb  Ba.Bxyxzyz+Am.Cc  Bb^2+Am.Cc+Ba.Bb  Bb.Bxyxzyz+Am^2.Bb+Am.Cxyz  Bxyxzyz^2
R5:  Bxyxzyz.Ca  Ba.Cb+Bb.Ca  Bb.Cb+Bb.Ca  Bxyxzyz.Cb  Bxyxzyz.Cc  Ba.Cxyz+Am.Ba.Bb+Bb.Cc  Bb.Cxy
==========================================
LSM:
4a Ca+Cc+Cxyz
4b Cb+Cc+Cxyz
4c Cb+Cxyz
4d Cxyz
true
```

In the example the naming convention for ring generators follows the paper [LY24a].

Chapter 9

Bredon homology

9.1 Davis complex

The following example computes the Bredon homology

$\underline{H}_0(W,\mathscr{R}) = \mathbb{Z}^{21}$

for the infinite Coxeter group W associated to the Dynkin diagram shown in the computation, with coefficients in the complex representation ring.

Example

```
gap> D:=[[1,[2,3]],[2,[3,3]],[3,[4,3]],[4,[5,6]]];;
gap> CoxeterDiagramDisplay(D);
```

Example

```
gap> C:=DavisComplex(D);;
gap> D:=TensorWithComplexRepresentationRing(C);;
gap> Homology(D,0);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

9.2 Arithmetic groups

The following example computes the Bredon homology

$\underline{H}_0(SL_2(\mathscr{O}_{-3}),\mathscr{R}) = \mathbb{Z}_2 \oplus \mathbb{Z}^9$

$\underline{H}_1(SL_2(\mathscr{O}_{-3}),\mathscr{R}) = \mathbb{Z}$

for $\mathscr{O}_{-3}$ the ring of integers of the number field $\mathbb{Q}(\sqrt{-3})$, and $\mathscr{R}$ the complex reflection ring.

Example

```
gap> R:=ContractibleGcomplex("SL(2,O-3)");;
gap> IsRigid(R);
false
gap> S:=BaryCentricSubdivision(R);;
gap> IsRigid(S);
true
gap> C:=TensorWithComplexRepresentationRing(S);;
gap> Homology(C,0);
[ 2, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(C,1);
```

```
[ 0 ]
```

9.3 Crystallographic groups

The following example computes the Bredon homology

$\underline{H}_0(G,\mathscr{R}) = \mathbb{Z}^{17}$

for G the second crystallographic group of dimension 4 in GAP's library of crystallographic groups, and for $\mathscr{R}$ the Burnside ring.

Example

```
gap> G:=SpaceGroup(4,2);;
gap> gens:=GeneratorsOfGroup(G);;
gap> B:=CrystGFullBasis(G);;
gap> R:=CrystGcomplex(gens,B,1);;
gap> IsRigid(R);
false
gap> S:=CrystGcomplex(gens,B,0);;
gap> IsRigid(S);
true
gap> D:=TensorWithBurnsideRing(S);;
gap> Homology(D,0);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Chapter 10

Chain Complexes

HAP uses implementations of chain complexes of free $\mathbb{K}$-modules for each of the rings $\mathbb{K}=\mathbb{Z}$, $\mathbb{K}=\mathbb{Q}$, $\mathbb{K}=\mathbb{F}_p$ with p a prime number, $\mathbb{K}=\mathbb{Z}G$, $\mathbb{K}=\mathbb{F}_pG$ with G a group. The implemented chain complexes have the form

$C_n \xrightarrow{d_n} C_{n-1} \xrightarrow{d_{n-1}} \cdots \xrightarrow{d_2} C_1 \xrightarrow{d_1} C_0 \xrightarrow{d_0} 0$.

Such a complex is said to have *length* n and the rank of the free $\mathbb{K}$-module C_k is referred to as the *dimenion* of the complex in degree k.

For the case $\mathbb{K}=\mathbb{Z}G$ (resp. $\mathbb{K}=\mathbb{F}_pG$) the main focus is on free chain complexes that are exact at each degree k, i.e. $\mathrm{im}(d_{k+1})=\ker(d_k)$, for $0<k<n$ and with $C_0/\mathrm{im}(d_1)\cong\mathbb{Z}$ (resp. $C_0/\mathrm{im}(d_1)\cong\mathbb{F}_p$). We refer to such a chain complex as a *resolution of length* n even though d_n will typically not be injective. More correct terminology would refer to such a chain complex as the first n degrees of a free resolution.

The following sections illustrate some constructions of chain complexes. Constructions for resolutions are described in the next chapter 11.

10.1 Chain complex of a simplicial complex and simplicial pair

The following example constructs the Quillen simplicial complex $Q=\mathscr{A}_p(G)$ for $p=2$ and $G=A_8$; this is the order complex of the poset of non-trivial elementary 2-subgroups of G. The chain complex $C_*=C_*(Q)$ is then computed and seen to have the same number of free generators as Q has simplices. (To ensure indexing of subcomplexes is consistent with that of the large complex it is best to work with vertices represented as integers.)

Example

```
gap> Q:=QuillenComplex(AlternatingGroup(8),2);
Simplicial complex of dimension 3.

gap> C:=ChainComplex(Q);
Chain complex of length 3 in characteristic 0 .

gap> Size(Q);
55015
gap> Size(C);
55015
```

Next the simplicial complex Q is converted to one whose vertices are represented by integers and a contactible subcomplex $L < Q$ is computed. The chain complex $D_* = C_*(Q,L)$ of the simplicial pair (Q,L) is constructed and seen to have the correct size.

Example

```
gap> Q:=IntegerSimplicialComplex(Q);
Simplicial complex of dimension 3.

gap> L:=ContractibleSubcomplex(Q);
Simplicial complex of dimension 3.

gap> D:=ChainComplexOfPair(Q,L);
Chain complex of length 3 in characteristic 0 .

gap> Size(D)=Size(Q)-Size(L);
true
gap> Size(D);
670
gap>
```

The next commands produce a smalled chain complex B_* chain homotopy equivalent to D_* and compute the homology $H_k(Q,\mathbb{Z}) \cong H_k(B_*)$ for $k = 1,2,3$.

Example

```
gap> B:=ContractedComplex(D);
Chain complex of length 3 in characteristic 0 .

gap> Size(B);
64
gap> Homology(B,1);
[  ]
gap> Homology(B,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(B,3);
[  ]
```

10.2 Chain complex of a cubical complex and cubical pair

The following example reads in the digital image

as a 2-dimensional pure cubical complex M and constructs the chain complex $C_* = C_*(M)$.

Example

```
gap> K:=ReadImageAsPureCubicalComplex(file,400);
Pure cubical complex of dimension 2.

gap> C:=ChainComplex(K);
Chain complex of length 2 in characteristic 0 .

gap> Size(C);
```

```
173243
```

Next an acyclic pure cubical subcomplex $L < M$ is computed and the chain complex $D_* = C_*(M,L)$ of the pair is constructed.

Example

```
gap> L:=AcyclicSubcomplexOfPureCubicalComplex(K);
Pure cubical complex of dimension 2.

gap> D:=ChainComplexOfPair(K,L);
Chain complex of length 2 in characteristic 0 .

gap> Size(D);
618
```

Finally the chain complex D_* is simplified to a homotopy equivalent chain complex B_* and the homology $H_1(M,\mathbb{Z}) \cong H_1(B_*)$ is computed.

Example

```
gap> B:=ContractedComplex(D);
Chain complex of length 2 in characteristic 0 .

gap> Size(B);
20
gap> Homology(B,1);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

10.3 Chain complex of a regular CW-complex

The next example constructs a 15-dimensional regular CW-complex Y that is homotopy equivalent to the 2-dimensional torus.

Example

```
gap> Circle:=PureCubicalComplex([[1,1,1,1,1],[1,1,0,1,1],[1,1,1,1,1]]);
Pure cubical complex of dimension 2.

gap> Torus:=DirectProductOfPureCubicalComplexes(Circle,Circle);
Pure cubical complex of dimension 4.

gap> CTorus:=CechComplexOfPureCubicalComplex(Torus);
Simplicial complex of dimension 15.

gap> Y:=RegularCWComplex(CTorus);
Regular CW-complex of dimension 15
```

Next the cellular chain complex $C_* = C_*(Y)$ is constructed. Also, a minimally generated chain complex $D_* = C_*(Y')$ of a non-regular CW-complex $Y' \simeq Y$ is constructed.

Example

```
gap> C:=ChainComplexOfRegularCWComplex(Y);
Chain complex of length 15 in characteristic 0 .

gap> Size(C);
1172776

gap> D:=ChainComplex(Y);
Chain complex of length 15 in characteristic 0 .

gap> Size(D);
4
```

10.4 Chain Maps of simplicial and regular CW maps

The next example realizes the complement of the first prime knot on 11 crossings as a pure permutahedral complex. The complement is converted to a regular CW-complex Y and the boundary inclusion $f:\partial Y \hookrightarrow Y$ is constructed as a map of regular CW-complexes. Then the induced chain map $F:C_*(\partial Y) \hookrightarrow C_*(Y)$ is constructed. Finally the homology homomorphism $H_1(F):H_1(C_*(\partial Y)) \to H_1(C_*(Y))$ is computed.

Example

```
gap> K:=PurePermutahedralKnot(11,1);;
gap> M:=PureComplexComplement(K);
Pure permutahedral complex of dimension 3.

gap> Y:=RegularCWComplex(M);
Regular CW-complex of dimension 3

gap> f:=BoundaryMap(Y);
Map of regular CW-complexes

gap> F:=ChainMap(f);
Chain Map between complexes of length 2 .

gap> H:=Homology(F,1);
[ g1, g2 ] -> [ g1^-1, g1^-1 ]

gap> Kernel(H);
Pcp-group with orders [ 0 ]
```

The command `ChainMap(f)` can be used to construct the chain map $C_*(K) \to C_*(K')$ induced by a map $f:K \to K'$ of simplicial complexes.

10.5 Constructions for chain complexes

It is straightforward to implement basic constructions on chain complexes. A few constructions are illustrated in the following example.

Example

```
gap> res:=ResolutionFiniteGroup(SymmetricGroup(5),5);;
gap> C:=TensorWithIntegers(res);
Chain complex of length 5 in characteristic 0 .

gap> D:=ContractedComplex(C);#A chain homotopic complex
Chain complex of length 5 in characteristic 0 .
gap> List([0..5],C!.dimension);
[ 1, 4, 10, 20, 35, 56 ]
gap> List([0..5],D!.dimension);
[ 1, 1, 2, 4, 6, 38 ]

gap> CxC:=TensorProduct(C,C);
Chain complex of length 10 in characteristic 0 .

gap> SC:=SuspendedChainComplex(C);
Chain complex of length 6 in characteristic 0 .

gap> RC:=ReducedSuspendedChainComplex(C);
Chain complex of length 6 in characteristic 0 .

gap> PC:=PathObjectForChainComplex(C);
Chain complex of length 5 in characteristic 0 .

gap> dualC:=HomToIntegers(C);
Cochain complex of length 5 in characteristic 0 .

gap> Cxp:=TensorWithIntegersModP(C,5);
Chain complex of length 5 in characteristic 5 .

gap> CxQ:=TensorWithRationals(C); #The quirky -1/2 denotes rationals
Chain complex of length 5 in characteristic -1/2 .
```

10.6 Filtered chain complexes

A sequence of inclusions of chain complexes $C_{0,*} \leq C_{1,*} \leq \cdots \leq C_{T-1,*} \leq C_{T,*}$ in which the preferred basis of $C_{k-1,\ell}$ is the beginning of the preferred basis of $C_{k,\ell}$ is referred to as a *filtered chain complex*. Filtered chain complexes give rise to spectral sequences such as the *equivariant spectral sequence* of a $G-CW$-complex with subgroup $H < G$. A particular case is the Lyndon-Hochschild-Serre spectral sequence for the homology of a group extension $N \rightarrowtail G \twoheadrightarrow Q$ with $E^2_{p,q} = H_p(Q, H_q(N, \mathbb{Z}))$.

The following commands construct the filtered chain complex underlying the Lyndon-Hochschild-Serre spectral sequence for the dihedral group $G = D_{32}$ of order 64 and its centre $N = Z(G)$.

Example

```
gap> G:=DihedralGroup(64);;
gap> N:=Center(G);;
gap> R:=ResolutionNormalSeries([G,N],3);;
gap> C:=FilteredTensorWithIntegersModP(R,2);
Chain complex of length 3 in characteristic 2 .
```

The differentials $d^r_{p,q}$ in a given page E^r of the spectral sequence arise from the induced homology homomorphisms $\iota_\ell^{s,t}: H_\ell(C_{s,*}) \to H_\ell(C_{t,*})$ for $s \leq t$. Textbooks traditionally picture the differential in E^r as an array of sloping arrows with non-zero groups $E^r_{p,q} \neq 0$ represented by dots. An alternative representation of this information is as a barcode (of the sort used in Topological Data Analysis). The homomorphisms $\iota_2^{*,*}$ in the example, with coefficients converted to mod 2, are pictured by the bar code

which was produced by the following commands.

Example

```
gap> p:=2;;k:=2;;
gap> P:=PersistentHomologyOfFilteredChainComplex(C,k,p);;
gap> BarCodeDisplay(P);
```

Let us view a barcode as a graph with vertices arranged in columns. We then refer to a connected component of this graph as a *bar*. Let us say that any bar with a vertex in the final (right-hand) column is of *length* ∞. Let us define the *length* of any other bar to be $r-1$ where r is the number of vertices in the bar. Theorem 3.1 in [RHM+13] implies that the differential $d^r_{p,q} = 0$ for $p+q = k$ if and only if there is no bar of length r in the barcode arising in this way for any degree k homology barcode. So the following commands demonstrate that $d^r = 0$ for $r \geq 2$ at least for $k \leq 7$.

Example

```
gap> for k in [1..7] do
> R:=ResolutionNormalSeries([G,N],k+1);;
> C:=FilteredTensorWithIntegersModP(R,2);
> P:=PersistentHomologyOfFilteredChainComplex(C,k,2);;
> BarCodeDisplay(P);
> od;
```

10.7 Sparse chain complexes

Boundary homomorphisms in all of the above examples of chain complexes are represented by matrices. In cases where the matrices are large and have many zero entries it is better to use sparse matrices.

The following commands demonstrate the conversion of the matrix

$$A = \begin{pmatrix} 0 & 2 & 0 \\ -3 & 0 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

to sparse form, and vice-versa.

Example

```
gap> A:=[[0,2,0],[-3,0,0],[0,0,4]];;
gap> S:=SparseMat(A);
Sparse matrix with 3 rows and 3 columns in characteristic 0

gap> NamesOfComponents(S);
[ "mat", "characteristic", "rows", "cols" ]
gap> S!.mat;
```

```
[ [ [ 2, 2 ] ], [ [ 1, -3 ] ], [ [ 3, 4 ] ] ]

gap> B:=SparseMattoMat(S);
[ [ 0, 2, 0 ], [ -3, 0, 0 ], [ 0, 0, 4 ] ]
```

To illustrate the use of sparse chain complexes we consider the data points represented in the following digital image.

The following commands read in this image as a 2-dimensional pure cubical complex and store the Euclidean coordinates of the black pixels in a list. Then 200 points are selected at random from this list and used to construct a 200×200 symmetric matrix S whose entries are the Euclidean distance between the sample data points.

Example

```
gap> file:=HapFile("data500.png");;
gap> M:=ReadImageAsPureCubicalComplex(file,400);;
gap> A:=M!.binaryArray;;
gap> data:=[];;
gap> for i in [1..Length(A)] do
> for j in [1..Length(A[1])] do
> if A[i][j]=1 then Add(data,[i,j]); fi;
> od;
> od;
gap> sample:=List([1..200],i->Random(data));;
gap> S:=VectorsToSymmetricMatrix(sample,EuclideanApproximatedMetric);;
```

The symmetric distance matrix S is next converted to a filtered chain complex arising from a filtered simplicial complex (using the standard *persistent homology* pipeline).

Example

```
gap> G:=SymmetricMatrixToFilteredGraph(S,10,100);;
#Filtration length T=10, distances greater than 100 discarded.
gap> N:=SimplicialNerveOfFilteredGraph(G,2);;
gap> C:=SparseFilteredChainComplexOfFilteredSimplicialComplex(N);;
Filtered sparse chain complex of length 2 in characteristic 0 .
```

Next, the induced homology homomorphisms in degrees 1 and 2, with rational coefficients, are computed and displayed a barcodes.

Example

```
gap> P0:=PersistentHomologyOfFilteredSparseChainComplex(C,0);;
gap> P1:=PersistentHomologyOfFilteredSparseChainComplex(C,1);;
gap> BarCodeCompactDisplay(P0);
```

Example

```
gap> BarCodeCompactDisplay(P1);
```

The barcodes are consistent with the data points having been sampled from a space with the homotopy type of an annulus.

Chapter 11

Resolutions

There is a range of functions in HAP that input a group G, integer n, and attempt to return the first n terms of a free $\mathbb{Z}G$-resolution R_* of the trivial module $\mathbb{Z}$. In some cases an explicit contracting homotopy is provided on the resolution. The function `Size(R)` returns a list whose kth term is the sum of the lengths of the boundaries of the generators in degree k.

11.1 Resolutions for small finite groups

The following uses discrete Morse theory to construct a resolution.

Example

```
gap> G:=SymmetricGroup(6);; n:=6;;
gap> R:=ResolutionFiniteGroup(G,n);
Resolution of length 6 in characteristic 0 for Group([ (1,2), (1,2,3,4,5,6)
 ]) .

gap> Size(R);
[ 10, 58, 186, 452, 906, 1436 ]
```

11.2 Resolutions for very small finite groups

The following uses linear algebra over $\mathbb{Z}$ to construct a resolution.

Example

```
gap> Q:=QuaternionGroup(128);;
gap> R:=ResolutionSmallGroup(Q,20);
Resolution of length 20 in characteristic 0 for <pc group of size 128 with
2 generators> .
No contracting homotopy available.

gap> Size(R);
[ 4, 42, 8, 128, 4, 42, 8, 128, 4, 42, 8, 128, 4, 42, 8, 128, 4, 42, 8, 128 ]
```

The suspicion that this resolution R_* is periodic of period 4 can be confirmed by constructing the chain complex $C_* = R_* \otimes_{\mathbb{Z}} \mathbb{Z}G$ and verifying that boundary matrices repeat with period 4.

A second example of a periodic resolution, for the Dihedral group $D_{2k+1} = \langle x, y \mid x^2 = xy^k x^{-1} y^{-k-1} = 1 \rangle$ of order $2k+2$ in the case $k = 1$, is constructed and verified for periodicity in the next example.

Example

```
gap> F:=FreeGroup(2);;D:=F/[F.1^2,F.1*F.2*F.1^-1*F.2^-2];;
gap> R:=ResolutionSmallGroup(D,15);;
gap> Size(R);
[ 4, 7, 8, 6, 4, 8, 8, 6, 4, 8, 8, 6, 4, 8, 8 ]
gap> C:=TensorWithIntegersOverSubgroup(R,Group(One(D)));;
gap> n:=4;;BoundaryMatrix(C,n)=BoundaryMatrix(C,n+4);
true
gap> n:=5;;BoundaryMatrix(C,n)=BoundaryMatrix(C,n+4);
true
gap> n:=6;;BoundaryMatrix(C,n)=BoundaryMatrix(C,n+4);
true
gap> n:=7;;BoundaryMatrix(C,n)=BoundaryMatrix(C,n+4);
true
gap> n:=8;;BoundaryMatrix(C,n)=BoundaryMatrix(C,n+4);
true
```

This periodic resolution for D_3 can be found in a paper by R. Swan [Swa60]. The resolution was proved for arbitrary D_{2k+1} by Irina Kholodna [Kho01] (Corollary 5.5) and is the cellular chain complex of the universal cover of a CW-complex X with two cells in dimensions $1,2$ mod 4 and one cell in dimensions $0,3$ mod 4. The 2-skelecton is the 2-complex for the given presentation of D_{2k+1} and an attaching map for the 3-cell is represented as follows.

A slightly different periodic resolution for D_{2k+1} has been obtain more recently by FEA Johnson [Joh16]. Johnson's resolution has two free generators in each degree. Interestingly, running the following code for many values of $k > 1$ seems to produce a periodic resolution with two free generators in each degree for most values of k.

Example

```
gap> k:=20;;rels:=[x^2,x*y^k*x^-1*y^(-1-k)];;D:=F/rels;;
gap> R:=ResolutionSmallGroup(D,7);;
gap> List([0..7],R!.dimension);
[ 1, 2, 2, 2, 2, 2, 2, 2 ]
```

The performance of the function `ResolutionSmallGroup(G,n)` is very sensistive to the choice of presentation for the input group G. If G is an fp-group then the defining presentation for G is used. If G is a permutaion group or finite matrix group then GAP functions are invoked to find a presentation for G. The following commands use a geometrically derived presentation for $SL(2,5)$ as input in order to obtain the first few terms of a periodic resolution for this group of period 4.

Example

```
gap> Y:=PoincareDodecahedronCWComplex(
> [[1,2,3,4,5],[6,7,8,9,10]],
> [[1,11,16,12,2],[19,9,8,18,14]],
> [[2,12,17,13,3],[20,10,9,19,15]],
> [[3,13,18,14,4],[16,6,10,20,11]],
> [[4,14,19,15,5],[17,7,6,16,12]],
> [[5,15,20,11,1],[18,8,7,17,13]]);;
```

```
gap> G:=FundamentalGroup(Y);
<fp group on the generators [ f1, f2 ]>
gap> RelatorsOfFpGroup(G);
[ f2^-1*f1^-1*f2*f1^-1*f2^-1*f1, f2^-1*f1*f2^2*f1*f2^-1*f1^-1 ]
gap> StructureDescription(G);
"SL(2,5)"
gap> R:=ResolutionSmallGroup(G,3);;
gap> List([0..3],R!.dimension);
[ 1, 2, 2, 1 ]
```

11.3 Resolutions for finite groups acting on orbit polytopes

The following uses Polymake convex hull computations and homological perturbation theory to construct a resolution.

Example

```
gap> G:=SignedPermutationGroup(5);;
gap> StructureDescription(G);
"C2 x ((C2 x C2 x C2 x C2) : S5)"

gap> v:=[1,2,3,4,5];;  #The resolution depends on the choice of vector.
gap> P:=PolytopalComplex(G,[1,2,3,4,5]);
Non-free resolution in characteristic 0 for <matrix group of size 3840 with
9 generators> .
No contracting homotopy available.

gap> R:=FreeGResolution(P,6);
Resolution of length 5 in characteristic 0 for <matrix group of size
3840 with 9 generators> .
No contracting homotopy available.
gap> Size(R);
[ 10, 60, 214, 694, 6247, 273600 ]
```

The convex polytope $P_G(v) = \text{Convex Hull}\{g \cdot v \mid g \in G\}$ used in the resolution depends on the choice of vector $v \in \mathbb{R}^n$. Two such polytopes for the alternating group $G = A_4$ acting on $\mathbb{R}^4$ can be visualized as follows.

Example

```
gap> G:=AlternatingGroup(4);;
gap> OrbitPolytope(G,[1,2,3,4],["VISUAL"]);
gap> OrbitPolytope(G,[1,1,3,4],["VISUAL"]);

gap> P1:=PolytopalComplex(G,[1,2,3,4]);;
gap> P2:=PolytopalComplex(G,[1,1,3,4]);;
gap> R1:=FreeGResolution(P1,20);;
gap> R2:=FreeGResolution(P2,20);;
gap> Size(R1);
[ 6, 11, 32, 24, 36, 60, 65, 102, 116, 168, 172, 248, 323, 628, 650, 1093,
  1107, 2456, 2344, 6115 ]
gap> Size(R2);
[ 4, 11, 20, 24, 36, 60, 65, 102, 116, 168, 172, 248, 323, 628, 650, 1093,
```

```
  1107, 2456, 2344, 6115 ]
```

11.4 Minimal resolutions for finite p-groups over $\mathbb{F}_p$

The following uses linear algebra to construct a minimal free $\mathbb{F}_pG$-resolution of the trivial module $\mathbb{F}$.

Example

```
gap> P:=SylowSubgroup(MathieuGroup(12),2);;
gap> R:=ResolutionPrimePowerGroup(P,20);
Resolution of length 20 in characteristic 2 for Group(
[ (2,8,4,12)(3,11,7,9), (2,3)(4,7)(6,10)(9,11), (3,7)(6,10)(8,11)(9,12),
  (1,10)(3,7)(5,6)(8,12), (2,4)(3,7)(8,12)(9,11), (1,5)(6,10)(8,12)(9,11)
 ]) .

gap> Size(R);
[ 6, 62, 282, 740, 1810, 3518, 6440, 10600, 17040, 24162, 34774, 49874,
  62416, 81780, 106406, 145368, 172282, 208926, 262938, 320558 ]
```

The resolution has the minimum number of generators possible in each degree and can be used to guess a formula for the Poincare series

$P(x) = \Sigma_{k\geq 0} \dim_{\mathbb{F}_p} H^k(G,\mathbb{F}_p)x^k$.

The guess is certainly correct for the coefficients of x^k for $k \leq 20$ and can be used to guess the dimension of say $H^{2000}(G,\mathbb{F}_p)$.

Most likely $\dim_{\mathbb{F}_2} H^{2000}(G,\mathbb{F}_2) = 2001000$.

Example

```
gap> P:=PoincareSeries(R,20);
(1)/(-x_1^3+3*x_1^2-3*x_1+1)

gap> ExpansionOfRationalFunction(P,2000)[2000];
2001000
```

11.5 Resolutions for abelian groups

The following uses the formula for the tensor product of chain complexes to construct a resolution.

Example

```
gap> A:=AbelianPcpGroup([2,4,8,0,0]);;
gap> StructureDescription(A);
"Z x Z x C8 x C4 x C2"

gap> R:=ResolutionAbelianGroup(A,10);
Resolution of length 10 in characteristic 0 for Pcp-group with orders
[ 2, 4, 8, 0, 0 ] .

gap> Size(R);
[ 14, 90, 296, 680, 1256, 2024, 2984, 4136, 5480, 7016 ]
```

11.6 Resolutions for nilpotent groups

The following uses the NQ package to express the free nilpotent group of class 3 on three generators as a Pcp group G, and then uses homological perturbation on the lower central series to construct a resolution. The resolution is used to exhibit 2-torsion in $H_4(G,\mathbb{Z})$.

Example

```
gap> F:=FreeGroup(3);;
gap> G:=Image(NqEpimorphismNilpotentQuotient(F,3));;
gap> R:=ResolutionNilpotentGroup(G,5);
Resolution of length 5 in characteristic 0 for Pcp-group with orders
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] .

gap> Size(R);
[ 28, 377, 2377, 9369, 25850 ]

gap> Homology(TensorWithIntegers(R),4);
[ 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

The following example uses a simplification procedure for resolutions to construct a resolution S_* for the free nilpotent group G of class 2 on 3 generators that has the minimal possible number of free generators in each degree.

Example

```
gap> G:=Image(NqEpimorphismNilpotentQuotient(FreeGroup(3),2));;
gap> R:=ResolutionNilpotentGroup(G,10);;
gap> S:=ContractedComplex(R);;
gap> C:=TensorWithIntegers(S);;
gap> List([1..10],i->IsZero(BoundaryMatrix(C,i)));
[ true, true, true, true, true, true, true, true, true, true ]
```

The following example uses homological perturbation on the lower central series to construct a resolution for the Sylow 2-subgroup $P = Syl_2(M_{12})$ of the Mathieu simple group M_{12}.

Example

```
gap> G:=MathieuGroup(12);;
gap> P:=SylowSubgroup(G,2);;
gap> StructureDescription(P);
"((C4 x C4) : C2) : C2"

gap> R:=ResolutionNilpotentGroup(P,9);
Resolution of length 9 in characteristic
0 for <permutation group with 279 generators> .

gap> Size(R);
[ 12, 80, 310, 939, 2556, 6768, 19302, 61786, 237068 ]
```

11.7 Resolutions for groups with subnormal series

The following uses homological perturbation on a subnormal series to construct a resolution for the Sylow 2-subgroup $P = Syl_2(M_{12})$ of the Mathieu simple group M_{12}.

Example

```
gap> P:=SylowSubgroup(MathieuGroup(12),2);;
gap> sn:=ElementaryAbelianSeries(P);;
gap> R:=ResolutionSubnormalSeries(sn,9);
Resolution of length 9 in characteristic
0 for <permutation group with 64 generators> .

gap> Size(R);
[ 12, 78, 288, 812, 1950, 4256, 8837, 18230, 39120 ]
```

11.8 Resolutions for groups with normal series

The following uses homological perturbation on a normal series to construct a resolution for the Sylow 2-subgroup $P = Syl_2(M_{12})$ of the Mathieu simple group M_{12}.

Example

```
gap> P:=SylowSubgroup(MathieuGroup(12),2);;
gap> P1:=EfficientNormalSubgroups(P)[1];;
gap> P2:=Intersection(DerivedSubgroup(P),P1);;
gap> P3:=Group(One(P));;
gap> R:=ResolutionNormalSeries([P,P1,P2,P3],9);
Resolution of length 9 in characteristic
0 for <permutation group with 64 generators> .

gap> Size(R);
[ 10, 60, 200, 532, 1238, 2804, 6338, 15528, 40649 ]
```

11.9 Resolutions for polycyclic (almost) crystallographic groups

The following uses the Polycyclic package and homological perturbation to construct a resolution for the crystallographic group `G:=SpaceGroup(3,165)`.

Example

```
gap> G:=SpaceGroup(3,165);;
gap> G:=Image(IsomorphismPcpGroup(G));;
gap> R:=ResolutionAlmostCrystalGroup(G,20);
Resolution of length 20 in characteristic 0 for Pcp-group with orders
[ 3, 2, 0, 0, 0 ] .

gap> Size(R);
[ 10, 49, 117, 195, 273, 351, 429, 507, 585, 663, 741, 819, 897, 975, 1053,
  1131, 1209, 1287, 1365, 1443 ]
```

The following constructs a resolution for an almost crystallographic Pcp group G. The final commands establish that G is not isomorphic to a crystallographic group.

Example

```
gap> G:=AlmostCrystallographicPcpGroup( 4, 50, [ 1, -4, 1, 2 ] );;
gap> R:=ResolutionAlmostCrystalGroup(G,20);
Resolution of length 20 in characteristic 0 for Pcp-group with orders
[ 4, 0, 0, 0, 0 ] .

gap> Size(R);
[ 10, 53, 137, 207, 223, 223, 223, 223, 223, 223, 223, 223, 223, 223,
  223, 223, 223, 223, 223 ]


gap> T:=Kernel(NaturalHomomorphismOnHolonomyGroup(G));;
gap> IsAbelian(T);
false
```

11.10 Resolutions for Bieberbach groups

The following constructs a resolution for the Bieberbach group `G=SpaceGroup(3,165)` by using convex hull algorithms to construct a Dirichlet domain for its free action on Euclidean space $\mathbb{R}^3$. By construction the resolution is trivial in degrees ≥ 3.

Example

```
gap> G:=SpaceGroup(3,165);;
gap> R:=ResolutionBieberbachGroup(G);
Resolution of length 4 in characteristic
0 for SpaceGroupOnRightBBNWZ( 3, 6, 1, 1, 4 ) .
No contracting homotopy available.

gap> Size(R);
[ 10, 18, 8, 0 ]
```

The fundamental domain constructed for the above resolution can be visualized using the following commands.

Example

```
gap> F:=FundamentalDomainBieberbachGroup(G);
<polymake object>
gap> Display(F);
```

A different fundamental domain and resolution for G can be obtained by changing the choice of vector $v \in \mathbb{R}^3$ in the definition of the Dirichlet domain

$D(v) = \{x \in \mathbb{R}^3 \mid ||x-v|| \leq ||x-g.v|| \text{ for all } g \in G\}$.

Example

```
gap> R:=ResolutionBieberbachGroup(G,[1/2,1/2,1/2]);
Resolution of length 4 in characteristic
0 for SpaceGroupOnRightBBNWZ( 3, 6, 1, 1, 4 ) .
No contracting homotopy available.

gap> Size(R);
```

```
[ 28, 42, 16, 0 ]

gap> F:=FundamentalDomainBieberbachGroup(G);
<polymake object>
gap> Display(F);
```

A higher dimensional example is handled in the next session. A list of the 62 7-dimensional Hantze-Wendt Bieberbach groups is loaded and a resolution is computed for the first group in the list.

Example

```
gap> file:=HapFile("HW-7dim.txt");;
gap> Read(file);
gap> G:=HW07Gr[1];
<matrix group with 7 generators>

gap> R:=ResolutionBieberbachGroup(G);
Resolution of length 8 in characteristic 0 for <matrix group with
7 generators> .
No contracting homotopy available.

gap> Size(R);
[ 284, 1512, 3780, 4480, 2520, 840, 84, 0 ]
```

The homological perturbation techniques needed to extend this method to crystallographic groups acting non-freely on $\mathbb{R}^n$ has not yet been implemenyed. This is on the TO-DO list.

11.11 Resolutions for arbitrary crystallographic groups

An implementation of the above method for Bieberbach groups is also available for arbitrary crystallographic groups. The following example constructs a resolution for the group `G:=SpaceGroupIT(3,227)`.

Example

```
gap> G:=SpaceGroupIT(3,227);;
gap> R:=ResolutionSpaceGroup(G,11);
Resolution of length 11 in characteristic 0 for <matrix group with
8 generators> .
No contracting homotopy available.

gap> Size(R);
[ 38, 246, 456, 644, 980, 1427, 2141, 2957, 3993, 4911, 6179 ]
```

11.12 Resolutions for crystallographic groups admitting cubical fundamental domain

The following uses subdivision techniques to construct a resolution for the Bieberbach group `G:=SpaceGroup(4,122)`. The resolution is endowed with a contracting homotopy.

Example

```
gap> G:=SpaceGroup(4,122);;
gap> R:=ResolutionCubicalCrystGroup(G,20);
Resolution of length 20 in characteristic 0 for <matrix group with
6 generators> .

gap> Size(R);
[ 8, 24, 24, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Subdivision and homological perturbation are used to construct the following resolution (with contracting homotopy) for a crystallographic group with non-free action.

Example

```
gap> G:=SpaceGroup(4,1100);;
gap> R:=ResolutionCubicalCrystGroup(G,20);
Resolution of length 20 in characteristic 0 for <matrix group with
8 generators> .

gap> Size(R);
[ 40, 215, 522, 738, 962, 1198, 1466, 1734, 2034, 2334, 2666, 2998, 3362,
  3726, 4122, 4518, 4946, 5374, 5834, 6294 ]
```

11.13 Resolutions for Coxeter groups

The following session constructs the Coxeter diagram for the Coxeter group $B = B_7$ of order 645120. A resolution for G is then computed.

Example

```
gap> D:=[[1,[2,3]],[2,[3,3]],[3,[4,3]],[4,[5,3]],[5,[6,3]],[6,[7,4]]];;
gap> CoxeterDiagramDisplay(D);;
```

Example

```
gap> R:=ResolutionCoxeterGroup(D,5);
Resolution of length 5 in characteristic
0 for <permutation group of size 645120 with 7 generators> .
No contracting homotopy available.

gap> Size(R);
[ 14, 112, 492, 1604, 5048 ]
```

The routine extension of this method to infinite Coxeter groups is on the TO-DO list.

11.14 Resolutions for Artin groups

The following session constructs a resolution for the infinite Artin group G associated to the Coxeter group B_7. Exactness of the resolution depends on the solution to the $K(\pi,1)$ Conjecture for Artin groups of spherical type.

Example

```
gap> R:=ResolutionArtinGroup(D,8);
Resolution of length 8 in characteristic 0 for <fp group on the generators
[ f1, f2, f3, f4, f5, f6, f7 ]> .
No contracting homotopy available.

gap> Size(R);
[ 14, 98, 310, 610, 918, 1326, 2186, 0 ]
```

11.15 Resolutions for $G = SL_2(\mathbb{Z}[1/m])$

The following uses homological perturbation to construct a resolution for $G = SL_2(\mathbb{Z}[1/6])$.

Example

```
gap> R:=ResolutionSL2Z(6,10);
Resolution of length 10 in characteristic 0 for SL(2,Z[1/6]) .

gap> Size(R);
[ 44, 679, 6910, 21304, 24362, 48506, 43846, 90928, 86039, 196210 ]
```

11.16 Resolutions for selected groups $G = SL_2(\mathcal{O}(\mathbb{Q}(\sqrt{d}))$

The following uses finite "Voronoi complexes" and homological perturbation to construct a resolution for $G = SL_2(\mathcal{O}(\mathbb{Q}(\sqrt{-5}))$. The finite complexes were contributed independently by A. Rahm, M. Dutour-Scikiric and S. Schoenenbeck and are stored in the folder `~pkg/Hap1.v/lib/Perturbations/Gcomplexes`.

Example

```
gap> R:=ResolutionSL2QuadraticIntegers(-5,10);
Resolution of length 10 in characteristic 0 for matrix group .
No contracting homotopy available.

gap> Size(R);
[ 22, 114, 120, 200, 146, 156, 136, 254, 168, 170 ]
```

11.17 Resolutions for selected groups $G = PSL_2(\mathcal{O}(\mathbb{Q}(\sqrt{d}))$

The following uses finite "Voronoi complexes" and homological perturbation to construct a resolution for $G = PSL_2(\mathcal{O}(\mathbb{Q}(\sqrt{-11}))$. The finite complexes were contributed independently by A. Rahm, M. Dutour-Scikiric and S. Schoenenbeck and are stored in the folder `~pkg/Hap1.v/lib/Perturbations/Gcomplexes`.

Example

```
gap> R:=ResolutionPSL2QuadraticIntegers(-11,10);
Resolution of length 10 in characteristic 0 for PSL(2,O-11) .
No contracting homotopy available.
```

```
gap> Size(R);
[ 12, 59, 89, 107, 125, 230, 208, 270, 326, 515 ]
```

11.18 Resolutions for a few higher-dimensional arithmetic groups

The following uses finite "Voronoi complexes" and homological perturbation to construct a resolution for $G = PSL_4(\mathbb{Z})$. The finite complexes were contributed by M. Dutour-Scikiric and are stored in the folder `~pkg/Hap1.v/lib/Perturbations/Gcomplexes`.

Example

```
gap>  V:=ContractibleGcomplex("PSL(4,Z)_d");
Non-free resolution in characteristic 0 for matrix group .
No contracting homotopy available.

gap> R:=FreeGResolution(V,5);
Resolution of length 5 in characteristic 0 for matrix group .
No contracting homotopy available.

gap> Size(R);
[ 18, 210, 1444, 26813 ]
```

11.19 Resolutions for finite-index subgroups

The next commands first construct the congruence subgroup $\Gamma_0(I)$ of index 144 in $SL_2(\mathcal{O}\mathbb{Q}(\sqrt{-2}))$ for the ideal I in $\mathcal{O}\mathbb{Q}(\sqrt{-2})$ generated by $4+5\sqrt{-2}$. The commands then compute a resolution for the congruence subgroup $G = \Gamma_0(I) \leq SL_2(\mathcal{O}\mathbb{Q}(\sqrt{-2}))$

Example

```
gap> Q:=QuadraticNumberField(-2);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,4+5*Sqrt(-2));;
gap> G:=HAP_CongruenceSubgroupGamma0(I);
<[group of 2x2 matrices in characteristic 0>
gap>
gap> IndexInSL2O(G);
144
gap> R:=ResolutionSL2QuadraticIntegers(-2,4,true);;
gap> S:=ResolutionFiniteSubgroup(R,G);
Resolution of length 4 in characteristic 0 for <matrix group with
290 generators> .

gap> Size(S);
[ 1152, 8496, 30960, 59616 ]
```

11.20 Simplifying resolutions

The next commands construct a resolution R_* for the symmetric group S_5 and convert it to a resolution S_* for the finite index subgroup $A_4 < S_5$. An heuristic algorithm is applied to S_* in the hope of obtaining a smaller resolution T_* for the alternating group A_4.

Example

```
gap> R:=ResolutionFiniteGroup(SymmetricGroup(5),5);;
gap> S:=ResolutionFiniteSubgroup(R,AlternatingGroup(4));
Resolution of length 5 in characteristic 0 for Alt( [ 1 .. 4 ] ) .

gap> Size(S);
[ 80, 380, 1000, 2040, 3400 ]
gap> T:=SimplifiedComplex(S);
Resolution of length 5 in characteristic 0 for Alt( [ 1 .. 4 ] ) .

gap> Size(T);
[ 4, 34, 22, 19, 196 ]
```

11.21 Resolutions for graphs of groups and for groups with aspherical presentations

The following example constructs a resolution for a finitely presented group whose presentation is known to have the property that its associated 2-complex is aspherical.

Example

```
gap> F:=FreeGroup(3);;x:=F.1;;y:=F.2;;z:=F.3;;
gap> rels:=[x*y*x*(y*x*y)^-1, y*z*y*(z*y*z)^-1, z*x*z*(x*z*x)^-1];;
gap> G:=F/rels;;
gap> R:=ResolutionAsphericalPresentation(G,10);
Resolution of length 10 in characteristic 0 for <fp group on the generators
[ f1, f2, f3 ]> .
No contracting homotopy available.

gap> Size(R);
[ 6, 18, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

The following commands create a resolution for a graph of groups corresponding to the amalgamated product $G = H *_A K$ where $H = S_5$ is the symmetric group of degree 5, $K = S_4$ is the symmetric group of degree 4 and the common subgroup is $A = S_3$.

Example

```
gap> S5:=SymmetricGroup(5);SetName(S5,"S5");;
Sym( [ 1 .. 5 ] )
gap> S4:=SymmetricGroup(4);SetName(S4,"S4");;
Sym( [ 1 .. 4 ] )
gap> A:=SymmetricGroup(3);SetName(A,"S3");;
Sym( [ 1 .. 3 ] )
gap> AS5:=GroupHomomorphismByFunction(A,S5,x->x);;
gap> AS4:=GroupHomomorphismByFunction(A,S4,x->x);;
gap> D:=[S5,S4,[AS5,AS4]];;
```

```
gap> GraphOfGroupsDisplay(D);;
```

Example

```
gap> R:=ResolutionGraphOfGroups(D,8);;
gap> Size(R);
[ 16, 68, 162, 302, 480, 627, 869, 1290 ]
```

11.22 Resolutions for $\mathbb{F}G$-modules

Let $\mathbb{F} = \mathbb{F}_p$ be the field of p elements and let M be some $\mathbb{F}G$-module for G a finite p-group. We might wish to construct a free $\mathbb{F}G$-resolution for M. We can handle this by constructing a short exact sequence

$DM \rightarrowtail P \twoheadrightarrow M$

in which P is free (or projective). Then any resolution of DM yields a resolution of M and we can represent DM as a submodule of P. We refer to DM as the *desuspension* of M. Consider for instance $G = Syl_2(GL(4,2))$ and $\mathbb{F} = \mathbb{F}_2$. The matrix group G acts via matrix multiplication on $M = \mathbb{F}^4$. The following example constructs a free $\mathbb{F}G$-resolution for M.

Example

```
gap> G:=GL(4,2);;
gap> S:=SylowSubgroup(G,2);;
gap> M:=GModuleByMats(GeneratorsOfGroup(S),GF(2));;
gap> DM:=DesuspensionMtxModule(M);;
gap> R:=ResolutionFpGModule(DM,20);
Resolution of length 20 in characteristic 2 for <matrix group of
size 64 with 3 generators> .

gap> List([0..20],R!.dimension);
[ 3, 6, 10, 15, 21, 28, 36, 45, 55, 66, 78, 91, 105, 120, 136,
153, 171, 190, 210, 231, 253 ]
```

Chapter 12

Simplicial groups

12.1 Crossed modules

A *crossed module* consists of a homomorphism of groups $\partial: M \to G$ together with an action $(g,m) \mapsto {}^g m$ of G on M satisfying

1. $\partial({}^g m) = gmg^{-1}$
2. ${}^{\partial m} m' = mm'm^{-1}$

for $g \in G$, $m, m' \in M$.

A crossed module $\partial: M \to G$ is equivalent to a cat^1-group (H,s,t) (see 6.11) where $H = M \rtimes G$, $s(m,g) = (1,g)$, $t(m,g) = (1,(\partial m)g)$. A cat^1-group is, in turn, equivalent to a simplicial group with Moore complex has length 1. The simplicial group is constructed by considering the cat^1-group as a category and taking its nerve. Alternatively, the simplicial group can be constructed by viewing the crossed module as a crossed complex and using a nonabelian version of the Dold-Kan theorem.

The following example concerns the crossed module

$\partial: G \to Aut(G), g \mapsto (x \mapsto gxg^{-1})$

associated to the dihedral group G of order 16. This crossed module represents, up to homotopy type, a connected space X with $\pi_i X = 0$ for $i \geq 3$, $\pi_2 X = Z(G)$, $\pi_1 X = Aut(G)/Inn(G)$. The space X can be represented, up to homotopy, by a simplicial group. That simplicial group is used in the example to compute

$H_1(X,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2$,
$H_2(X,\mathbb{Z}) = \mathbb{Z}_2$,
$H_3(X,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$,
$H_4(X,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$,
$H_5(X,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2$.

Example

```
gap> C:=AutomorphismGroupAsCatOneGroup(DihedralGroup(16));
Cat-1-group with underlying group Group(
[ f1, f2, f3, f4, f5, f6, f7, f8, f9 ] ) .

gap> Size(C);
512
gap> Q:=QuasiIsomorph(C);
Cat-1-group with underlying group Group( [ f9, f8, f1, f2*f3, f5 ] ) .
```

```
gap> Size(Q);
32

gap> N:=NerveOfCatOneGroup(Q,6);
Simplicial group of length 6

gap> K:=ChainComplexOfSimplicialGroup(N);
Chain complex of length 6 in characteristic 0 .

gap> Homology(K,1);
[ 2, 2 ]
gap> Homology(K,2);
[ 2 ]
gap> Homology(K,3);
[ 2, 2, 2 ]
gap> Homology(K,4);
[ 2, 2, 2 ]
gap> Homology(K,5);
[ 2, 2, 2, 2, 2, 2 ]
```

12.2 Eilenberg-MacLane spaces as simplicial groups (not recommended)

The following example concerns the Eilenberg-MacLane space $X = K(\mathbb{Z}_3,3)$ which is a path-connected space with $\pi_3 X = \mathbb{Z}_3$, $\pi_i X = 0$ for $3 \neq i \geq 1$. This space is represented by a simplicial group, and perturbation techniques are used to compute

$H_7(X,\mathbb{Z}) = \mathbb{Z}_3 \oplus \mathbb{Z}_3$.

Example

```
gap> A:=AbelianGroup([3]);;AbelianInvariants(A);
[ 3 ]
gap>  K:=EilenbergMacLaneSimplicialGroup(A,3,8);
Simplicial group of length 8

gap> C:=ChainComplex(K);
Chain complex of length 8 in characteristic 0 .

gap> Homology(C,7);
[ 3, 3 ]
```

12.3 Eilenberg-MacLane spaces as simplicial free abelian groups (recommended)

For integer $n > 1$ and abelian group A the Eilenberg-MacLane space $K(A,n)$ is better represented as a simplicial free abelian group. (The reason is that the functorial bar resolution of a group can be replaced in computations by the smaller functorial Chevalley-Eilenberg complex of the group when the group is free abelian, obviating the need for perturbation techniques. When A has torision we can

replace it with an inclusion of free abelian groups $A_1 \hookrightarrow A_0$ with $A \cong A_0/A_1$ and again invoke the Chevalley-Eilenberg complex. The current implementation unfortunately handles only free abelian A but the easy extension to non-free A is planned for a future release.)

The following commands compute the integral homology $H_n(K(\mathbb{Z},3),\mathbb{Z})$ for $0 \leq n \leq 16$. (Note that one typically needs fewer than n terms of the Eilenberg-MacLance space to compute its n-th homology -- an error is printed if too few terms of the space are available for a given computation.)

Example

```
gap> A:=AbelianPcpGroup([0]);; #infinite cyclic group
gap> K:=EilenbergMacLaneSimplicialFreeAbelianGroup(A,3,14);
Simplicial free abelian group of length 14

gap> for n in [0..16] do
> Print("Degree ",n," integral homology of K is ",Homology(K,n),"\n");
> od;
Degree 0 integral homology of K is [ 0 ]
Degree 1 integral homology of K is [  ]
Degree 2 integral homology of K is [  ]
Degree 3 integral homology of K is [ 0 ]
Degree 4 integral homology of K is [  ]
Degree 5 integral homology of K is [ 2 ]
Degree 6 integral homology of K is [  ]
Degree 7 integral homology of K is [ 3 ]
Degree 8 integral homology of K is [ 2 ]
Degree 9 integral homology of K is [ 2 ]
Degree 10 integral homology of K is [ 3 ]
Degree 11 integral homology of K is [ 5, 2 ]
Degree 12 integral homology of K is [ 2 ]
Degree 13 integral homology of K is [  ]
Degree 14 integral homology of K is [ 10, 2 ]
Degree 15 integral homology of K is [ 7, 6 ]
Degree 16 integral homology of K is [  ]
```

For an n-connected pointed space X the Freudenthal Suspension Theorem states that the map $X \to \Omega(\Sigma X)$ induces a map $\pi_k(X) \to \pi_k(\Omega(\Sigma X))$ which is an isomorphism for $k \leq 2n$ and epimorphism for $k = 2n+1$. Thus the Eilenberg-MacLane space $K(A,n+1)$ can be constructed from the suspension $\Sigma K(A,n)$ by attaching cells in dimensions $\geq 2n+1$. In particular, there is an isomorphism $H_{k-1}(K(A,n),\mathbb{Z}) \to H_k(K(A,n+1),\mathbb{Z})$ for $k \leq 2n$ and epimorphism for $k = 2n+1$.

For instance, $H_{k-1}(K(\mathbb{Z},3),\mathbb{Z}) \cong H_k(K(\mathbb{Z},4),\mathbb{Z})$ for $k \leq 6$ and $H_6(K(\mathbb{Z},3),\mathbb{Z}) \twoheadrightarrow H_7(K(\mathbb{Z},4),\mathbb{Z})$. This assertion is seen in the following session.

Example

```
gap> A:=AbelianPcpGroup([0]);; #infinite cyclic group
gap> K:=EilenbergMacLaneSimplicialFreeAbelianGroup(A,4,11);
Simplicial free abelian group of length 11

gap> for n in [0..13] do
> Print("Degree ",n," integral homology of K is ",Homology(K,n),"\n");
> od;
Degree 0 integral homology of K is [ 0 ]
Degree 1 integral homology of K is [  ]
Degree 2 integral homology of K is [  ]
```

```
Degree 3 integral homology of K is [  ]
Degree 4 integral homology of K is [ 0 ]
Degree 5 integral homology of K is [  ]
Degree 6 integral homology of K is [ 2 ]
Degree 7 integral homology of K is [  ]
Degree 8 integral homology of K is [ 3, 0 ]
Degree 9 integral homology of K is [  ]
Degree 10 integral homology of K is [ 2, 2 ]
Degree 11 integral homology of K is [  ]
Degree 12 integral homology of K is [ 5, 12, 0 ]
Degree 13 integral homology of K is [ 2 ]
```

12.4 Elementary theoretical information on $H^*(K(\pi,n),\mathbb{Z})$

The cup product is not implemented for the cohomology ring $H^*(K(\pi,n),\mathbb{Z})$. Standard theoretical spectral sequence arguments have to be applied to obtain basic information relating to the ring structure. To illustrate this the following commands compute $H^n(K(\mathbb{Z},2),\mathbb{Z})$ for the first few values of n.

Example

```
gap> K:=EilenbergMacLaneSimplicialFreeAbelianGroup(A,2,10);;
gap> List([0..10],k->Cohomology(K,k));
[ [ 0 ], [  ], [ 0 ], [  ], [ 0 ], [  ], [ 0 ], [  ], [ 0 ], [  ], [ 0 ] ]
```

There is a fibration sequence $K(\pi,n) \hookrightarrow * \twoheadrightarrow K(\pi,n+1)$ in which $*$ denotes a contractible space. For $n=1, \pi=\mathbb{Z}$ the terms of the E_2 page of the Serre integral cohomology spectral sequence for this fibration are

- $E_2^{pq} = H^p(K(\mathbb{Z},2), H^q(K(\mathbb{Z},1),\mathbb{Z}))$.

Since $K(\mathbb{Z},1)$ can be taken to be the circle S^1 we know that it has non-trivial cohomology in degrees 0 and 1 only. The first few terms of the E_2 page are given in the following table.

1	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$
0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$	0	$\mathbb{Z}$
q/p	0	1	2	3	4	5	6	7	8	9	10

Table: E^2 cohomology page for $K(\mathbb{Z},1) \hookrightarrow * \twoheadrightarrow K(\mathbb{Z},2)$

Let x denote the generator of $H^1(K(\mathbb{Z},1),\mathbb{Z})$ and y denote the generator of $H^2(K(\mathbb{Z},2),\mathbb{Z})$. Since $*$ has zero cohomology in degrees ≥ 1 we see that the differential must restrict to an isomorphism $d_2: E_2^{0,1} \to E_2^{2,0}$ with $d_2(x) = y$. Then we see that the differential must restrict to an isomorphism $d_2: E_2^{2,1} \to E_2^{4,0}$ defined on the generator xy of $E_2^{2,1}$ by

$$d_2(xy) = d_2(x)y + (-1)^{\deg(x)} x d_2(y) = y^2 \ .$$

Hence $E_2^{4,0} \cong H^4(K(\mathbb{Z},2),\mathbb{Z})$ is generated by y^2. The argument extends to show that $H^6(K(\mathbb{Z},2),\mathbb{Z})$ is generated by y^3, $H^8(K(\mathbb{Z},2),\mathbb{Z})$ is generated by y^4, and so on.

In fact, to obtain a complete description of the ring $H^*(K(\mathbb{Z},2),\mathbb{Z})$ in this fashion there is no benefit to using computer methods at all. We only need to know the cohomology ring $H^*(K(\mathbb{Z},1),\mathbb{Z}) = H^*(S^1,\mathbb{Z})$ and the single cohomology group $H^2(K(\mathbb{Z},2),\mathbb{Z})$.

A similar approach can be attempted for $H^*(K(\mathbb{Z},3),\mathbb{Z})$ using the fibration sequence $K(\mathbb{Z},2) \hookrightarrow * \twoheadrightarrow K(\mathbb{Z},3)$ and, as explained in Chapter 5 of [Hat01], yields the computation of the group $H^i(K(\mathbb{Z},3),\mathbb{Z})$ for $4 \leq i \leq 13$. The method does not directly yield $H^3(K(\mathbb{Z},3),\mathbb{Z})$ and breaks down in degree 14 yielding only that $H^{14}(K(\mathbb{Z},3),\mathbb{Z}) = 0$ or $\mathbb{Z}_3$. The following commands provide $H^3(K(\mathbb{Z},3),\mathbb{Z}) = \mathbb{Z}$ and $H^{14}(K(\mathbb{Z},3),\mathbb{Z}) = 0$.

Example

```
gap> A:=AbelianPcpGroup([0]);;
gap> K:=EilenbergMacLaneSimplicialFreeAbelianGroup(A,3,15);;
gap> Cohomology(K,3);
[ 0 ]
gap> Cohomology(K,14);
[  ]
```

However, the implementation of these commands is currently a bit naive, and computationally inefficient, since they do not currently employ any homological perturbation techniques.

12.5 The first three non-trivial homotopy groups of spheres

The Hurewicz Theorem immediately gives

$$\pi_n(S^n) \cong \mathbb{Z} \quad (n \geq 1)$$

and

$$\pi_k(S^n) = 0 \quad (k \leq n-1).$$

As a CW-complex the Eilenberg-MacLane space $K = K(\mathbb{Z},n)$ can be obtained from an n-sphere $S^n = e^0 \cup e^n$ by attaching cells in dimensions $\geq n+2$ so as to kill the higher homotopy groups of S^n. From the inclusion $\iota : S^n \hookrightarrow K(\mathbb{Z},n)$ we can form the mapping cone $X = C(\iota)$. The long exact homotopy sequence

$\cdots \to \pi_{k+1}K \to \pi_{k+1}(K,S^n) \to \pi_k S^n \to \pi_k K \to \pi_k(K,S^n) \to \cdots$

implies that $\pi_k(K,S^n) = 0$ for $0 \leq k \leq n+1$ and $\pi_{n+2}(K,S^n) \cong \pi_{n+1}(S^n)$. The relative Hurewicz Theorem gives an isomorphism $\pi_{n+2}(K,S^n) \cong H_{n+2}(K,S^n,\mathbb{Z})$. The long exact homology sequence

$\cdots H_{n+2}(S^n,\mathbb{Z}) \to H_{n+2}(K,\mathbb{Z}) \to H_{n+2}(K,S^n,\mathbb{Z}) \to H_{n+1}(S^n,\mathbb{Z}) \to \cdots$

arising from the cofibration $S^n \hookrightarrow K \twoheadrightarrow X$ implies that $\pi_{n+1}(S^n) \cong \pi_{n+2}(K,S^n) \cong H_{n+2}(K,S^n,\mathbb{Z}) \cong H_{n+2}(K,\mathbb{Z})$. From the GAP computations in 12.3 and the Freudenthal Suspension Theorem we find:

$$\pi_3 S^2 \cong \mathbb{Z}, \qquad \pi_{n+1}(S^n) \cong \mathbb{Z}_2 \quad (n \geq 3).$$

The Hopf fibration $S^3 \to S^2$ has fibre $S^1 = K(\mathbb{Z},1)$. It can be constructed by viewing S^3 as all pairs $(z_1,z_2) \in \mathbb{C}^2$ with $|z_1|^2 + |z_2|^2 = 1$ and viewing S^2 as $\mathbb{C} \cup \infty$; the map sends $(z_1,z_2) \mapsto z_1/z_2$. The homotopy exact sequence of the Hopf fibration yields $\pi_k(S^3) \cong \pi_k(S^2)$ for $k \geq 3$, and in particular

$$\pi_4(S^2) \cong \pi_4(S^3) \cong \mathbb{Z}_2 \,.$$

It will require further techniques (such as the Postnikov tower argument in Section 12.7 below) to establish that $\pi_5(S^3) \cong \mathbb{Z}_2$. Once we have this isomorphism for $\pi_5(S^3)$, the generalized Hopf fibration

$S^3 \hookrightarrow S^7 \twoheadrightarrow S^4$ comes into play. This fibration is contructed as for the classical fibration, but using pairs (z_1, z_2) of quaternions rather than pairs of complex numbers. The Hurewicz Theorem gives $\pi_3(S^7) = 0$; the fibre S^3 is thus homotopic to a point in S^7 and the inclusion of the fibre induces the zero homomorphism $\pi_k(S^3) \xrightarrow{0} \pi_k(S^7)$ $(k \geq 1)$. The exact homotopy sequence of the generalized Hopf fibration then gives $\pi_k(S^4) \cong \pi_k(S^7) \oplus \pi_{k-1}(S^3)$. On taking $k = 6$ we obtain $\pi_6(S^4) \cong \pi_5(S^3) \cong \mathbb{Z}_2$. Freudenthal suspension then gives

$$\pi_{n+2}(S^n) \cong \mathbb{Z}_2, \quad (n \geq 2).$$

12.6 The first two non-trivial homotopy groups of the suspension and double suspension of a $K(G,1)$

For any group G we consider the homotopy groups $\pi_n(\Sigma K(G,1))$ of the suspension $\Sigma K(G,1)$ of the Eilenberg-MacLance space $K(G,1)$. On taking $G = \mathbb{Z}$, and observing that $S^2 = \Sigma K(\mathbb{Z},1)$, we specialize to the homotopy groups of the 2-sphere S^2.

By construction,

$$\pi_1(\Sigma K(G,1)) = 0\ .$$

The Hurewicz Theorem gives

$$\pi_2(\Sigma K(G,1)) \cong G_{ab}$$

via the isomorphisms $\pi_2(\Sigma K(G,1)) \cong H_2(\Sigma K(G,1),\mathbb{Z}) \cong H_1(K(G,1),\mathbb{Z}) \cong G_{ab}$. R. Brown and J.-L. Loday [BL87] obtained the formulae

$$\pi_3(\Sigma K(G,1)) \cong \ker(G \otimes G \to G, x \otimes y \mapsto [x,y])\ ,$$

$$\pi_4(\Sigma^2 K(G,1)) \cong \ker(G \widetilde{\otimes} G \to G, x \widetilde{\otimes} y \mapsto [x,y])$$

involving the nonabelian tensor square and nonabelian symmetric square of the group G. The following commands use the nonabelian tensor and symmetric product to compute the third and fourth homotopy groups for $G = Syl_2(M_{12})$ the Sylow 2-subgroup of the Mathieu group M_{12}.

Example

```
gap> G:=SylowSubgroup(MathieuGroup(12),2);;
gap> ThirdHomotopyGroupOfSuspensionB(G);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
gap>
gap> FourthHomotopyGroupOfDoubleSuspensionB(G);
[ 2, 2, 2, 2, 2, 2 ]
```

12.7 Postnikov towers and $\pi_5(S^3)$

A Postnikov system for the sphere S^3 consists of a sequence of fibrations $\cdots X_3 \xrightarrow{p_3} X_2 \xrightarrow{p_2} X_1 \xrightarrow{p_1} *$ and a sequence of maps $\phi_n : S^3 \to X_n$ such that

- $p_n \circ \phi_n = \phi_{n-1}$

- The map $\phi_n: S^3 \to X_n$ induces an isomorphism $\pi_k(S^3) \to \pi_k(X_n)$ for all $k \leq n$
- $\pi_k(X_n) = 0$ for $k > n$
- and consequently each fibration p_n has fibre an Eilenberg-MacLane space $K(\pi_n(S^3), n)$.

The space X_n is obtained from S^3 by adding cells in dimensions $\geq n+2$ and thus

- $H_k(X_n, \mathbb{Z}) = H_k(S^3, \mathbb{Z})$ for $k \leq n+1$.

So in particular $X_1 = X_2 = *, X_3 = K(\mathbb{Z}, 3)$ and we have a fibration sequence $K(\pi_4(S^3), 4) \hookrightarrow X_4 \twoheadrightarrow K(\mathbb{Z}, 3)$. The terms in the E_2 page of the Serre integral cohomology spectral sequence of this fibration are

- $E_2^{p,q} = H^p(K(\mathbb{Z}, 3), H_q(K(\mathbb{Z}_2, 4), \mathbb{Z}))$.

The first few terms in the E_2 page can be computed using the commands of Sections 12.2 and 12.3 and recorded as follows.

8	$\mathbb{Z}_2$	0	0							
7	$\mathbb{Z}_2$	0	0							
6	0	0	0							
5	$\pi_4(S^3)$	0	0	$\pi_4(S^3)$	0	0	0			
4	0	0	0	0	0	0				
3	0	0	0	0	0	0				
2	0	0	0	0	0	0	0	0		
1	0	0	0	0	0	0	0	0		
0	$\mathbb{Z}$	0	0	$\mathbb{Z}$	0	0	$\mathbb{Z}_2$	0	$\mathbb{Z}_3$	$\mathbb{Z}_2$
q/p	0	1	2	3	4	5	6	7	8	9

Table: E_2 cohomology page for $K(\pi_4(S^3), 4) \hookrightarrow X_4 \twoheadrightarrow X_3$

Since we know that $H^5(X_4, \mathbb{Z}) = 0$, the differentials in the spectral sequence must restrict to an isomorphism $E_2^{0,5} = \pi_4(S^3) \xrightarrow{\cong} E_2^{6,0} = \mathbb{Z}_2$. This provides an alternative derivation of $\pi_4(S^3) \cong \mathbb{Z}_2$. We can also immediately deduce that $H^6(X_4, \mathbb{Z}) = 0$. Let x be the generator of $E_2^{0,5}$ and y the generator of $E_2^{3,0}$. Then the generator xy of $E_2^{3,5}$ gets mapped to a non-zero element $d_7(xy) = d_7(x)y - xd_7(y)$. Hence the term $E_2^{0,7} = \mathbb{Z}_2$ must get mapped to zero in $E_2^{3,5}$. It follows that $H^7(X_4, \mathbb{Z}) = \mathbb{Z}_2$.

The integral cohomology of Eilenberg-MacLane spaces yields the following information on the E_2 page $E_2^{p,q} = H_p(X_4, H^q(K(\pi_5 S^3, 5), \mathbb{Z}))$ for the fibration $K(\pi_5(S^3), 5) \hookrightarrow X_5 \twoheadrightarrow X_4$.

6	$\pi_5(S^3)$	0	0	$\pi_5(S^3)$	0	0		
5	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	
0	$\mathbb{Z}$	0	0	$\mathbb{Z}$	0	0	0	$H^7(X_4, \mathbb{Z})$
q/p	0	1	2	3	4	5	6	7

Table: E_2 cohomology page for $K(\pi_5(S^3), 5) \hookrightarrow X_5 \twoheadrightarrow X_4$

Since we know that $H^6(X_5,\mathbb{Z})=0$, the differentials in the spectral sequence must restrict to an isomorphism $E_2^{0,6}=\pi_5(S^3)\xrightarrow{\cong}E_2^{7,0}=H^7(X_4,\mathbb{Z})$. We can conclude the desired result:

$$\pi_5(S^3)=\mathbb{Z}_2\ .$$

Note that the fibration $X_4\twoheadrightarrow K(\mathbb{Z},3)$ is determined by a cohomology class $\kappa\in H^5(K(\mathbb{Z},3),\mathbb{Z}_2)=\mathbb{Z}_2$. If $\kappa=0$ then we'd have $X_4=K(\mathbb{Z}_2,4)\times K(\mathbb{Z},3)$ and, as the following commands show, we'd then have $H_4(X_4,\mathbb{Z})=\mathbb{Z}_2$.

Example

```
gap> K:=EilenbergMacLaneSimplicialGroup(AbelianPcpGroup([0]),3,7);;
gap> L:=EilenbergMacLaneSimplicialGroup(CyclicGroup(2),4,7);;
gap> CK:=ChainComplex(K);;
gap> CL:=ChainComplex(L);;
gap> T:=TensorProduct(CK,CL);;
gap> Homology(T,4);
[ 2 ]
```

Since we know that $H_4(X_4,\mathbb{Z})=0$ we can conclude that the Postnikov invariant κ is the non-zero class in $H^5(K(\mathbb{Z},3),\mathbb{Z}_2)=\mathbb{Z}_2$.

12.8 Towards $\pi_4(\Sigma K(G,1))$

Consider the suspension $X=\Sigma K(G,1)$ of a classifying space of a group G once again. This space has a Postnikov system in which $X_1=*$, $X_2=K(G_{ab},2)$. We have a fibration sequence $K(\pi_3X,3)\hookrightarrow X_3\twoheadrightarrow K(G_{ab},2)$. The corresponding integral cohomology Serre spectral sequence has E_2 page with terms

- $E_2^{p,q}=H^p(K(G_{ab},2),H^q(K(\pi_3X,3)),\mathbb{Z}))$.

As an example, for the Alternating group $G=A_4$ of order 12 the following commands of Section 12.6 compute $G_{ab}=\mathbb{Z}_3$ and $\pi_3X=\mathbb{Z}_6$.

Example

```
gap> AbelianInvariants(G);
[ 3 ]
gap> ThirdHomotopyGroupOfSuspensionB(G);
[ 2, 3 ]
```

The first terms of the E_2 page can be calculated using the commands of Sections 12.2 and 12.3.

7	$\mathbb{Z}_2$	0					
6	$\mathbb{Z}_2$	0	0	0			
5	0	0	0	0			
4	$\mathbb{Z}_6$	0	0	$\mathbb{Z}_3$			
3	0	0	0	0	0	0	
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	$\mathbb{Z}$	0	0	$\mathbb{Z}_3$	0	$\mathbb{Z}_3$	0	$\mathbb{Z}_9$
q/p	0	1	2	3	4	5	6	7

Table: E^2 cohomology page for $K(\pi_3 X,3) \hookrightarrow X_3 \twoheadrightarrow X_2$

We know that $H^1(X_3,\mathbb{Z}) = 0$, $H^2(X_3,\mathbb{Z}) = H^1(G,\mathbb{Z}) = 0$, $H^3(X_3,\mathbb{Z}) = H^2(G,\mathbb{Z}) = \mathbb{Z}_3$, and that $H^4(X_3,\mathbb{Z})$ is a subgroup of $H^3(G,\mathbb{Z}) = \mathbb{Z}_2$. It follows that the differential induces a surjection $E_2^{0,4} = \mathbb{Z}_6 \twoheadrightarrow E_2^{5,0} = \mathbb{Z}_3$. Consequently $H^4(X_3,\mathbb{Z}) = \mathbb{Z}_2$ and $H^5(X_3,\mathbb{Z}) = 0$ and $H^6(X_3,\mathbb{Z}) = \mathbb{Z}_2$.

The E_2 page for the fibration $K(\pi_4 X,4) \hookrightarrow X_4 \twoheadrightarrow X_3$ contains the following terms.

5	$\pi_4 X$	0	0				
4	0	0	0	0			
3	0	0	0	0	0	0	
2	0	0	0	0	0	0	
1	0	0	0	0	0	0	0
0	$\mathbb{Z}$	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_2$
q/p	0	1	2	3	4	5	6

Table: E^2 cohomology page for $K(\pi_4 X,4) \hookrightarrow X_4 \twoheadrightarrow X_3$

We know that $H^5(X_4,\mathbb{Z})$ is a subgroup of $H^4(G,\mathbb{Z}) = \mathbb{Z}_6$, and hence that there is a homomorphisms $\pi_4 X \to \mathbb{Z}_2$ whose kernel is a subgroup of $\mathbb{Z}_6$. If follows that $|\pi_4 X| \leq 12$.

12.9 Enumerating homotopy 2-types

A *2-type* is a CW-complex X whose homotopy groups are trivial in dimensions $n = 0$ and $n > 2$. As explained in 6.11 the homotopy type of such a space can be captured algebraically by a cat^1-group G. Let X, Y be 2-tytpes represented by cat^1-groups G, H. If X and Y are homotopy equivalent then there exists a sequence of morphisms of cat^1-groups

$$G \to K_1 \to K_2 \leftarrow K_3 \to \cdots \to K_n \leftarrow H$$

in which each morphism induces isomorphisms of homotopy groups. When such a sequence exists we say that G is *quasi-isomorphic* to H. We have the following result.

THEOREM. The 2-types X and Y are homotopy equivalent if and only if the associated cat^1-groups G and H are quasi-isomorphic.

The following commands produce a list L of all of the 62 non-isomorphic cat^1-groups whose underlying group has order 16.

Example

```
gap> L:=[];;
gap> for G in AllSmallGroups(16) do
> Append(L,CatOneGroupsByGroup(G));
> od;
gap> Length(L);
62
```

The next commands use the first and second homotopy groups to prove that the list L contains at least 37 distinct quasi-isomorphism types.

Example

```
gap> Invariants:=function(G)
> local inv;
> inv:=[];
```

```
> inv[1]:=IdGroup(HomotopyGroup(G,1));
> inv[2]:=IdGroup(HomotopyGroup(G,2));
> return inv;
> end;;

gap> C:=Classify(L,Invariants);;
gap> Length(C);
```

The following additional commands use second and third integral homology in conjunction with the first two homotopy groups to prove that the list L contains AT LEAST 49 distinct quasi-isomorphism types.

Example

```
gap> Invariants2:=function(G)
> local inv;
> inv:=[];
> inv[1]:=Homology(G,2);
> inv[2]:=Homology(G,3);
> return inv;
> end;;
gap> C:=RefineClassification(C,Invariants2);;

gap> Length(C);
49
```

The following commands show that the above list L contains AT MOST 51 distinct quasi-isomorphism types.

Example

```
gap> Q:=List(L,QuasiIsomorph);;
gap> M:=[];;

gap> for q in Q do
> bool:=true;;
> for m in M do
> if not IsomorphismCatOneGroups(m,q)=fail then bool:=false; break; fi;
> od;
> if bool then Add(M,q); fi;
> od;

gap> Length(M);
51
```

12.10 Identifying cat^1-groups of low order

Let us define the *order* of a cat^1-group to be the order of its underlying group. The function `IdQuasiCatOneGroup(C)` inputs a cat^1-group C of "low order" and returns an integer pair $[n,k]$ that uniquely idenifies the quasi-isomorphism type of C. The integer n is the order of a smallest cat^1-group quasi-isomorphic to C. The integer k identifies a particular cat^1-group of order n.

The following commands use this function to show that there are precisely 49 distinct quasi-isomorphism types of cat^1-groups of order 16.

Example

```
gap> L:=[];;
gap> for G in AllSmallGroups(16) do
> Append(L,CatOneGroupsByGroup(G));
> od;
gap> M:=List(L,IdQuasiCatOneGroup);
[ [ 16, 1 ], [ 16, 2 ], [ 16, 3 ], [ 16, 4 ], [ 16, 5 ], [ 4, 4 ], [ 1, 1 ],
  [ 16, 6 ], [ 16, 7 ], [ 16, 8 ], [ 16, 9 ], [ 16, 10 ], [ 16, 11 ],
  [ 16, 9 ], [ 16, 12 ], [ 16, 13 ], [ 16, 14 ], [ 16, 15 ], [ 4, 1 ],
  [ 4, 2 ], [ 16, 16 ], [ 16, 17 ], [ 16, 18 ], [ 16, 19 ], [ 16, 20 ],
  [ 16, 21 ], [ 16, 22 ], [ 16, 23 ], [ 16, 24 ], [ 16, 25 ], [ 16, 26 ],
  [ 16, 27 ], [ 16, 28 ], [ 4, 3 ], [ 4, 1 ], [ 4, 4 ], [ 4, 4 ], [ 4, 2 ],
  [ 4, 5 ], [ 16, 29 ], [ 16, 30 ], [ 16, 31 ], [ 16, 32 ], [ 16, 33 ],
  [ 16, 34 ], [ 4, 3 ], [ 4, 4 ], [ 4, 4 ], [ 16, 35 ], [ 16, 36 ], [ 4, 3 ],
  [ 16, 37 ], [ 16, 38 ], [ 16, 39 ], [ 16, 40 ], [ 16, 41 ], [ 16, 42 ],
  [ 16, 43 ], [ 4, 3 ], [ 4, 4 ], [ 1, 1 ], [ 4, 5 ] ]
gap> Length(SSortedList(M));
49
```

The next example first identifies the order and the identity number of the cat^1-group C corresponding to the crossed module (see 12.1)

$$\iota: G \longrightarrow Aut(G), g \mapsto (x \mapsto gxg^{-1})$$

for the dihedral group G of order 10. It then realizes a smallest possible cat^1-group D of this quasi-isomorphism type.

Example

```
gap> C:=AutomorphismGroupAsCatOneGroup(DihedralGroup(10));
Cat-1-group with underlying group Group( [ f1, f2, f3, f4, f5 ] ) .

gap> Order(C);
200
gap> IdCatOneGroup(C);
[ 200, 42, 4 ]
gap>
gap> IdQuasiCatOneGroup(C);
[ 2, 1 ]
gap> D:=SmallCatOneGroup(2,1);
Cat-1-group with underlying group Group( [ f1 ] ) .
```

12.11 Identifying crossed modules of low order

The following commands construct the crossed module $\partial: G \otimes G \to G$ involving the nonabelian tensor square of the dihedral group G of order 10, identify it as being number 71 in the list of crossed modules of order 100, create a quasi-isomorphic crossed module of order 4, and finally construct the corresponding cat^1-group of order 100.

Example

```
gap> G:=DihedralGroup(10);;
gap> T:=NonabelianTensorSquareAsCrossedModule(G);
Crossed module with group homomorphism GroupHomomorphismByImages( Group(
[ f3*f1*f3^-1*f1^-1, f3*f2*f3^-1*f2^-1 ] ), Group( [ f1, f2 ] ),
[ f3*f1*f3^-1*f1^-1, f3*f2*f3^-1*f2^-1 ], [ <identity> of ..., f2^3 ] )

gap> IdCrossedModule(T);
[ 100, 71 ]
gap> Q:=QuasiIsomorph(T);
Crossed module with group homomorphism Pcgs([ f2 ]) -> [ <identity> of ... ]

gap> Order(Q);
4
gap> C:=CatOneGroupByCrossedModule(T);
Cat-1-group with underlying group Group( [ F1, F2, F1 ] ) .
```

Chapter 13

Congruence Subgroups, Cuspidal Cohomology and Hecke Operators

In this chapter we explain how HAP can be used to make computions about modular forms associated to congruence subgroups Γ of $SL_2(\mathbb{Z})$. Also, in Subsection 10.8 onwards, we demonstrate cohomology computations for the *Picard group* $SL_2(\mathbb{Z}[i])$, some *Bianchi groups* $PSL_2(\mathscr{O}_{-d})$ where $\mathscr{O}_d$ is the ring of integers of $\mathbb{Q}(\sqrt{-d})$ for square free positive integer d, and some other groups of the form $SL_m(\mathscr{O})$, $GL_m(\mathscr{O})$, $PSL_m(\mathscr{O})$, $PGL_m(\mathscr{O})$, for $m = 2,3,4$ and certain $\mathscr{O} = \mathbb{Z}, \mathscr{O}_{-d}$.

13.1 Eichler-Shimura isomorphism

We begin by recalling the Eichler-Shimura isomorphism [Eic57][Shi59]

$$S_k(\Gamma) \oplus \overline{S_k(\Gamma)} \oplus E_k(\Gamma) \cong_{\mathsf{Hecke}} H^1(\Gamma, P_{\mathbb{C}}(k-2))$$

which relates the cohomology of groups to the theory of modular forms associated to a finite index subgroup Γ of $SL_2(\mathbb{Z})$. In subsequent sections we explain how to compute with the right-hand side of the isomorphism. But first, for completeness, let us define the terms on the left-hand side.

Let N be a positive integer. A subgroup Γ of $SL_2(\mathbb{Z})$ is said to be a *congruence subgroup* of level N if it contains the kernel of the canonical homomorphism $\pi_N : SL_2(\mathbb{Z}) \to SL_2(\mathbb{Z}/N\mathbb{Z})$. So any congruence subgroup is of finite index in $SL_2(\mathbb{Z})$, but the converse is not true.

One congruence subgroup of particular interest is the group $\Gamma(N) = \ker(\pi_N)$, known as the *principal congruence subgroup* of level N. Another congruence subgroup of particular interest is the group $\Gamma_0(N)$ of those matrices that project to upper triangular matrices in $SL_2(\mathbb{Z}/N\mathbb{Z})$.

A *modular form* of weight k for a congruence subgroup Γ is a complex valued function on the upper-half plane, $f : \mathfrak{h} = \{z \in \mathbb{C} : Im(z) > 0\} \to \mathbb{C}$, satisfying:

- $f(\dfrac{az+b}{cz+d}) = (cz+d)^k f(z)$ for $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$,
- f is 'holomorphic' on the *extended upper-half plane* $\mathfrak{h}^* = \mathfrak{h} \cup \mathbb{Q} \cup \{\infty\}$ obtained from the upper-half plane by 'adjoining a point at each cusp'.

The collection of all weight k modular forms for Γ form a vector space $M_k(\Gamma)$ over $\mathbb{C}$.

A modular form f is said to be a *cusp form* if $f(\infty)=0$. The collection of all weight k cusp forms for Γ form a vector subspace $S_k(\Gamma)$. There is a decomposition

$$M_k(\Gamma) \cong S_k(\Gamma) \oplus E_k(\Gamma)$$

involving a summand $E_k(\Gamma)$ known as the *Eisenstein space*. See [Ste07] for further introductory details on modular forms.

The Eichler-Shimura isomorphism is more than an isomorphism of vector spaces. It is an isomorphism of Hecke modules: both sides admit notions of *Hecke operators*, and the isomorphism preserves these operators. The bar on the left-hand side of the isomorphism denotes complex conjugation, or *anti-holomorphic* forms. See [Wie78] for a full account of the isomorphism.

On the right-hand side of the isomorphism, the $\mathbb{Z}\Gamma$-module $P_{\mathbb{C}}(k-2) \subset \mathbb{C}[x,y]$ denotes the space of homogeneous degree $k-2$ polynomials with action of Γ given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot p(x,y) = p(dx-by, -cx+ay) \ .$$

In particular $P_{\mathbb{C}}(0) = \mathbb{C}$ is the trivial module. Below we shall compute with the integral analogue $P_{\mathbb{Z}}(k-2) \subset \mathbb{Z}[x,y]$.

In the following sections we explain how to use the right-hand side of the Eichler-Shimura isomorphism to compute eigenvalues of the Hecke operators restricted to the subspace $S_k(\Gamma)$ of cusp forms.

13.2 Generators for $SL_2(\mathbb{Z})$ and the cubic tree

The matrices $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ generate $SL_2(\mathbb{Z})$ and it is not difficult to devise an algorithm for expressing an arbitrary integer matrix A of determinant 1 as a word in S, T and their inverses. The following illustrates such an algorithm.

Example

```
gap> A:=[[4,9],[7,16]];;
gap> word:=AsWordInSL2Z(A);
[ [ [ 1, 0 ], [ 0, 1 ] ], [ [ 0, 1 ], [ -1, 0 ] ], [ [ 1, -1 ], [ 0, 1 ] ],
  [ [ 0, 1 ], [ -1, 0 ] ], [ [ 1, 1 ], [ 0, 1 ] ], [ [ 0, 1 ], [ -1, 0 ] ],
  [ [ 1, -1 ], [ 0, 1 ] ], [ [ 1, -1 ], [ 0, 1 ] ], [ [ 1, -1 ], [ 0, 1 ] ],
  [ [ 0, 1 ], [ -1, 0 ] ], [ [ 1, 1 ], [ 0, 1 ] ], [ [ 1, 1 ], [ 0, 1 ] ] ]
gap> Product(word);
[ [ 4, 9 ], [ 7, 16 ] ]
```

It is convenient to introduce the matrix $U = ST = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}$. The matrices S and U also generate $SL_2(\mathbb{Z})$. In fact we have a free presentation $SL_2(\mathbb{Z}) = \langle S,U \,|\, S^4 = U^6 = 1, S^2 = U^3 \rangle$.

The *cubic tree* $\mathscr{T}$ is a tree (*i.e.* a 1-dimensional contractible regular CW-complex) with countably infinitely many edges in which each vertex has degree 3. We can realize the cubic tree $\mathscr{T}$ by taking the left cosets of $\mathscr{U} = \langle U \rangle$ in $SL_2(\mathbb{Z})$ as vertices, and joining cosets $x\mathscr{U}$ and $y\mathscr{U}$ by an edge if, and only if, $x^{-1}y \in \mathscr{U}S\mathscr{U}$. Thus the vertex $\mathscr{U}$ is joined to $S\mathscr{U}$, $US\mathscr{U}$ and $U^2S\mathscr{U}$. The vertices of this tree are in one-to-one correspondence with all reduced words in S, U and U^2 that, apart from the identity, end in S.

From our realization of the cubic tree $\mathscr{T}$ we see that $SL_2(\mathbb{Z})$ acts on $\mathscr{T}$ in such a way that each vertex is stabilized by a cyclic subgroup conjugate to $\mathscr{U} = \langle U \rangle$ and each edge is stabilized by a cyclic subgroup conjugate to $\mathscr{S} = \langle S \rangle$.

In order to store this action of $SL_2(\mathbb{Z})$ on the cubic tree $\mathscr{T}$ we just need to record the following finite amount of information.

13.3 One-dimensional fundamental domains and generators for congruence subgroups

It is useful to invoke the following general result which follows from a perturbation result about free $\mathbb{Z}G$-resolutons in [EHS06, Theorem 2] and an old observation of John Milnor that a free $\mathbb{Z}G$-resolution can be realized as the cellular chain complex of a CW-complex if it can be so realized in low dimensions.

THEOREM. Let X be a contractible CW-complex on which a group G acts by permuting cells. The cellular chain complex C_*X is a $\mathbb{Z}G$-resolution of $\mathbb{Z}$ which typically is not free. Let $[e^n]$ denote the orbit of the n-cell e^n under the action. Let $G^{e^n} \leq G$ denote the stabilizer subgroup of e^n, in which group elements are not required to stabilize e^n point-wise. Let Y_{e^n} denote a contractible CW-complex on which G^{e^n} acts cellularly and freely. Then there exists a contractible CW-complex W on which G acts cellularly and freely, and in which the orbits of n-cells are labelled by $[e^p] \otimes [f^q]$ where $p+q=n$ and $[e^p]$ ranges over the G-orbits of p-cells in X, $[f^q]$ ranges over the G^{e^p}-orbits of q-cells in Y_{e^p}.

Let W be as in the theorem. Then the quotient CW-complex $B_G = W/G$ is a classifying space for G. Let T denote a maximal tree in the 1-skeleton B_G^1. Basic geometric group theory tells us that the 1-cells in $B_G^1 \setminus T$ correspond to a generating set for G.

Suppose we wish to compute a set of generators for a principal congruence subgroup $\Gamma = \Gamma(N)$,$N > 2$. In the above theorem take $X = \mathscr{T}$ to be the cubic tree, and note that Γ acts freely on $\mathscr{T}$ and thus that $W = \mathscr{T}$. To determine the 1-cells of $B_\Gamma \setminus T$ we need to determine a cellular subspace $D_\Gamma \subset \mathscr{T}$ whose images under the action of Γ cover $\mathscr{T}$ and are pairwise either disjoint or identical. The subspace D_Γ will not be a CW-complex as it won't be closed, but it can be chosen to be connected, and hence contractible. We call D_Γ a *fundamental region* for Γ. We denote by $\mathring{D}_\Gamma$ the largest CW-subcomplex of D_Γ. The vertices of $\mathring{D}_\Gamma$ are the same as the vertices of D_Γ. Thus $\mathring{D}_\Gamma$ is a subtree of the cubic tree with $|SL_2(\mathbb{Z}) : \Gamma|/6$ vertices. For each vertex v in the tree $\mathring{D}_\Gamma$ define $\eta(v) = 3 - \text{degree}(v)$. Then the number of generators for Γ will be $(1/2)\sum_{v \in \mathring{D}_\Gamma} \eta(v)$.

The following commands determine generators for $\Gamma(6)$ and display $\mathring{D}_{\Gamma(6)}$. The subgroup $\Gamma(6)$ is free on 13 generators.

Example

```
gap> G:=HAP_PrincipalCongruenceSubgroup(6);;
gap> HAP_SL2TreeDisplay(G);


gap> gens:=GeneratorsOfGroup(G);
[ [ [ -83, -18 ], [ 60, 13 ] ], [ [ -77, -18 ], [ 30, 7 ] ],
  [ [ -65, -12 ], [ 168, 31 ] ], [ [ -53, -12 ], [ 84, 19 ] ],
  [ [ -47, -18 ], [ 222, 85 ] ], [ [ -41, -12 ], [ 24, 7 ] ],
  [ [ -35, -6 ], [ 6, 1 ] ], [ [ -11, -18 ], [ 30, 49 ] ],
  [ [ -11, -6 ], [ 24, 13 ] ], [ [ -5, -18 ], [ 12, 43 ] ],
  [ [ -5, -12 ], [ 18, 43 ] ], [ [ -5, -6 ], [ 6, 7 ] ],
```

```
    [ [ 1, 0 ], [ -6, 1 ] ] ]
```

A related approach to the problem of computing *independent generators* (= no nontrivial identities between them) of congruence subgroups of $SL_2(\mathbb{Z})$ is described in [Kul91]. The principal congruence subgroup $\Gamma(N)$ is free (since it acts on a tree with trivial stabilizers) and thus any minimal generating set is independent. Such a generating set is returned by the above method.

The congruence subgroup $\Gamma_0(N)$ does not act freely on the vertices of $\mathscr{T}$, and so one needs to incorporate a generator for the cyclic stabilizer group according to the above theorem. Alternatively, we can replace the cubic tree by a six-fold cover $\mathscr{T}'$ on whose vertex set $\Gamma_0(N)$ acts freely. This alternative approach will produce a redundant set of generators. The following commands display $\mathring{D}_{\Gamma_0(39)}$ for a fundamental region in $\mathscr{T}'$. They also use the corresponding generating set for $\Gamma_0(39)$, involving 18 generators, to compute the abelianization $\Gamma_0(39)^{ab} = \mathbb{Z}_2 \oplus \mathbb{Z}_3^2 \oplus \mathbb{Z}^9$. The abelianization shows that any generating set has at least 11 generators.

Example

```
gap> G:=HAP_CongruenceSubgroupGamma0(39);;
gap> HAP_SL2TreeDisplay(G);
gap> Length(GeneratorsOfGroup(G));
18
gap> AbelianInvariants(G);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 3, 3 ]
```

Note that to compute D_Γ one only needs to be able to test whether a given matrix lies in Γ or not. Given an inclusion $\Gamma' \subset \Gamma$ of congruence subgroups, it is straightforward to use the trees $\mathring{D}_{\Gamma'}$ and $\mathring{D}_\Gamma$ to compute a system of coset representative for $\Gamma' \setminus \Gamma$.

13.3.1 Ambient transversals

For any congruence subgroup $\Gamma \leq SL_2(\mathbb{Z})$ a fundamental domain for the free action of Γ on some tree can always be computed by brute force search, but for principal subgroups and for $\Gamma_0(N)$ one can use theoretical descriptions of the coset space $SL_2(\mathbb{Z})/\Gamma$ to vastly improve efficiency. However the fundamental domain is constructed, it provides and algorithm for producing a transversal of Γ in $SL_2(\mathbb{Z})$. This transversal is referred to as the *ambient transversal* in the HAP implementation of congruence subgroups.

13.4 Independent generators

Once one has access to an ambient transversal for a congruence subgroup $\Gamma \leq SL_2(\mathbb{Z})$ one can construct the finite regular CW-complex $Y = K/\Gamma$ where K is the barycentric subdivision of the cubic tree $K = sd\mathscr{T}$. The complex Y will be a graph. In order to visualize Y it is better to let K be the double barycenctric subdivision K. Every homology cycle in Y gives rise to a generator of Γ of infinite order. Every vertex of degree 1 in Y gives rise to a generator of finite order. These infinite and finite generators constitute an independent generating set for Γ (i.e. there are no non-trivial relators).

For example, the following commands establish that $\Gamma(5)$ has an independent generating set consisting of 11 generators of infinite order. They then establish that $\Gamma_0(13)$ has an independent generating set with one generator of infinite order and four generators of finite order.

Example
```
gap> gamma:=HAP_PrincipalCongruenceSubgroup(5);;
gap> K:=ContractibleGcomplex("SL(2,Z)");;
gap> K:=BarycentricSubdivision(K);;K:=BarycentricSubdivision(K);;
gap> Y:=GComplexToRegularCWComplex(K,gamma);;
gap> Homology(Y,1);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Display( GraphOfRegularCWComplex( Y ));
```

Example
```
gap> gamma:=HAP_CongruenceSubgroupGamma0(13);;
gap> Y:=GComplexToRegularCWComplex(K,gamma);;
gap> Homology(Y,1);
[ 0 ]
gap> Display( GraphOfRegularCWComplex( Y ));
```

Explicit generators can be extracted from the CW-complex Y but this has not yet been implemented as a function in HAP. However, the Congruence package for GAP [DJKV24] has implemented Kulkrani's algorithm for independent generators. We find:

Example
```
gap> gamma:=PrincipalCongruenceSubgroup(5);; #using Congruence package
gap> gens:=GeneratorsOfGroup(gamma);;
#I  Using the Congruence package for GeneratorsOfGroup ...
gap> List(gens,Order);
[ infinity, infinity, infinity, infinity, infinity, infinity, infinity,
  infinity, infinity, infinity, infinity ]

gap> gamma:=CongruenceSubgroupGamma0(13);; #using Congruence package
gap> gens:=GeneratorsOfGroup(gamma);;
#I  Using the Congruence package for GeneratorsOfGroup ...
gap> List(gens,Order);
[ infinity, 3, 4, 4, 3 ]
```

13.5 Genus

A congruence subgroup $\Gamma \leq SL_2(\mathbb{Z})$ acts on the upper half plane $\mathfrak{h}$ (see 13.7). The quotient $\mathfrak{h}/\Gamma$ is a non-compact surface. Compactifying, using the Borel-Serre compactification, yields a closed surface with boundary. Filling the boundary curves with disks then yields a closed surface without boundary. The genus of this surface is by definition the *genus* of Γ.

The following computes the genus of $\Gamma(5)$ to be 0 and also the genus of $\Gamma_0(13)$ to be 0.

Example
```
gap> K:=ContractibleGcomplex("BorelSerreSL2Z");;
gap> K:=BarycentricSubdivision(K);;
gap> dK:=ContractibleGcomplex("BorelSerreBoundarySL2Z");;
gap> dK:=BarycentricSubdivision(dK);;

gap> gamma:=HAP_PrincipalCongruenceSubgroup(5);;
```

```
gap> Y:=GComplexToRegularCWComplex(K,gamma);;
gap> dY:=GComplexToRegularCWComplex(dK,gamma);;
gap> EulerCharacteristic(Y) + Length(Homology(dY,1)); # =Euler characteristic of surface
2

gap> gamma:=HAP_CongruenceSubgroupGamma0(13);;
gap> Y:=GComplexToRegularCWComplex(K,gamma);;
gap> dY:=GComplexToRegularCWComplex(dK,gamma);;
gap> EulerCharacteristic(Y) + Length(Homology(dY,1)); # =Euler characteristic of surface
2

gap> genus:=0;; EulerCharacteristic(ClosedSurface(genus));
2
```

13.6 Cohomology of congruence subgroups

To compute the cohomology $H^n(\Gamma, A)$ of a congruence subgroup Γ with coefficients in a $\mathbb{Z}\Gamma$-module A we need to construct $n+1$ terms of a free $\mathbb{Z}\Gamma$-resolution of $\mathbb{Z}$. We can do this by first using perturbation techniques (as described in [BE14]) to combine the cubic tree with resolutions for the cyclic groups of order 4 and 6 in order to produce a free $\mathbb{Z}G$-resolution R_* for $G = SL_2(\mathbb{Z})$. This resolution is also a free $\mathbb{Z}\Gamma$-resolution with each term of rank

$$\text{rank}_{\mathbb{Z}\Gamma} R_k = |G:\Gamma| \times \text{rank}_{\mathbb{Z}G} R_k \ .$$

For congruence subgroups of lowish index in G this resolution suffices to make computations. The following commands compute

$$H^1(\Gamma_0(39), \mathbb{Z}) = \mathbb{Z}^9 \ .$$

Example

```
gap> R:=ResolutionSL2Z_alt(2);
Resolution of length 2 in characteristic 0 for SL(2,Integers) .

gap> gamma:=HAP_CongruenceSubgroupGamma0(39);;
gap> S:=ResolutionFiniteSubgroup(R,gamma);
Resolution of length 2 in characteristic 0 for
CongruenceSubgroupGamma0( 39)  .

gap> Cohomology(HomToIntegers(S),1);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

This computation establishes that the space $M_2(\Gamma_0(39))$ of weight 2 modular forms is of dimension 9.

The following commands show that $\text{rank}_{\mathbb{Z}\Gamma_0(39)} R_1 = 112$ but that it is possible to apply 'Tietze like' simplifications to R_* to obtain a free $\mathbb{Z}\Gamma_0(39)$-resolution T_* with $\text{rank}_{\mathbb{Z}\Gamma_0(39)} T_1 = 11$. It is more efficient to work with T_* when making cohomology computations with coefficients in a module A of large rank.

Example

```
gap> S!.dimension(1);
112
gap> T:=TietzeReducedResolution(S);
Resolution of length 2 in characteristic 0 for CongruenceSubgroupGamma0(
39)  .

gap> T!.dimension(1);
11
```

The following commands compute

$$H^1(\Gamma_0(39), P_{\mathbb{Z}}(8)) = \mathbb{Z}_3 \oplus \mathbb{Z}_6 \oplus \mathbb{Z}_{168} \oplus \mathbb{Z}^{84},$$

$$H^1(\Gamma_0(39), P_{\mathbb{Z}}(9)) = \mathbb{Z}_2 \oplus \mathbb{Z}_2.$$

Example

```
gap> P:=HomogeneousPolynomials(gamma,8);;
gap> c:=Cohomology(HomToIntegralModule(T,P),1);
[ 3, 6, 168, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Length(c);
87

gap> P:=HomogeneousPolynomials(gamma,9);;
gap> c:=Cohomology(HomToIntegralModule(T,P),1);
[ 2, 2 ]
```

This computation establishes that the space $M_{10}(\Gamma_0(39))$ of weight 10 modular forms is of dimension 84, and $M_{11}(\Gamma_0(39))$ is of dimension 0. (There are never any modular forms of odd weight, and so $M_k(\Gamma) = 0$ for all odd k and any congruence subgroup Γ.)

13.6.1 Cohomology with rational coefficients (via resolutions)

To calculate cohomology $H^n(\Gamma, A)$ with coefficients in a $\mathbb{Q}\Gamma$-module A it suffices to construct a resolution of $\mathbb{Z}$ by non-free $\mathbb{Z}\Gamma$-modules where Γ acts with finite stabilizer groups on each module in the resolution. Computing over $\mathbb{Q}$ is computationally less expensive than computing over $\mathbb{Z}$. The following commands first compute $H^1(\Gamma_0(39), \mathbb{Q}) = H_1(\Gamma_0(39), \mathbb{Q}) = \mathbb{Q}^9$. As a larger example, they then compute $H^1(\Gamma_0(2^{13}-1), \mathbb{Q}) = \mathbb{Q}^{1365}$ where $\Gamma_0(2^{13}-1)$ has index $2^{13}+2 = 8192$ in $SL_2(\mathbb{Z})$.

Example

```
gap> K:=ContractibleGcomplex("SL(2,Z)");
Non-free resolution in characteristic 0 for SL(2,Integers) .

gap> gamma:=HAP_CongruenceSubgroupGamma0(39);;
gap> KK:=NonFreeResolutionFiniteSubgroup(K,gamma);
Non-free resolution in characteristic 0 for <matrix group with
```

```
18 generators> .

gap> C:=TensorWithRationals(KK);
gap> Homology(C,1);
9

gap> G:=HAP_CongruenceSubgroupGamma0(2^13-1);;
gap> IndexInSL2Z(G);
8192
gap> KK:=NonFreeResolutionFiniteSubgroup(K,G);;
gap> C:=TensorWithRationals(KK);;
gap> Homology(C,1);
1365
```

13.6.2 Cohomology with rational coefficients (via regular CW-complexes)

The following computes the quotient $Y = K/\Gamma_0(2^{17}-1)$ of the cubic tree K as a regular CW-complex and then computes $H^1(\Gamma_0(2^{17}-1),\mathbb{Q}) = H^1(Y,\mathbb{Q}) = \mathbb{Q}^{21845}$.

Example

```
gap> K:=BarycentricSubdivision(ContractibleGcomplex("SL(2,Z)"));;
gap> G:=HAP_CongruenceSubgroupGamma0(2^17-1);;
gap> Y:=GComplexToRegularCWComplex(K,G);
Regular CW-complex of dimension 1

gap> Length(Homology(Y,1));
21845
```

It is known that for primes p the cohomology $H^1(\Gamma_0(p),\mathbb{Q})$ has rank

$$1+2\left\{\frac{p+1}{12}-\frac{1+\left(\frac{-1}{p}\right)}{4}-\frac{1+\left(\frac{-3}{p}\right)}{3}\right\}$$

and the following provides confirmation of the rational cohomology computations.

Example

```
gap> f:=function(p)
> return 1 + 2* ((p+1)/12 - (1+Legendre(-1,p))/4 - (1+Legendre(-3,p))/3);
> end;;

gap> f(2^13-1);
1365
gap> f(2^17-1);
21845
```

13.7 Cuspidal cohomology

To define and compute cuspidal cohomology we consider the action of $SL_2(\mathbb{Z})$ on the upper-half plane $\mathfrak{h}$ given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} z = \frac{az+b}{cz+d}\,.$$

A standard 'fundamental domain' for this action is the region

$$\begin{aligned} D = \quad & \{z \in \mathfrak{h} \ : \ |z| > 1, |\mathrm{Re}(z)| < \tfrac{1}{2}\} \\ & \cup \{z \in \mathfrak{h} \ : \ |z| \geq 1, \mathrm{Re}(z) = -\tfrac{1}{2}\} \\ & \cup \{z \in \mathfrak{h} \ : \ |z| = 1, -\tfrac{1}{2} \leq \mathrm{Re}(z) \leq 0\} \end{aligned}$$

illustrated below.

The action factors through an action of $PSL_2(\mathbb{Z}) = SL_2(\mathbb{Z})/\langle \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \rangle$. The images of D under the action of $PSL_2(\mathbb{Z})$ cover the upper-half plane, and any two images have at most a single point in common. The possible common points are the bottom left-hand corner point which is stabilized by $\langle U \rangle$, and the bottom middle point which is stabilized by $\langle S \rangle$.

A congruence subgroup Γ has a 'fundamental domain' D_Γ equal to a union of finitely many copies of D, one copy for each coset in $\Gamma \setminus SL_2(\mathbb{Z})$. The quotient space $X = \Gamma \setminus \mathfrak{h}$ is not compact, and can be compactified in several ways. We are interested in the Borel-Serre compactification. This is a space X^{BS} for which there is an inclusion $X \hookrightarrow X^{BS}$ and this inclusion is a homotopy equivalence. One defines the *boundary* $\partial X^{BS} = X^{BS} - X$ and uses the inclusion $\partial X^{BS} \hookrightarrow X^{BS} \simeq X$ to define the cuspidal cohomology group, over the ground ring $\mathbb{C}$, as

$$H^n_{cusp}(\Gamma, P_{\mathbb{C}}(k-2)) = \ker(\ H^n(X, P_{\mathbb{C}}(k-2)) \to H^n(\partial X^{BS}, P_{\mathbb{C}}(k-2))\ ).$$

Strictly speaking, this is the definition of *interior cohomology* $H^n_!(\Gamma, P_{\mathbb{C}}(k-2))$ which in general contains the cuspidal cohomology as a subgroup. However, for congruence subgroups of $SL_2(\mathbb{Z})$ there is equality $H^n_!(\Gamma, P_{\mathbb{C}}(k-2)) = H^n_{cusp}(\Gamma, P_{\mathbb{C}}(k-2))$.

Working directly with the chain inclusion $C_*(\partial X^{BS}) \hookrightarrow C_*(X^{BS})$ would mean handling chain groups that are not free $SL_2(\mathbb{Z})$-modules due the cyclic torsion subgroups groups in $SL_2(\mathbb{Z})$. This can be done using regular CW-complexes, but here we avoid the technicality by opting to work with a contractible CW-complex $\widetilde{X^{BS}}$ on which Γ acts freely, and Γ-equivariant inclusion $\widetilde{\partial X^{BS}} \hookrightarrow \widetilde{X^{BS}}$. Computations on $\widetilde{X^{BS}}$ agree with those on X^{BS} modulo any 2- and 3-torsion. The definition of interior cohomology that we use, which coincides with the above definition when working over $\mathbb{C}$, is

$$H^n_!(\Gamma, A) = \ker(\ H^n(\mathrm{Hom}_{\mathbb{Z}\Gamma}(C_*(\widetilde{X^{BS}}), A))\ ) \to H^n(\ \mathrm{Hom}_{\mathbb{Z}\Gamma}(C_*(\widetilde{\partial X^{BS}}), A)\ ).$$

The following data is recorded and, using perturbation theory, is combined with free resolutions for C_4 and C_6 to constuct $C_*(\widetilde{X^{BS}})$. The subchain complex $C_*(\widetilde{\partial X^{BS}})$ is obtained by restricting the construction to cells e^0_2 and e^1_3

The following commands calculate

$$H^1_!(\Gamma_0(39), \mathbb{Z}) = \mathbb{Z}^6\ .$$

Example

```
gap> gamma:=HAP_CongruenceSubgroupGamma0(39);;
gap> k:=2;; deg:=1;; c:=InteriorCohomologyHomomorphism(gamma,deg,k);
[ g1, g2, g3, g4, g5, g6, g7, g8, g9 ] -> [ g1^-1*g3, g1^-1*g3, g1^-1*g3,
  g1^-1*g3, g1^-1*g2, g1^-1*g3, g1^-1*g4, g1^-1*g4, g1^-1*g4 ]
gap> AbelianInvariants(Kernel(c));
[ 0, 0, 0, 0, 0, 0 ]
```

From the Eichler-Shimura isomorphism and the already calculated dimension of $M_2(\Gamma_0(39)) \cong \mathbb{C}^9$, we deduce from this cuspidal cohomology that the space $S_2(\Gamma_0(39))$ of cuspidal weight 2 forms is of dimension 3, and the Eisenstein space $E_2(\Gamma_0(39)) \cong \mathbb{C}^3$ is of dimension 3.

The following commands show that the space $S_4(\Gamma_0(39))$ of cuspidal weight 4 forms is of dimension 12.

Example

```
gap> gamma:=HAP_CongruenceSubgroupGamma0(39);;
gap> k:=4;; deg:=1;; c:=InteriorCohomologyHomomorphism(gamma,deg,k);;
gap> AbelianInvariants(Kernel(c));
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]


gap> # Some additional information on torsion
gap> AbelianInvariants(Source(c));
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 2, 3 ]
gap> AbelianInvariants(Target(c));
[ 0, 0, 0, 0, 2, 2, 2, 2, 3, 3, 3, 3, 13, 13, 13, 13 ]
```

13.8 Hecke operators on forms of weight 2

A congruence subgroup $\Gamma \leq SL_2(\mathbb{Z})$ and element $g \in GL_2(\mathbb{Q})$ determine the subgroup $\Gamma' = \Gamma \cap g\Gamma g^{-1}$ and homomorphisms

$$\Gamma \hookleftarrow \Gamma' \xrightarrow{\gamma \mapsto g^{-1}\gamma g} g^{-1}\Gamma' g \hookrightarrow \Gamma .$$

These homomorphisms give rise to homomorphisms of cohomology groups

$$H^n(\Gamma,\mathbb{Z}) \xleftarrow{tr} H^n(\Gamma',\mathbb{Z}) \xleftarrow{\alpha} H^n(g^{-1}\Gamma' g,\mathbb{Z}) \xleftarrow{\beta} H^n(\Gamma,\mathbb{Z})$$

with α, β functorial maps, and tr the transfer map. We define the composite $T_g = tr \circ \alpha \circ \beta : H^n(\Gamma,\mathbb{Z}) \to H^n(\Gamma,\mathbb{Z})$ to be the *Hecke component* determined by g.

For $\Gamma = \Gamma_0(N)$, prime integer p coprime to N, and cohomology degree $n = 1$ we define the *Hecke operator* $T_p = T_g$ where $g = \begin{pmatrix} 1 & 0 \\ 0 & p \end{pmatrix}$. Further details on this description of Hecke operators can be found in [Ste07, Appendix by P. Gunnells].

The following commands compute T_2 and T_5 and $\Gamma = \Gamma_0(39)$. The commands also compute the eigenvalues of these two Hecke operators. The final command confirms that T_2 and T_5 commute. (It is a fact that $T_pT_q = T_qT_p$ for all p,q.)

Example

```
gap> gamma:=HAP_CongruenceSubgroupGamma0(39);;
gap> p:=2;;k:=2;;T2:=HeckeOperator(gamma,p,k);;
gap> Display(T2);
[ [  -2,  -2,   2,   2,   1,   2,   0,   0,   0 ],
  [  -2,   0,   1,   2,  -2,   2,   2,   2,  -2 ],
  [  -2,  -1,   2,   2,  -1,   2,   1,   1,  -1 ],
  [  -2,  -1,   2,   2,   1,   1,   0,   0,   0 ],
  [  -1,   0,   0,   2,  -3,   2,   3,   3,  -3 ],
  [   0,   1,   1,   1,  -1,   0,   1,   1,  -1 ],
```

```
  [  -1,   1,   1,  -1,   0,   1,   2,  -1,   1 ],
  [  -1,  -1,   0,   2,  -3,   2,   1,   4,  -1 ],
  [   0,   1,   0,  -1,  -2,   1,   1,   1,   2 ] ]
gap> Eigenvalues(Rationals,T2);
[ 3, 1 ]

gap> p:=5;;k:=2;;h:=HeckeOperator(gamma,p,k);;
gap> Display(T5);
[ [  -1,  -1,   3,   4,   0,   0,   1,   1,  -1 ],
  [  -5,  -1,   5,   4,   0,   0,   3,   3,  -3 ],
  [  -2,   0,   4,   4,   1,   0,  -1,  -1,   1 ],
  [  -2,   0,   3,   2,  -3,   2,   4,   4,  -4 ],
  [  -4,  -2,   4,   4,   3,   0,   1,   1,  -1 ],
  [  -6,  -4,   5,   6,   1,   2,   2,   2,  -2 ],
  [   1,   5,   0,  -4,  -3,   2,   5,  -1,   1 ],
  [  -2,  -2,   2,   4,   0,   0,  -2,   4,   2 ],
  [   1,   3,   0,  -4,  -4,   2,   2,   2,   4 ] ]
gap> Eigenvalues(Rationals,T5);
[ 6, 2 ]

gap>T2*T5=T5*T2;
true
```

13.9 Hecke operators on forms of weight ≥ 2

The above definition of Hecke operator T_p for $\Gamma = \Gamma_0(N)$ extends to a Hecke operator $T_p: H^1(\Gamma, P_{\mathbb{Q}}(k-2)) \to H^1(\Gamma, P_{\mathbb{Q}}(k-2))$ for $k \geq 2$. We work over the rationals since that is a setting of much interest. The following commands compute the matrix of $T_2: H^1(\Gamma, P_{\mathbb{Q}}(k-2)) \to H^1(\Gamma, P_{\mathbb{Q}}(k-2))$ for $\Gamma = SL_2(\mathbb{Z})$ and $k = 4$;

Example

```
gap> H:=HAP_CongruenceSubgroupGamma0(1);;
gap> h:=HeckeOperator(H,2,12);;Display(h);
[ [   2049,  -7560,      0 ],
  [      0,    -24,      0 ],
  [      0,      0,    -24 ] ]
```

13.10 Reconstructing modular forms from cohomology computations

Given a modular form $f: \mathfrak{h} \to \mathbb{C}$ associated to a congruence subgroup Γ, and given a compact edge e in the tessellation of $\mathfrak{h}$ (*i.e.* an edge in the cubic tree $\mathscr{T}$) arising from the above fundamental domain for $SL_2(\mathbb{Z})$, we can evaluate

$$\int_e f(z)\,dz\,.$$

In this way we obtain a cochain $f_1: C_1(\mathscr{T}) \to \mathbb{C}$ in $Hom_{\mathbb{Z}\Gamma}(C_1(\mathscr{T}), \mathbb{C})$ representing a cohomology class $c(f) \in H^1(Hom_{\mathbb{Z}\Gamma}(C_*(\mathscr{T}), \mathbb{C})) = H^1(\Gamma, \mathbb{C})$. The correspondence $f \mapsto c(f)$ underlies the Eichler-Shimura isomorphism. Hecke operators can be used to recover modular forms from cohomology classes.

The above defined Hecke operators restrict to operators on cuspidal cohomology. On the left-hand side of the Eichler-Shimura isomorphism Hecke operators restrict to operators $T_s: S_2(\Gamma) \to S_2(\Gamma)$ for $s \geq 1$.

Let us now restrict attention to $\Gamma = \Gamma_0(N)$. Consider the function $q = q(z) = e^{2\pi iz}$ which is holomorphic on $\mathbb{C}$. For any modular form $f(z) \in M_k(\Gamma_0(N))$ there are numbers a_s such that

$$f(z) = \sum_{s=0}^{\infty} a_s q^s$$

for all $z \in \mathfrak{h}$. The form f is a cusp form if $a_0 = 0$.

A non-zero cusp form $f \in S_2(\Gamma_0(N))$ is a cusp *eigenform* if it is simultaneously an eigenvector for the Hecke operators T_s for all $s = 1, 2, 3, \cdots$ coprime to the level N. A cusp eigenform is said to be *normalized* if its coefficient $a_1 = 1$. It turns out that if f is normalized then the coefficient a_s is an eigenvalue for T_s (see for instance [Ste07] for details). It can be shown [AL70] that $S_2(\Gamma_0(N))$ admits a "basis constructed from eigenforms".

This all implies that, in principle, we can construct an approximation to an explicit basis for the space $S_2(\Gamma_0(N))$ of cusp forms by computing eigenvalues for Hecke operators.

Suppose that we would like a basis for $S_2(\Gamma_0(11))$. The following commands first show that $H^1_!(\Gamma_0(11), \mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}$ from which we deduce that $S_2(\Gamma_0(11)) = \mathbb{C}$ is 1-dimensional and thus admits a basis of eigenforms. Then eigenvalues of Hecke operators are calculated to establish that the modular form

$$f = q - 2q^2 - q^3 + 2q^4 + q^5 + 2q^6 - 2q^7 + -2q^9 - 2q^{10} + \cdots$$

constitutes a basis for $S_2(\Gamma_0(11))$.

Example

```
gap> gamma:=HAP_CongruenceSubgroupGamma0(11);;
gap> AbelianInvariants(Kernel(InteriorCohomologyHomomorphism(gamma,1,2)));
[ 0, 0 ]

gap> T1:=HeckeOperator(gamma,1,2);; Display(T1);
[ [  1,  0,  0 ],
  [  0,  1,  0 ],
  [  0,  0,  1 ] ]
gap> T2:=HeckeOperator(gamma,2,2);; Display(T2);
[ [   3,  -4,   4 ],
  [   0,  -2,   0 ],
  [   0,   0,  -2 ] ]
gap> T3:=HeckeOperator(gamma,3,2);; Display(T3);
[ [   4,  -4,   4 ],
  [   0,  -1,   0 ],
  [   0,   0,  -1 ] ]
gap> T5:=HeckeOperator(gamma,5,2);; Display(T5);
[ [   6,  -4,   4 ],
  [   0,   1,   0 ],
  [   0,   0,   1 ] ]
gap> T7:=HeckeOperator(gamma,7,2);; Display(T7);
[ [   8,  -8,   8 ],
  [   0,  -2,   0 ],
  [   0,   0,  -2 ] ]
```

For a normalized eigenform $f = 1 + \sum_{s=2}^{\infty} a_s q^s$ the coefficients a_s with s a composite integer can be expressed in terms of the coefficients a_p for prime p. If r,s are coprime then $T_{rs} = T_r T_s$. If p is a prime that is not a divisor of the level N of $\Gamma_0(N)$ then $a_{p^m} = a_{p^{m-1}} a_p - p^{k-1} a_{p^{m-2}}$ where k is the weight. If the prime p divides N then $a_{p^m} = (a_p)^m$. It thus suffices to compute the coefficients a_p for prime integers p only.

The following commands establish that $S_{12}(SL_2(\mathbb{Z}))$ has a basis consisting of one cusp eigenform
$q - 24q^2 + 252q^3 - 1472q^4 + 4830q^5 - 6048q^6 - 16744q^7 + 84480q^8 - 113643q^9$
$-115920q^{10} + 534612q^{11} - 370944q^{12} - 577738q^{13} + 401856q^{14} + 1217160q^{15} + 987136q^{16}$
$-6905934q^{17} + 2727432q^{18} + 10661420q^{19} + \ldots$

Example

```
gap> R:=ResolutionSL2Z_alt(2);;
gap> G:=R!.group;;
gap> P:=HomogeneousPolynomials(G,14);
MappingByFunction( SL(2,Integers), <matrix group with
2 generators>, function( x ) ... end )
gap> Cohomology(HomToIntegralModule(R,P),1);
[ 2, 2, 156, 0, 0, 0 ]
gap> #Thus the space S_12 of cusp forms is of dimension 1


gap> G:=HAP_CongruenceSubgroupGamma0(1);;
gap> for p in [2,3,5,7,11,13,17,19] do
> T:=HeckeOperator(G,p,12);;
> Print("eigenvalues= ",Eigenvalues(Rationals,T), " and eigenvectors = ", Eigenvectors(Rationals,
> od;
eigenvalues= [ 2049, -24 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ] 1
eigenvalues= [ 177148, 252 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [ 0, 0, 1 ] ]
eigenvalues= [ 48828126, 4830 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [ 0, 0, 1
eigenvalues= [ 1977326744, -16744 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [ 0, 0
eigenvalues= [ 285311670612, 534612 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [ 0,
eigenvalues= [ 1792160394038, -577738 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ], [
eigenvalues= [ 34271896307634, -6905934 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ],
eigenvalues= [ 116490258898220, 10661420 ] and eigenvectors = [ [ 1, -2520/691, 0 ], [ 0, 1, 0 ],
```

13.11 The Picard group

Let us now consider the *Picard group* $G = SL_2(\mathbb{Z}[i])$ and its action on *upper-half space*

$$\mathfrak{h}^3 = \{(z,t) \in \mathbb{C} \times \mathbb{R} \mid t > 0\} .$$

To describe the action we introduce the symbol j satisfying $j^2 = -1$, $ij = -ji$ and write $z + tj$ instead of (z,t). The action is given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot (z+tj) \;=\; (a(z+tj)+b)\,(c(z+tj)+d)^{-1} \; .$$

Alternatively, and more explicitly, the action is given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot (z+tj) \;=\; \frac{(az+b)\overline{(cz+d)} + a\bar{c}t^2}{|cz+d|^2 + |c|^2 t^2} \;+\; \frac{t}{|cz+d|^2 + |c|^2 t^2}\, j \; .$$

A standard 'fundamental domain' D for this action is the following region (with some of the boundary points removed).

$$\{z+tj \in \mathfrak{h}^3 \mid 0 \leq |\mathrm{Re}(z)| \leq \frac{1}{2}, 0 \leq \mathrm{Im}(z) \leq \frac{1}{2}, z\bar{z}+t^2 \geq 1\}$$

The four bottom vertices of D are $a = -\frac{1}{2}+\frac{1}{2}i+\frac{\sqrt{2}}{2}j$, $b = -\frac{1}{2}+\frac{\sqrt{3}}{2}j$, $c = \frac{1}{2}+\frac{\sqrt{3}}{2}j$, $d = \frac{1}{2}+\frac{1}{2}i+\frac{\sqrt{2}}{2}j$.

The upper-half space $\mathfrak{h}^3$ can be retracted onto a 2-dimensional subspace $\mathscr{T} \subset \mathfrak{h}^3$. The space $\mathscr{T}$ is a contractible 2-dimensional regular CW-complex, and the action of the Picard group G restricts to a cellular action of G on $\mathscr{T}$.

Using perturbation techniques, the 2-complex $\mathscr{T}$ can be combined with free resolutions for the cell stabilizer groups to contruct a regular CW-complex X on which the Picard group G acts freely. The following commands compute the first few terms of the free $\mathbb{Z}G$-resolution $R_* = C_*X$. Then R_* is used to compute

$$H^1(G,\mathbb{Z}) = 0\,,$$

$$H^2(G,\mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}_2\,,$$

$$H^3(G,\mathbb{Z}) = \mathbb{Z}_6\,,$$

$$H^4(G,\mathbb{Z}) = \mathbb{Z}_4 \oplus \mathbb{Z}_{24}\,,$$

and compute a free presentation for G involving four generators and seven relators.

Example

```
gap> K:=ContractibleGcomplex("SL(2,O-1)");;
gap> R:=FreeGResolution(K,5);;
gap> Cohomology(HomToIntegers(R),1);
[  ]
gap> Cohomology(HomToIntegers(R),2);
[ 2, 2 ]
gap> Cohomology(HomToIntegers(R),3);
[ 6 ]
gap> Cohomology(HomToIntegers(R),4);
[ 4, 24 ]
gap> P:=PresentationOfResolution(R);
rec( freeGroup := <free group on the generators [ f1, f2, f3, f4 ]>,
  gens := [ 184, 185, 186, 187 ],
  relators := [ f1^2*f2^-1*f1^-1*f2^-1, f1*f2*f1*f2^-2,
      f3*f2^2*f1*(f2*f1^-1)^2*f3^-1*f1^2*f2^-2,
      f1*(f2*f1^-1)^2*f3^-1*f1^2*f2^-1*f3^-1,
      f4*f2*f1*(f2*f1^-1)^2*f4^-1*f1*f2^-1, f1*f4^-1*f1^-2*f4^-1,
      f3*f2*f1*(f2*f1^-1)^2*f4^-1*f1*f2^-1*f3^-1*f4*f2 ] )
```

We can also compute the cohomology of $G = SL_2(\mathbb{Z}[i])$ with coefficients in a module such as the module $P_{\mathbb{Z}[i]}(k)$ of degree k homogeneous polynomials with coefficients in $\mathbb{Z}[i]$ and with the action described above. For instance, the following commands compute

$$H^1(G, P_{\mathbb{Z}[i]}(24)) = (\mathbb{Z}_2)^4 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_8 \oplus \mathbb{Z}_{40} \oplus \mathbb{Z}_{80}\,,$$

$$H^2(G,P_{\mathbb{Z}[i]}(24)) = (\mathbb{Z}_2)^{24} \oplus \mathbb{Z}_{520030} \oplus \mathbb{Z}_{1040060} \oplus \mathbb{Z}^2\,,$$

$$H^3(G,P_{\mathbb{Z}[i]}(24)) = (\mathbb{Z}_2)^{22} \oplus \mathbb{Z}_4 \oplus (\mathbb{Z}_{12})^2\,.$$

Example

```
gap> G:=R!.group;;
gap> M:=HomogeneousPolynomials(G,24);;
gap> C:=HomToIntegralModule(R,M);;
gap> Cohomology(C,1);
[ 2, 2, 2, 2, 4, 8, 40, 80 ]
gap> Cohomology(C,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  520030, 1040060, 0, 0 ]
gap> Cohomology(C,3);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 12, 12
 ]
```

13.12 Bianchi groups

The *Bianchi groups* are the groups $G = PSL_2(\mathscr{O}_{-d})$ where d is a square free positive integer and $\mathscr{O}_{-d}$ is the ring of integers of the imaginary quadratic field $\mathbb{Q}(\sqrt{-d})$. More explicitly,

$$\mathscr{O}_{-d} = \mathbb{Z}\left[\sqrt{-d}\right] \qquad \text{if } d \equiv 1,2 \bmod 4\,,$$

$$\mathscr{O}_{-d} = \mathbb{Z}\left[\frac{1+\sqrt{-d}}{2}\right] \qquad \text{if } d \equiv 3 \bmod 4\,.$$

These groups act on upper-half space $\mathfrak{h}^3$ in the same way as the Picard group. Upper-half space can be tessellated by a 'fundamental domain' for this action. Moreover, as with the Picard group, this tessellation contains a 2-dimensional cellular subspace $\mathscr{T} \subset \mathfrak{h}^3$ where $\mathscr{T}$ is a contractible CW-complex on which G acts cellularly. It should be mentioned that the fundamental domain and the contractible 2-complex $\mathscr{T}$ are not uniquely determined by G. Various algorithms exist for computing $\mathscr{T}$ and its cell stabilizers. One algorithm due to Swan [Swa71a] has been implemented by Alexander Rahm [Rah10] and the output for various values of d are stored in HAP. Another approach is to use Voronoi's theory of perfect forms. This approach has been implemented by Sebastian Schoennenbeck [BCNS15] and, again, its output for various values of d are stored in HAP. The following commands combine data from Schoennenbeck's algorithm with free resolutions for cell stabiliers to compute

$$H^1(PSL_2(\mathscr{O}_{-6}),P_{\mathscr{O}_{-6}}(24)) = (\mathbb{Z}_2)^4 \oplus \mathbb{Z}_{12} \oplus \mathbb{Z}_{24} \oplus \mathbb{Z}_{9240} \oplus \mathbb{Z}_{55440} \oplus \mathbb{Z}^4\,,$$

$$H^2(PSL_2(\mathscr{O}_{-6}),P_{\mathscr{O}_{-6}}(24)) = \begin{array}{l} (\mathbb{Z}_2)^{26} \oplus (Z_6)^8 \oplus (Z_{12})^9 \oplus \mathbb{Z}_{24} \oplus (\mathbb{Z}_{120})^2 \oplus (\mathbb{Z}_{840})^3 \\ \oplus \mathbb{Z}_{2520} \oplus (\mathbb{Z}_{27720})^2 \oplus (\mathbb{Z}_{24227280})^2 \oplus (\mathbb{Z}_{411863760})^2 \\ \oplus \mathbb{Z}_{245443824374892865187742514283666449812984 0} \\ \oplus \mathbb{Z}_{14726629462493571911264550857019986988779040} \\ \oplus \mathbb{Z}^4 \end{array}\,,$$

$$H^3(PSL_2(\mathscr{O}_{-6}), P_{\mathscr{O}_{-6}}(24)) = (\mathbb{Z}_2)^{23} \oplus \mathbb{Z}_4 \oplus (\mathbb{Z}_{12})^2 \ .$$

Note that the action of $SL_2(\mathscr{O}_{-d})$ on $P_{\mathscr{O}_{-d}}(k)$ induces an action of $PSL_2(\mathscr{O}_{-d})$ provided k is even.

Example

```
gap> R:=ResolutionPSL2QuadraticIntegers(-6,4);
Resolution of length 4 in characteristic 0 for PSL(2,O-6) .
No contracting homotopy available.

gap> G:=R!.group;;
gap> M:=HomogeneousPolynomials(G,24);;
gap> C:=HomToIntegralModule(R,M);;
gap> Cohomology(C,1);
[ 2, 2, 2, 2, 12, 24, 9240, 55440, 0, 0, 0, 0 ]
gap> Cohomology(C,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 6, 6, 6, 6, 6, 6, 6, 6, 12, 12, 12, 12, 12, 12, 12, 12, 12, 24, 120, 120,
  840, 840, 840, 2520, 27720, 27720, 24227280, 24227280, 411863760, 411863760,
  24544382437489286518774251428366644981298 40,
  14726629462493571911264550857019986988779040, 0, 0, 0, 0 ]
gap> Cohomology(C,3);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 12,
  12 ]
```

We can also consider the coefficient module

$$P_{\mathscr{O}_{-d}}(k,\ell) = P_{\mathscr{O}_{-d}}(k) \otimes_{\mathscr{O}_{-d}} \overline{P_{\mathscr{O}_{-d}}(\ell)}$$

where the bar denotes a twist in the action obtained from complex conjugation. For an action of the projective linear group we must insist that $k+\ell$ is even. The following commands compute

$$H^2(PSL_2(\mathscr{O}_{-11}), P_{\mathscr{O}_{-11}}(5,5)) = (\mathbb{Z}_2)^8 \oplus \mathbb{Z}_{60} \oplus (\mathbb{Z}_{660})^3 \oplus \mathbb{Z}^6 ,$$

a computation which was first made, along with many other cohomology computationsfor Bianchi groups, by Mehmet Haluk Sengun [Sen11].

Example

```
gap> R:=ResolutionPSL2QuadraticIntegers(-11,3);;
gap> M:=HomogeneousPolynomials(R!.group,5,5);;
gap> C:=HomToIntegralModule(R,M);;
gap> Cohomology(C,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 60, 660, 660, 660, 0, 0, 0, 0, 0, 0 ]
```

The function `ResolutionPSL2QuadraticIntegers(-d,n)` relies on a limited data base produced by the algorithms implemented by Schoennenbeck and Rahm. The function also covers some cases covered by entering a sring "-d+I" as first variable. These cases correspond to projective special groups of module automorphisms of lattices of rank 2 over the integers of the imaginary quadratic number field $\mathbb{Q}(\sqrt{-d})$ with non-trivial Steinitz-class. In the case of a larger class group there are cases labelled "-d+I2",...,"-d+Ik" and the Ij together with O-d form a system of representatives of elements of the class group modulo squares and Galois action. For instance, the following commands compute

$$H_2(PSL(\mathscr{O}_{-21+I2}), \mathbb{Z}) = \mathbb{Z}_2 \oplus \mathbb{Z}^6 .$$

Example

```
gap> R:=ResolutionPSL2QuadraticIntegers("-21+I2",3);
Resolution of length 3 in characteristic 0 for PSL(2,O-21+I2)) .
No contracting homotopy available.

gap> Homology(TensorWithIntegers(R),2);
[ 2, 0, 0, 0, 0, 0, 0 ]
```

13.13 (Co)homology of Bianchi groups and $SL_2(\mathscr{O}_{-d})$

The (co)homology of Bianchi groups has been studied in papers such as [SV83] [Vog85] [Ber00] [Ber06] [RF13] [Rah13b] [Rah13a] [BLR20]. Calculations in these papers can often be verified by computer. For instance, the calculation

$$H_q(PSL_2(\mathscr{O}_{-15}),\mathbb{Z}) = \begin{cases} \mathbb{Z}^2 \oplus \mathbb{Z}_6 & q = 1, \\ \mathbb{Z} \oplus \mathbb{Z}_6 & q = 2, \\ \mathbb{Z}_6 & q \geq 3 \end{cases}$$

obtained in [RF13] can be verified as follows, once we note that Bianchi groups have virtual cohomological dimension 2 and, if all stabilizer groups are periodic with period dividing m, then the homology has period dividing m in degree ≥ 3.

Example

```
gap> K:=ContractibleGcomplex("SL(2,O-15)");;
gap> PK:=QuotientOfContractibleGcomplex(K,Group(-One(K!.group)));;
gap> for n in [0..2] do
> for k in [1..K!.dimension(n)] do
> Print( CohomologicalPeriod(K!.stabilizer(n,k))," ");
> od;od;
2  2  2  2  2  2  2  2  2  2  2  2  2  2
gap> R:=FreeGResolution(PK,5);;
gap> for n in [0..4] do
> Print("H_",n," = ", Homology(TensorWithIntegers(R),n),"\n");
> od;
H_0 = [ 0 ]
H_1 = [ 6, 0, 0 ]
H_2 = [ 6, 0 ]
H_3 = [ 6 ]
H_4 = [ 6 ]
```

All finite subgroups of $SL_2(\mathscr{O}_{-d})$ are periodic. Thus the above example can be adapted from PSL to SL for any square=free $d \geq 1$. For example, the calculation

$$H^q(SL_2(\mathscr{O}_{-2}),\mathbb{Z}) = \begin{cases} \mathbb{Z} & q = 1, \\ \mathbb{Z}_6 & q = 2 \bmod 4, \\ \mathbb{Z}_2 \oplus \mathbb{Z}_{12} & q = 3 \bmod 4 \\ \mathbb{Z}_2 \oplus \mathbb{Z}_{24} & q = 0 \bmod 4 (q > 0) \\ \mathbb{Z}_{12} & q = 1 \bmod 4 (q > 1) \end{cases}$$

obtained in [SV83] can be verified as follows.

Example

```
gap> K:=ContractibleGcomplex("SL(2,O-2)");;
gap> for n in [0..2] do
> for k in [1..K!.dimension(n)] do
> Print(CohomologicalPeriod(K!.stabilizer(n,k))," ");
> od;od;
2  4  2  4  2  2  2  2  2  2  2  2
gap> R:=FreeGResolution(K,11);;
gap> for n in [0..10] do
> Print("H^",n," = ", Cohomology(HomToIntegers(R),n),"\n");
> od;
H^0 = [ 0 ]
H^1 = [ 0 ]
H^2 = [ 6 ]
H^3 = [ 2, 12 ]
H^4 = [ 2, 24 ]
H^5 = [ 12 ]
H^6 = [ 6 ]
H^7 = [ 2, 12 ]
H^8 = [ 2, 24 ]
H^9 = [ 12 ]
H^10 = [ 6 ]
```

A quotient of a periodic group by a central subgroup of order 2 need not be periodic. For this reason the (co)homology of PSL can be a bit more tricky than SL. For example, the calculation

$$H^q(PSL_2(\mathscr{O}_{-13}),\mathbb{Z}) = \begin{cases} \mathbb{Z}^3 \oplus (\mathbb{Z}_2)^2 & q = 1, \\ \mathbb{Z}^2 \oplus \mathbb{Z}_4 \oplus (\mathbb{Z}_3)^2 \oplus \mathbb{Z}_2 & q = 2, \\ (\mathbb{Z}_2)^q \oplus (\mathbb{Z}_3)^2 & q = 3 \bmod 4 \\ (\mathbb{Z}_2)^q & q = 0 \bmod 4 (q > 0) \\ (\mathbb{Z}_2)^q & q = 1 \bmod 4 (q > 1) \\ (\mathbb{Z}_2)^q \oplus (\mathbb{Z}_3)^2 & q = 2 \bmod 4 (q > 2) \end{cases}$$

was obtained in [RF13]. The following commands verify the calculation in the first 34 degrees, but for a proof valid for all degrees one needs to analyse the computation to spot that there is a certain "periodicity of period 2" in the computations for $q \geq 3$. This analysis is done in [RF13].

Example

```
gap> K:=ContractibleGcomplex("SL(2,O-13)");;
gap> PK:=QuotientOfContractibleGcomplex(K,Group(-One(K!.group)));;
gap> for n in [0..2] do
> for k in [1..PK!.dimension(n)] do
> S:=SmallGroup(IdGroup(PK!.stabilizer( n, k )));
> Print( [n,k]," is periodic ",IsPeriodic(S),"\n  ");
> od;od;
[ 0, 1 ] is periodic true
[ 0, 2 ] is periodic true
[ 0, 3 ] is periodic true
[ 0, 4 ] is periodic true
[ 0, 5 ] is periodic true
[ 0, 6 ] is periodic true
[ 0, 7 ] is periodic false
```

```
[ 0, 8 ] is periodic false
[ 1, 1 ] is periodic true
[ 1, 2 ] is periodic true
[ 1, 3 ] is periodic true
[ 1, 4 ] is periodic true
[ 1, 5 ] is periodic true
[ 1, 6 ] is periodic true
[ 1, 7 ] is periodic true
[ 1, 8 ] is periodic true
[ 1, 9 ] is periodic true
[ 1, 10 ] is periodic true
[ 1, 11 ] is periodic true
[ 1, 12 ] is periodic true
[ 1, 13 ] is periodic true
[ 2, 1 ] is periodic true
[ 2, 2 ] is periodic true
[ 2, 3 ] is periodic true
[ 2, 4 ] is periodic true
[ 2, 5 ] is periodic true
[ 2, 6 ] is periodic true
[ 2, 7 ] is periodic true
[ 2, 8 ] is periodic true
[ 2, 9 ] is periodic true
[ 2, 10 ] is periodic true
[ 2, 11 ] is periodic true

gap> R:=ResolutionPSL2QuadraticIntegers(-13,35);;
gap> for n in [0..34] do
> Print("H_",n," = ", Homology(TensorWithIntegers(R),n),"\n");
> od;
H_0 = [ 0 ]
H_1 = [ 2, 2, 0, 0, 0 ]
H_2 = [ 6, 12, 0, 0 ]
H_3 = [ 2, 6, 6 ]
H_4 = [ 2, 2, 2, 2 ]
H_5 = [ 2, 2, 2, 2, 2 ]
H_6 = [ 2, 2, 2, 2, 6, 6 ]
H_7 = [ 2, 2, 2, 2, 2, 6, 6 ]
H_8 = [ 2, 2, 2, 2, 2, 2, 2, 2 ]
H_9 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_10 = [ 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_11 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_12 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_13 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_14 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_15 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_16 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_17 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_18 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_19 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_20 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_21 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

```
H_22 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_23 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_24 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_25 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_26 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_27 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6 ]
H_28 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_29 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_30 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6, 6
H_31 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 6
H_32 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
H_33 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
H_34 = [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2
```

The Lyndon-Hochschild-Serre spectral sequence $H_p(G/N,H_q(N,A)) \Rightarrow H_{p+q}(G,A)$ for the groups $G = SL_2(\mathcal{O}_{-d})$ and $N \cong C_2$ the central subgroup with $G/N \cong PSL_2(\mathcal{O}_{-d})$, and the trivial module $A = \mathbb{Z}_\ell$, implies that for primes $\ell > 2$ we have a natural isomorphism $H_n(PSL_2(\mathcal{O}_{-d}),\mathbb{Z}_\ell) \cong H_n(SL_2(\mathcal{O}_{-d}),\mathbb{Z}_\ell)$. It follows that we have an isomorphism of ℓ-primary parts $H_n(PSL_2(\mathcal{O}_{-d}),\mathbb{Z})_{(\ell)} \cong H_n(SL_2(\mathcal{O}_{-d}),\mathbb{Z})_{(\ell)}$. Since $H_n(SL_2(\mathcal{O}_{-d}),\mathbb{Z})_{(\ell)}$ is periodic in degrees ≥ 3 we can recover the 3-primary part of $H_n(PSL_2(\mathcal{O}_{-13}),\mathbb{Z})$ in all degrees $q \geq 1$ from the following computation by ignoring all 2-power factors in the output.

Example

```
gap> R:=ResolutionSL2QuadraticIntegers(-13,13);;
gap> for n in [3..12] do
> Print("H_",n," at prime p=3 is: ", Filtered(Homology(TensorWithIntegers(R),n), m->IsInt(m/3)),"
> od;
H_3 at prime p=3 is: [ 6, 24 ]
H_4 at prime p=3 is: [  ]
H_5 at prime p=3 is: [  ]
H_6 at prime p=3 is: [ 6, 12 ]
H_7 at prime p=3 is: [ 6, 24 ]
H_8 at prime p=3 is: [  ]
H_9 at prime p=3 is: [  ]
H_10 at prime p=3 is: [ 6, 12 ]
H_11 at prime p=3 is: [ 6, 24 ]
H_12 at prime p=3 is: [  ]
gap> #Ignore the 2-power factors in the output
```

The ring $\mathcal{O}_{-163}$ is an example of a principal ideal domain that is not a Euclidean domain. It seems that no complete calculation of $H_n(PSL_2(\mathcal{O}_{-163}),\mathbb{Z})$ is yet available in the literature. The following comands compute this homology in the first 31 degrees. The computation suggests a general formula in higher degrees. All but two of the stabilizer groups for the action of $PSL_2(\mathcal{O}_{-163})$ are periodic. The non-periodic group A_4 occurs twice in degree 0.

Example

```
gap> R:=ResolutionPSL2QuadraticIntegers(-163,32);;
gap> for n in [1..31] do
> Print("H_",n,"= ",Homology(TensorWithIntegers(R),n),"\n");
> od;
H_1= [ 0, 0, 0, 0, 0, 0, 0 ]
```

```
H_2= [ 2, 12, 0, 0, 0, 0, 0, 0 ]
H_3= [ 6 ]
H_4= [  ]
H_5= [ 2, 2, 2 ]
H_6= [ 2, 6 ]
H_7= [ 6 ]
H_8= [ 2, 2, 2, 2 ]
H_9= [ 2, 2, 2 ]
H_10= [ 2, 6 ]
H_11= [ 2, 2, 2, 2, 6 ]
H_12= [ 2, 2, 2, 2 ]
H_13= [ 2, 2, 2 ]
H_14= [ 2, 2, 2, 2, 2, 6 ]
H_15= [ 2, 2, 2, 2, 6 ]
H_16= [ 2, 2, 2, 2 ]
H_17= [ 2, 2, 2, 2, 2, 2, 2 ]
H_18= [ 2, 2, 2, 2, 2, 6 ]
H_19= [ 2, 2, 2, 2, 6 ]
H_20= [ 2, 2, 2, 2, 2, 2, 2, 2 ]
H_21= [ 2, 2, 2, 2, 2, 2, 2 ]
H_22= [ 2, 2, 2, 2, 2, 6 ]
H_23= [ 2, 2, 2, 2, 2, 2, 2, 2, 6 ]
H_24= [ 2, 2, 2, 2, 2, 2, 2, 2 ]
H_25= [ 2, 2, 2, 2, 2, 2, 2 ]
H_26= [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 6 ]
H_27= [ 2, 2, 2, 2, 2, 2, 2, 2, 6 ]
H_28= [ 2, 2, 2, 2, 2, 2, 2, 2 ]
H_29= [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
H_30= [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 6 ]
H_31= [ 2, 2, 2, 2, 2, 2, 2, 2, 6 ]
```

13.14 Some other infinite matrix groups

Analogous to the functions for Bianchi groups, HAP has functions

- `ResolutionSL2QuadraticIntegers(d,n)`
- `ResolutionSL2ZInvertedInteger(m,n)`
- `ResolutionGL2QuadraticIntegers(d,n)`
- `ResolutionPGL2QuadraticIntegers(d,n)`
- `ResolutionGL3QuadraticIntegers(d,n)`
- `ResolutionPGL3QuadraticIntegers(d,n)`

for computing free resolutions for certain values of $SL_2(\mathscr{O}_d)$, $SL_2(\mathbb{Z}[\frac{1}{m}])$, $GL_2(\mathscr{O}_d)$, $PGL_2(\mathscr{O}_d)$, $GL_3(\mathscr{O}_d)$ and $PGL_3(\mathscr{O}_d)$. Additionally, the function

- `ResolutionArithmeticGroup("string",n)`

can be used to compute resolutions for groups whose data (provided by Sebastian Schoennenbeck, Alexander Rahm and Mathieu Dutour) is stored in the directory `gap/pkg/Hap/lib/Perturbations/Gcomplexes` .

For instance, the following commands compute

$$H^1(SL_2(\mathscr{O}_{-6}), P_{\mathscr{O}_{-6}}(24)) = (\mathbb{Z}_2)^4 \oplus \mathbb{Z}_{12} \oplus \mathbb{Z}_{24} \oplus \mathbb{Z}_{9240} \oplus \mathbb{Z}_{55440} \oplus \mathbb{Z}^4,$$

$$H^2(SL_2(\mathscr{O}_{-6}), P_{\mathscr{O}_{-6}}(24)) = \begin{array}{l} (\mathbb{Z}_2)^{26} \oplus (Z_6)^7 \oplus (Z_{12})^{10} \oplus \mathbb{Z}_{24} \oplus (\mathbb{Z}_{120})^2 \oplus (\mathbb{Z}_{840})^3 \\ \oplus \mathbb{Z}_{2520} \oplus (\mathbb{Z}_{27720})^2 \oplus (\mathbb{Z}_{24227280})^2 \oplus (\mathbb{Z}_{411863760})^2 \\ \oplus \mathbb{Z}_{2454438243748928651877425142836664498129840} \\ \oplus \mathbb{Z}_{14726629462493571911264550857019986988779040} \\ \oplus \mathbb{Z}^4 \end{array},$$

$$H^3(SL_2(\mathscr{O}_{-6}), P_{\mathscr{O}_{-6}}(24)) = (\mathbb{Z}_2)^{58} \oplus (\mathbb{Z}_4)^4 \oplus (\mathbb{Z}_{12}) .$$

Example

```
gap> R:=ResolutionSL2QuadraticIntegers(-6,4);
Resolution of length 4 in characteristic 0 for SL(2,O-6) .
No contracting homotopy available.

gap> G:=R!.group;;
gap> M:=HomogeneousPolynomials(G,24);;
gap> C:=HomToIntegralModule(R,M);;
gap> Cohomology(C,1);
[ 2, 2, 2, 2, 12, 24, 9240, 55440, 0, 0, 0, 0 ]
gap> Cohomology(C,2);
gap> Cohomology(C,2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 6, 6, 6, 6, 6, 6, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 24, 120,
  120, 840, 840, 840, 2520, 27720, 27720, 24227280, 24227280, 411863760,
  411863760, 2454438243748928651877425142836664498129840,
  14726629462493571911264550857019986988779040, 0, 0, 0, 0 ]
gap> Cohomology(C,3);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4, 12, 12 ]
```

The following commands construct free resolutions up to degree 5 for the groups $SL_2(\mathbb{Z}[\frac{1}{2}])$, $GL_2(\mathscr{O}_{-2})$, $GL_2(\mathscr{O}_2)$, $PGL_2(\mathscr{O}_2)$, $GL_3(\mathscr{O}_{-2})$, $PGL_3(\mathscr{O}_{-2})$. The final command constructs a free resolution up to degree 3 for $PSL_4(\mathbb{Z})$.

Example

```
gap> R1:=ResolutionSL2ZInvertedInteger(2,5);
Resolution of length 5 in characteristic 0 for SL(2,Z[1/2]) .

gap> R2:=ResolutionGL2QuadraticIntegers(-2,5);
Resolution of length 5 in characteristic 0 for GL(2,O-2) .
No contracting homotopy available.

gap> R3:=ResolutionGL2QuadraticIntegers(2,5);
```

```
Resolution of length 5 in characteristic 0 for GL(2,O2) .
No contracting homotopy available.

gap> R4:=ResolutionPGL2QuadraticIntegers(2,5);
Resolution of length 5 in characteristic 0 for PGL(2,O2) .
No contracting homotopy available.

gap> R5:=ResolutionGL3QuadraticIntegers(-2,5);
Resolution of length 5 in characteristic 0 for GL(3,O-2) .
No contracting homotopy available.

gap> R6:=ResolutionPGL3QuadraticIntegers(-2,5);
Resolution of length 5 in characteristic 0 for PGL(3,O-2) .
No contracting homotopy available.

gap> R7:=ResolutionArithmeticGroup("PSL(4,Z)",3);
Resolution of length 3 in characteristic 0 for <matrix group with 655 generators> .
No contracting homotopy available.
```

13.15 Ideals and finite quotient groups

The following commands first construct the number field $\mathbb{Q}(\sqrt{-7})$, its ring of integers $\mathscr{O}_{-7} = \mathscr{O}(\mathbb{Q}(\sqrt{-7}))$, and the principal ideal $I = \langle 5+2\sqrt{-7}\rangle \triangleleft \mathscr{O}(\mathbb{Q}(\sqrt{-7}))$ of norm $\mathscr{N}(I) = 53$. The ring I is prime since its norm is a prime number. The primality of I is also demonstrated by observing that the quotient ring $R = \mathscr{O}_{-7}/I$ is an integral domain and hence isomorphic to the unique finite field of order 53, $R \cong \mathbb{Z}/53\mathbb{Z}$. (In a ring of quadratic integers *prime ideal* is the same as *maximal ideal*).

The finite group $G = SL_2(\mathscr{O}_{-7}/I)$ is then constructed and confirmed to be isomorphic to $SL_2(\mathbb{Z}/53\mathbb{Z})$. The group G is shown to admit a periodic $\mathbb{Z}G$-resolution of $\mathbb{Z}$ of period dividing 52.

Finally the integral homology

$$H_n(G,\mathbb{Z}) = \begin{cases} 0 & n \neq 3,7, \text{ for } 0 \leq n \leq 8, \\ \mathbb{Z}_{2808} & n = 3,7, \end{cases}$$

is computed.

Example

```
gap> Q:=QuadraticNumberField(-7);
Q(Sqrt(-7))

gap> OQ:=RingOfIntegers(Q);
O(Q(Sqrt(-7)))

gap> I:=QuadraticIdeal(OQ,5+2*Sqrt(-7));
ideal of norm 53 in O(Q(Sqrt(-7)))

gap> R:=OQ mod I;
ring mod ideal of norm 53

gap> IsIntegralRing(R);
true
```

```
gap> gens:=GeneratorsOfGroup( SL2QuadraticIntegers(-7) );;
gap> G:=Group(gens*One(R));;G:=Image(IsomorphismPermGroup(G));;
gap> StructureDescription(G);
"SL(2,53)"

gap> IsPeriodic(G);
true
gap> CohomologicalPeriod(G);
52

gap> GroupHomology(G,1);
[  ]
gap> GroupHomology(G,2);
[  ]
gap> GroupHomology(G,3);
[ 8, 27, 13 ]
gap> GroupHomology(G,4);
[  ]
gap> GroupHomology(G,5);
[  ]
gap> GroupHomology(G,6);
[  ]
gap> GroupHomology(G,7);
[ 8, 27, 13 ]
gap> GroupHomology(G,8);
[  ]
```

The following commands show that the rational prime 7 is not prime in $\mathcal{O}_{-5} = \mathcal{O}(\mathbb{Q}(\sqrt{-5}))$. Moreover, 7 totally splits in $\mathcal{O}_{-5}$ since the final command shows that only the rational primes 2 and 5 ramify in $\mathcal{O}_{-5}$.

Example

```
gap> Q:=QuadraticNumberField(-5);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,7);;
gap> IsPrime(I);
false

gap> Factors(Discriminant(OQ));
[ -2, 2, 5 ]
```

For $d < 0$ the rings $\mathcal{O}_d = \mathcal{O}(\mathbb{Q}(\sqrt{d}))$ are unique factorization domains for precisely

$$d = -1, -2, -3, -7, -11, -19, -43, -67, -163.$$

This result was conjectured by Gauss, and essentially proved by Kurt Heegner, and then later proved by Harold Stark.

The following commands construct the classic example of a prime ideal I that is not principal. They then illustrate reduction modulo I.

Example

```
gap> Q:=QuadraticNumberField(-5);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,[2,1+Sqrt(-5)]);
ideal of norm 2 in O(Q(Sqrt(-5)))

gap> 6 mod I;
0
```

13.16 Congruence subgroups for ideals

Given a ring of integers $\mathscr{O}$ and ideal $I \triangleleft \mathscr{O}$ there is a canonical homomorphism $\pi_I : SL_2(\mathscr{O}) \to SL_2(\mathscr{O}/I)$. A subgroup $\Gamma \leq SL_2(\mathscr{O})$ is said to be a *congruence subgroup* if it contains $\ker \pi_I$. Thus congruence subgroups are of finite index. Generalizing the definition in 13.1 above, we define the *principal congruence subgroup* $\Gamma(I) = \ker \pi_I$, and the congruence subgroup $\Gamma_0(I)$ consisting of preimages of the upper triangular matrices in $SL_2(\mathscr{O}/I)$.

The following commands construct $\Gamma = \Gamma_0(I)$ for the ideal $I \triangleleft \mathscr{O}\mathbb{Q}(\sqrt{-5})$ generated by 12 and $36\sqrt{-5}$. The group Γ has index 385 in $SL_2(\mathscr{O}\mathbb{Q}(\sqrt{-5}))$. The final command displays a tree in a Cayley graph for $SL_2(\mathscr{O}\mathbb{Q}(\sqrt{-5}))$ whose nodes represent a transversal for Γ.

Example

```
gap> Q:=QuadraticNumberField(-5);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,[36*Sqrt(-5), 12]);;
gap> G:=HAP_CongruenceSubgroupGamma0(I);
CongruenceSubgroupGamma0(ideal of norm 144 in O(Q(Sqrt(-5))))

gap> IndexInSL2O(G);
385

gap> HAP_SL2TreeDisplay(G);
```

The next commands first construct the congruence subgroup $\Gamma_0(I)$ of index 144 in $SL_2(\mathscr{O}\mathbb{Q}(\sqrt{-2}))$ for the ideal I in $\mathscr{O}\mathbb{Q}(\sqrt{-2})$ generated by $4+5\sqrt{-2}$. The commands then compute

$$H_1(\Gamma_0(I),\mathbb{Z}) = \mathbb{Z}_3 \oplus \mathbb{Z}_6 \oplus \mathbb{Z}_{30} \oplus \mathbb{Z}^8,$$

$$H_2(\Gamma_0(I),\mathbb{Z}) = (\mathbb{Z}_2)^9 \oplus \mathbb{Z}^7,$$

$$H_3(\Gamma_0(I),\mathbb{Z}) = (\mathbb{Z}_2)^9.$$

Example

```
gap> Q:=QuadraticNumberField(-2);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,4+5*Sqrt(-2));;
gap> G:=HAP_CongruenceSubgroupGamma0(I);
CongruenceSubgroupGamma0(ideal of norm 66 in O(Q(Sqrt(-2))))
```

```
gap> IndexInSL2O(G);
144

gap> R:=ResolutionSL2QuadraticIntegers(-2,4,true);;
gap> S:=ResolutionFiniteSubgroup(R,G);;

gap> Homology(TensorWithIntegers(S),1);
[ 3, 6, 30, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(TensorWithIntegers(S),2);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0 ]
gap> Homology(TensorWithIntegers(S),3);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

13.17 First homology

The isomorphism $H_1(G,\mathbb{Z}) \cong G_{ab}$ allows for the computation of first integral homology using computational methods for finitely presented groups. Such methods underly the following computation of

$$H_1(\Gamma_0(I),\mathbb{Z}) \cong \mathbb{Z} \oplus \mathbb{Z}_2 \oplus \cdots \oplus \mathbb{Z}_{124336989193985597}$$

where I is the prime ideal in the Gaussian integers generated by $-95+66\sqrt{-1}$.

Example

```
gap> Q:=QuadraticNumberField(-1);;
gap> OQ:=RingOfIntegers(Q);;
gap> I:=QuadraticIdeal(OQ,-95+66*Sqrt(-1));
ideal of norm 13381 in O(GaussianRationals)
gap> G:=HAP_CongruenceSubgroupGammaO(I);;
gap> AbelianInvariants(G);
[ 0, 2, 2, 2, 2, 3, 4, 5, 16, 31, 223, 857, 13913, 106103, 1267133821,
  141880858099, 1290077458421, 92119226692240157, 124336989193985597 ]
```

The command `AbelianInvariants(G)` can be quite time consuming. An alternative approach is to compute the first homology of G from a chain complex C using the command `ContractedComplex(C,b)`. This command inputs a chain complex and optionally a positive integer b and attempts to return a quasi-isomorphic chain complex with fewer generators in each degree. The optional integer b is used to limit the extent to which simplification of C takes place. If no integer b is entered then no limit is placed on the simplification procedure.

Example

```
gap> R:=ResolutionSL2QuadraticIntegers(-1,3,true);;
gap> S:=ResolutionFiniteSubgroup(R,G);;
gap> C:=TensorWithIntegers(S);;
gap> List([0..3],C!.dimension);
[ 13382, 53528, 93674, 93674 ]
gap> C:=ContractedComplex(C,20);;
gap> List([0..3],C!.dimension);
[ 1, 681, 11418, 50884 ]
gap> C:=ContractedComplex(C,40);;
```

```
gap> List([0..3],C!.dimension);
[ 1, 541, 10765, 50371 ]
gap> C:=ContractedComplex(C,60);;
gap> List([0..3],C!.dimension);
[ 1, 521, 10555, 50181 ]
gap> C:=ContractedComplex(C,80);;
gap> List([0..3],C!.dimension);
[ 1, 506, 10251, 49892 ]
gap> C:=ContractedComplex(C,100);;
gap> List([0..3],C!.dimension);
[ 1, 481, 9778, 49444 ]
gap> C:=ContractedComplex(C,200);;
gap> List([0..3],C!.dimension);
[ 1, 433, 7648, 47362 ]
gap> C:=ContractedComplex(C,300);;
gap> List([0..3],C!.dimension);
[ 1, 354, 6652, 46445 ]
gap> h:=Homology(C,1);
[ 2, 2, 2, 2, 4, 557596161692196308029082507705341945751992979834538960897220681209674496822564111
```

We write G^{ab}_{tors} to denote the maximal finite summand of the first homology group of G and refer to this as the *torsion subgroup*. Nicholas Bergeron and Akshay Venkatesh [Ber16] have conjectured relationships between the torsion in congruence subgroups Γ and the volume of their quotient manifold $\mathfrak{h}^3/\Gamma$. For instance, for the Gaussian integers they conjecture

$$\frac{\log|\Gamma_0(I)^{ab}_{tors}|}{\mathrm{Norm}(I)} \to \frac{\lambda}{18\pi},\ \lambda = L(2,\chi_{\mathbb{Q}(\sqrt{-1})}) = 1-\frac{1}{9}+\frac{1}{25}-\frac{1}{49}+\cdots$$

as the norm of the prime ideal I tends to ∞. The following approximates $\lambda/18\pi = 0.0161957$ and $\frac{\log|\Gamma_0(I)^{ab}_{tors}|}{\mathrm{Norm}(I)} = 0.0147988$ for the above example.

Example

```
gap> Q:=QuadraticNumberField(-1);;
gap> Lfunction(Q,2)/(18*3.142);
0.0161957

gap> Loge10:=0.434294481903;; #Log_10(e)
gap> T:=Product([ 2, 2, 2, 2, 3, 4, 5, 16, 31, 223, 857, 13913, 106103,
>                 1267133821, 141880858099, 1290077458421, 92119226692240157,
>                 124336989193985597 ]);;
gap> 1.0*Log(Product(T,10)/(Loge*Norm(I));
0.0147988
```

The link with volume is given by the Humbert volume formula

$$\mathrm{Vol}(\mathfrak{h}^3/PSL_2(\mathcal{O}_d)) = \frac{|D|^{3/2}}{24}\zeta_{\mathbb{Q}(\sqrt{d})}(2)/\zeta_{\mathbb{Q}}(2)$$

valid for square-free $d<0$, where D is the discriminant of $\mathbb{Q}(\sqrt{d})$. The volume of a finite index subgroup Γ is obtained by multiplying the right-hand side by the index $|PSL_2(\mathcal{O}_d) : \Gamma|$.

The following commands produce a graph of $\frac{\log|\Gamma_0(I)^{ab}_{tors}|}{\mathrm{Norm}(I)}$ against $\mathrm{Norm}(I)$ for prime ideals I of norm $49 \leq \mathrm{Norm}(I) \leq 6833$ (where one ideal for each norm is taken).

Example

```
gap> Q:=QuadraticNumberField(-1);;
gap> OQ:=RingOfIntegers(Q);;
gap> N:=QuadraticIntegersByNorm(OQ,20000);;

gap> ##########################################
gap> fn:=function(x);
> if IsRat(x) then return x; fi;
> return x!.rational+x!.irrational*Sqrt(-1);
> end;
gap> ##########################################

gap> NN:=List(N,fn);
gap> P:=Filtered(NN,x->IsPrime(QuadraticIdeal(OQ,x)));
gap> PP:=Classify(P,x->Norm(Q,x));
gap> PP:=List(PP,x->x[1]);;
gap> PP:=Filtered(PP,x->not x=0);

gap> Loge:=0.434294481903;; ###Log_10(e)
gap> ##########################################
gap> ffn:=function(x)
> local I, G, A, T, S, F;
> I:=QuadraticIdeal(OQ,x);
> G:=HAP_CongruenceSubgroupGamma0(I);;
> A:=AbelianInvariants(G);
> T:=Filtered(A,x->not x=0);
> return [Norm(Q,x),1.0*Log(Product(T),10)/(Loge*Norm(I))];
> end;
gap> ##########################################

gap> S:=List(PP{[9..447]},ffn);
gap> ScatterPlot(S);
```

13.18 Generators and relators for $SL_3(\mathbb{Z})$

The group $SL_n(\mathbb{Z})$ is well known to be generated by the n^2-n elementary matrices with a 1 in each diagonal entry, a 1 in a single off diagonal entry, and 0 in every remaining entry. The Steinberg relations provide a finite presentation for $SL_n(\mathbb{Z})$ on these generators. In fact, $SL_n(\mathbb{Z})$ admits a 2-generator presentation, one of order 2 and the other of infinite order, for all $n > 2$. For $n \geq 5$ there are also 2-generator presentations with generators of orders 2 and 3. Details and references can be found in [CLV25]. The paper [CLV25] begins with a 3-generator presentation for $SL_3(\mathbb{Z})$ and some 2-generator presentations derived from it.

The bottom dimensions of resolutions for $SL_3(\mathbb{Z})$ arising from CW-complexes can also be used to produce presentations. For instance, the following commands produce the presentation

$< x,y,z \mid x^2 = y^4 = z^2 = (xy)^3 = xyzxy^2xzy^{-1} =$
$xyzy^{-1}xyxy^{-1}xzy(xy^{-1})^2xzy =$

$$yxy^{-1}xyxy^{-2}x(yz)^2y^{-1}zyxy^{-1}x(yz)^2y^{-1}zy^{-1}xy = 1 >,$$

$$x = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ -4 & 0 & -1 \end{pmatrix}, y = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 1 \\ -3 & -3 & -2 \end{pmatrix}, z = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ -2 & -2 & -1 \end{pmatrix}.$$

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)a");;
gap> R:=FreeGResolution(K,2);;
gap> P:=PresentationOfResolution(R);;
gap> G:=P.freeGroup/P.relators;
<fp group on the generators [ x, y, z ]>
gap> GG:=SimplifiedFpGroup(G);
<fp group on the generators [ x, y, z ]>
gap> RelatorsOfFpGroup(GG);
[ x^2, z^2, y^4, (x*y)^3, x*y*z*x*y^2*x*z*y^-1,
  x*y*z*y^-1*x*y*x*y^-1*x*z*y*(x*y^-1)^2*x*z*y,
  y*x*y^-1*x*y*x*y^-2*x*(y*z)^2*y^-1*z*y*x*y^-1*x*(y*z)^2*y^-1*z*y^-1*x*y ]
gap> gens:=R!.elts{P.gens};
[ [ [ 1, 0, 0 ], [ 1, -1, 0 ], [ -4, 0, -1 ] ],
  [ [ 2, 1, 1 ], [ 1, 1, 1 ], [ -3, -3, -2 ] ],
  [ [ 0, 1, 0 ], [ 1, 0, 0 ], [ -2, -2, -1 ] ] ]
```

The next commands produce a 2-generator presentation on the generators $h = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}, m = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ of orders 6 and 3.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)s");;
gap> R:=FreeGResolution(K,3);;
gap> P:=PresentationOfResolution(R);;
gap> G:=P.freeGroup/P.relators;
<fp group on the generators [ f, g, h, k, m, n, p, q, r, s, t, u, v, w, x, y,
  z ]>
gap> GG:=SimplifiedFpGroup(G);
<fp group on the generators [ h, m ]>
gap> RelatorsOfFpGroup(GG);
[ m^3, h^6, (h*m^-1)^4, (m^-1*h)^4, (m^-1*h^3)^2*m^-1*h^-3, (h*m^-1*h)^4,
  (h*m^-1)^2*(h^-1*m*h^-1)^2*(m^-1*h)^2*m,
  m^-1*h^-1*m*h^-1*m^-1*h*(m*h^-1)^2*(m^-1*h)^2*h*m^-1*h^-1*m*h^-1,
  m^-1*(h^-1*m*h^-1)^2*m*h*(m^-1*h^-1*m*h^-1)^2*m^-1*h*m^-1,
  (h^-1*m^-1*h*m^-1)^2*h^2*m^-1*h*m*h*m^-1*h^-2*m*h^-1*m^-1*h^-1*m,
  h^-1*m^-1*h^2*m^-1*h*m*h*(m^-1*h^-2)^2*m^-1*h*m*h*m^-1*h^2*m^-1,
  h*m^-1*h^-1*m*h^-2*m*h*m^-1*h^2*m*h*m^-1*(h^-1*m)^2*h*m^-1*h*m*h,
  (h^-1*m)^2*h*m*h^-1*m^-1*h^-1*m*h^-2*(m*h^-1*m*h)^2*m^-1*h*m^-1,
  m^-1*h^-1*m*h^-2*m^-1*(h*m^-1*h)^2*m^-1*h^-2*m*h*m^-1*h^-1*m*h^-2*m*h*m^-1*(\
h^2*m^-1*h)^2*(m*h^-1)^2*h^-1*m*h^2*(h*m^-1)^3*h^-2 ]

gap> gens:=R!.elts{P.gens{[3,5]}};
[ [ [ 1, 0, 0 ], [ 0, 0, -1 ], [ 0, 1, 1 ] ],
  [ [ 0, 0, 1 ], [ 1, 0, 0 ], [ 0, 1, 0 ] ] ]
gap> List(gens,Order);
```

```
[ 6, 3 ]
```

13.18.1 Expressing elements of $SL_3(\mathbb{Z})$ as words in generators

The generating matrices x,y,z for $SL_3(\mathbb{Z})$ given in Section 13.18 arise from the 1-skeleton of the CW-complex underlying the resolution constructed by applying `R:=FreeGResolution(K,2)` to the well-rounded retract K for $SL_3(\mathbb{Z})$. The retract is stored in the file `"SL3Za"` and accessed using `K:=ContractibleGcomplex("SL3(Z)")`. For certain computations we need to express an element $w \in SL_3(\mathbb{Z})$ as a word in the generators.

One naive approach is to use gaussian elimination to first express w as a product of elementary matrices $E_{ij}(1)$, $E_{ij}(-1)$ with ± 1 in the ij-position, and then replace each elementary matrix by its expression as a word in x,y,z. This is easy to implement, but obvously produces unnecessarily long expressions for w.

A better approach is to use the six generators

$$g_1 = \begin{pmatrix} -1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \; g_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \; g_3 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & -1 \end{pmatrix} \; g_4 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \; g_5 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \; g_6 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

for $SL_3(\mathbb{Z})$ corresponding to the six 'facet pairings' of the Voronoi domain (whose closure is a 6-dimensional simplex) and then to use Voronoi reduction in place of gaussian elimination. The following code uses this technique to express our previous generators in terms of the g_i:

$x = g_3 g_1^{-1} g_2^2 g_1 g_3^{-1} g_5^{-1} g_3^{-2} g_1^{-1} g_2^{-1} g_4^{-1}$,
$y = g_3 g_1^{-1} g_2^2 g_1 g_3^{-1} g_5^{-1} g_3^{-1} g_4^{-1} g_1^{-1} g_3^{-1}$,
$z = g_3 g_1^{-1} g_2^2 g_1 g_3^{-1} g_1^{-1} i g_3^{-1}$,
$h = g_2^2 g_3 g_1^{-1} g_6^{-1}$,
$m = g_6^{-1}$.

Example

```
gap> x:=[ [ 1, 0, 0 ], [ 1, -1, 0 ], [ -4, 0, -1 ] ];;
gap> y:=[ [ 2, 1, 1 ], [ 1, 1, 1 ], [ -3, -3, -2 ] ];;
gap> z:=[ [ 0, 1, 0 ], [ 1, 0, 0 ], [ -2, -2, -1 ] ];;
gap> h:=[ [ 1, 0, 0 ], [ 0, 0, -1 ], [ 0, 1, 1 ] ];;
gap> m:=[ [ 0, 0, 1 ], [ 1, 0, 0 ], [ 0, 1, 0 ] ];;
gap> VoronoiFactorize(SL(3,Integers),x);
[ 3, -1, 2, 2, 1, -3, -5, -3, -3, -1, -3, -1, -2, -4 ]
gap> VoronoiFactorize(SL(3,Integers),y);
[ 3, -1, 2, 2, 1, -3, -5, -3, -4, -1, -3 ]
gap> VoronoiFactorize(SL(3,Integers),z);
[ 3, -1, 2, 2, 1, -3, -1, -3 ]
gap> VoronoiFactorize(SL(3,Integers),h);
[ 2, 2, 3, -1, -6 ]
gap> VoronoiFactorize(SL(3,Integers),m);
[ -5 ]
```

The following code shows that Voronoi reduction expresses a particular random product of 1000 generators g_i as a word of length 426 in the g_i and g_i^{-1}. An analogus Voronoi reduction is then applied to

a random product of generators for $SL_2(\mathbb{Z})$ to express it as a word of length 192.

Example

```
gap> G:=SL(3,Integers);;
gap> w:=[];;
gap> for i in [1..1000] do
> Add(w,Random(Concatenation([-6..-1],[1..6])));
> od;
gap> m:=MatrixFromWord(VoronoiGenerators(G),w);
[ [ -44674790530437562, -24123269560333493, -34694674085693519 ],
  [ -10804815618994915, -5834330203498842, -8391075861936847 ],
  [ -13747757726719023, -7423445338055900, -10676579969918880 ] ]
gap> e:=VoronoiFactorize(G,m);;
gap> Length(e);
426
gap> MatrixFromWord(VoronoiGenerators(G),e)=m;
true


gap> G:=SL(2,Integers);;
gap> w:=[];;
gap> for i in [1..1000] do
> Add(w,Random([-2,-1,1,2]));
> od;
gap> m:=MatrixFromWord(VoronoiGenerators(G),w);
[ [ -4403553319158184826647, 937258106083950248383 ],
  [ -941180019957335454351 1, 2003219988620237311896 ] ]
gap> e:=VoronoiFactorize(G,m);;
gap> Length(e);
192
gap> MatrixFromWord(VoronoiGenerators(G),e)=m;
true
```

Chapter 14

Fundamental domains for Bianchi groups

14.1 Bianchi groups

The *Bianchi groups* are the groups $G_{-d} = PSL_2(\mathscr{O}_{-d})$ where d is a square free positive integer and $\mathscr{O}_{-d}$ is the ring of integers of the imaginary quadratic field $\mathbb{Q}(\sqrt{-d})$. These groups act on *upper-half space*

$$\mathfrak{h}^3 = \{(z,t) \in \mathbb{C} \times \mathbb{R} \mid t > 0\}$$

by the formula

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot (z+tj) \;=\; (a(z+tj)+b)\,(c(z+tj)+d)^{-1}$$

where we use the symbol j satisfying $j^2 = -1$, $ij = -ji$ and write $z+tj$ instead of (z,t). Alternatively, the action is given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot (z+tj) \;=\; \frac{(az+b)\overline{(cz+d)} + a\bar{c}t^2}{|cz+d|^2+|c|^2t^2} \;+\; \frac{t}{|cz+d|^2+|c|^2t^2}\, j\,.$$

We take the boundary $\partial\mathfrak{h}^3$ to be the Riemann sphere $\mathbb{C} \cup \infty$ and let $\overline{\mathfrak{h}}^3$ denote the union of $\mathfrak{h}^3$ and its boundary. The action of G_{-d} extends to the boundary. The element ∞ and each element of the number field $\mathbb{Q}(\sqrt{-d})$ are thought of as lying in the boundary $\partial\mathfrak{h}^3$ and are referred to as *cusps*. Let X denote the union of $\mathfrak{h}^3$ with the set of cusps, $X = \mathfrak{h}^3 \cup \{\infty\} \cup \mathbb{Q}(\sqrt{-d})$. It follows from work of Bianchi and Humbert that the space X admits the structure of a regular CW-complex (depending on d) for which the action of G_{-d} on $\mathfrak{h}^3$ extends to a cellular action on X which permutes cells. Moreover, G_{-d} acts transitively on the 3-cells of X and each 3-cell has trivial stabilizer in G_{-d}. Details are provided in Richard Swan's paper [Swa71b].

We refer to the closure in X of any one of these 3-cells as a *fundamental domain* for the action G_{-d}. Cohomology of G_{-d} can be computed from a knowledge of the combinatorial structure of this fundamental domain together with a knowledge of the stabilizer groups of the cells of dimension ≤ 2.

14.2 Swan's description of a fundamental domain

A pair (a,b) of elements in $\mathscr{O}_{-d}$ is said to be *unimodular* if the ideal generated by a,b is the whole ring $\mathscr{O}_{-d}$ and $a \neq 0$. A unimodular pair can be represented by a hemisphere in $\overline{\mathfrak{h}}^3$ with base centred

at the point $b/a \in \mathbb{C}$ and of radius $|1/a|$. The radius is ≤ 1. Think of the points in $\mathfrak{h}^3$ as lying strictly above $\mathbb{C}$. Let B denote the space obtained by removing the interiors of all such hemispheres from $\mathfrak{h}^3$.

When $d \equiv 3 \bmod 4$ let F be the subspace of $\overline{\mathfrak{h}}^3$ consisting of the points $x+iy+jt$ with $-1/2 \leq x \leq 1/2$, $-1/4 \leq y \leq 1/4$, $t \geq 0$. Otherwise, let F be the subspace of $\overline{\mathfrak{h}}^3$ consisting of the points $x+iy+jt$ with $-1/2 \leq x \leq 1/2$, $-1/2 \leq y \leq 1/2$, $t \geq 0$.

It is explained in [Swa71b] that $F \cap B$ is a 3-cell in the above mentioned regular CW-complex structure on X.

14.3 Computing a fundamental domain

Explicit fundamental domains for certain values of d were calculated by Bianchi in the 1890s and further calculations were made by Swan in 1971 [Swa71b]. In the 1970s, building on Swan's work, Robert Riley developed a computer program for computing fundamental domains of certain Kleinian groups (including Bianchi groups). In their 2010 PhD theses Alexander Rahm and M.T. Aranes independently developed Pari/GP and Sage software based on Swan's ideas. In 2011 Dan Yasaki used a different approach based on Voronoi's theory of perfect forms in his Magma software for fundamental domains of Bianchi groups. Aurel Page developed software for fundamental domains of Kleinian groups in his 2010 masters thesis. In 2018 Sebastian Schoennenbeck used a more general approach based on perfect forms in his Magma software for computing fundamental domains of Bianchi and other groups. Output from the code of Alexander Rahm and Sebastian Schoennenbeck for certain Bianchi groups has been stored iin HAP for use in constructing free resolutions.

More recently a GAP implementation of Swan's algorithm has been included in HAP. The implementation uses exact computations in $\mathbb{Q}(\sqrt{-d})$ and in $\mathbb{Q}(\sqrt{d})$. A bespoke implementation of these two fields is part of the implementation so as to avoid making apparently slower computations with cyclotomic numbers. The account of Swan's algorithm in the thesis of Alexander Rahm was the main reference during the implementation.

14.4 Examples

The fundamental domain $D = \overline{F \cap B}$ (where the overline denotes closure) has boundary ∂D involving the four vertical quadrilateral 2-cells contained in the four vertical quadrilateral 2-cells of ∂F. We refer to these as the *vertical 2-cells* of D. When visualizing D we ignore the 3-cell and the four vertical 2-cells entirely and visualize only the remaining 2-cells. These 2-cells can be viewed as a 2-dimensional image by projecting them onto the complex plane, or they can be viewed as an interactive 3-dimensional image.

A fundamental domain for G_{-39} can be visualized using the following commands.

Example

```
gap> D:=BianchiPolyhedron(-39);
3-dimensional Bianchi polyhedron over OQ( Sqrt(-39) )
involving hemispheres of minimum squared radius 1/39
and non-cuspidal vertices of minimum squared height 1/49 .

gap> Display3D(D);;
gap> Display2D(D);;
```

A *cusp vertex* of D is any vertex of D lying in $\mathbb{C} \cup \infty$. In the above visualizations for G_{-39} several cusp vertices in $\mathbb{C}$ are : in the 2-dimensional visualization they are represented by red dots. Computer calculations show that these cusps lie in precisely three orbits under the action of G_{-d}. Thus, together with the orbit of ∞ there are four distinct orbits of cusps. By the well-known correspondence between cusp orbits and elements of the class group it follows that the class group of $\mathbb{Q}(\sqrt{-39})$ is of order 4.

The following additional commands comvert the "bottom surface" of the Bianchi polyhedron D to a 2-dimensional regular CW-complex and then displays the 1-skeleton of this 2-complex.

Example

```
gap> D:=BianchiPolyhedron(-39);;
gap> Y:=RegularCWComplex(D);
Regular CW-complex of dimension 2

gap> Display(GraphOfRegularCWComplex(Y));
```

A fundamental domain for G_{-22} can be visualized using the following commands.

Example

```
gap> D:=BianchiPolyhedron(-22);;
gap> Display3D(OQ,D);;
gap> Display2D(OQ,D);;
```

Two cusps are visible in these visualizations for G_{-22}. They lie in a single orbit. Thus, together with the orbit of ∞, there are two orbits of cusps for this group.

An alternative visualization for G_{-22} can be produced, in which the cells in a given orbit are coloured with a common colour. The following commands produce this visualization.

Example

```
gap> K:=BianchiGcomplex(-22);;
gap> Display3D(K);
```

A fundamental domain for G_{-163} can be visualized using the following commands.

Example

```
gap> D:=BianchiPolyhedron(-163);;
gap> Display3D(OQ,D);;
gap> Display2D(OQ,D);;
gap> K:=BianchiGcomplex(-163);;
gap> Display3D(K);
```

There is just a single orbit of cusps in this example, the orbit containing ∞, since $\mathbb{Q}(\sqrt{-163})$ is a principal ideal domain and hence has trivial class group.

A fundamental domain for G_{-33} is visualized using the following commands.

Example

```
gap> D:=BianchiPolyhedron(-33);;
gap> Display3D(OQ,D);;
gap> Display2D(OQ,D);;
```

14.5 Establishing correctness of a fundamental domain

The cusps of a fundamental domain can be calculated independently of the domain computation. The remaining vertices of the domain will have positive heights. To prove that the computation is correct we need to establish that no non-cuspidal vertex lies below any hemishpere centered on the complex plane at $b/a \in \mathbb{C}$ with (a,b) a unimodular pair. As these hemispheres have increasingly smaller radius we only need to check those finitely many hemispheres with radius smaller than the height of the lowest non-cuspidal vertex.

For a few values of d the smallest radius r of a hemisphere contributing to the fundamental domain boundary has been stored. For cases where this smallest radius is not stored a very slow method for finding r is implemented and the user is advised to speed things up by guessiing a value $N = 1/r^2$ and then test that this value of N is indeed large enough. The following commands illustrate this for $d = -46$ with a guess of $N = 600$. Once the test is done we can see that in fact a smaller guess of $N = 441$ would have sufficed.

Example

```
gap> P:=BianchiPolyhedron(-46);
Try
  P:=BianchiPolyhedron(OQ,N);
for some guessed positive integer value of N and then try
  SwanBianchiCriterion(P);
to test if the value of N was large enough. If the test returns false then you\
'll need to try a larger value of N.

A successful value of N can be stored as a pair [d,N] in the list HAPRECORD wh\
ich can be edited manually in the file hap/lib/Congruence/bianchi.gi .


gap> P:=BianchiPolyhedron(-46,600);
3-dimensional Bianchi polyhedron over OQ( Sqrt(
-46) ) involving hemispheres of minimum squared radius 1/
441 and non-cuspidal vertices of minimum squared height 1/8280 .

gap> SwanBianchiCriterion(P);
true
```

14.6 Computing quotient 2-complexes

The space $X = \mathfrak{h}^3 \cup \{\infty\} \cup \mathbb{Q}(\sqrt{-d})$ retracts onto an $SL_2(\mathcal{O}_{-d})$-equivariant 2-dimensional regular CW-complex $K \subset X$. By barycentrically subdividing K we can assume that the group action on K is *rigid* in the sense that any group element in the stabilizer of a cell fixes that cell pointwise. Given a congruence subgroup $\Gamma \leq SL_2(\mathcal{O}_{-d})$ the quotient space $Y = K/\Gamma$ inherits a regular CW-structure. We have $H_n(\Gamma,\mathbb{Q}) = H_n(Y,\mathbb{Q})$ and $H_n(\Gamma,\mathbb{Z}_p) = H_n(Y,\mathbb{Z}_p)$ for any prime not dividing the order of any group element in Γ. At these primes the p-torsion in $H_n(Y,\mathbb{Z})$ equals that in $H_n(\Gamma,\mathbb{Z})$. The following session computes the integral homology of Y for $\Gamma = \Gamma_0(71) \leq SL(2,\mathcal{O}_{-2})$ and establishes that Γ has no p-torsion for primes $p > 3$ and has index 5042.

Example

```
gap> K:=BianchiGcomplex(-2);;
gap> iso:=K!.isomorphismToGroupCyclotomicMats;;
```

```
gap> KK:=BarycentricSubdivision(K);; #KK is a rigid complex
gap> ResolutionIsomorphismGroup(KK,iso);; #matrices in G=SL(2,O-2) now have
gap>                                      #cyclotomic entries

gap> OQ:=RingOfIntegers(QuadraticNumberField(-2));;
gap> I:=QuadraticIdeal(OQ,71);;
gap> N:=HAP_CongruenceSubgroupGammaO(I);; #A congruence subgroup of G
gap> IndexInSL2O(N);
5042

gap> Y:=GComplexToRegularCWComplex(KK,N); #Y = KK/N is a quotient CW-complex
Regular CW-complex of dimension 2

gap> Homology(Y,0);
[ 0 ]
gap> Homology(Y,1);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 4, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 16, 16,
  16, 16, 16, 16, 16, 16, 16, 32, 32, 96, 672, 84717713505488213136786 24,
  42358856752744106568393120, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0 ]
gap> Homology(Y,2);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

gap> Flat(List([0..2], n->
>               List([1..K!.dimension(n)],k->Order(K!.stabilizer(n,k)))  ));
[ 24, 6, 8, 4 ]
gap> Factors(84717713505488213136786 24);
[ 2, 2, 2, 2, 2, 3, 7, 17, 17, 17, 19, 617, 873263, 250653841 ]
gap> Factors(42358856752744106568393120);
[ 2, 2, 2, 2, 2, 3, 5, 7, 17, 17, 17, 19, 617, 873263, 250653841 ]
```

The above computation can be corroborated by using a free resolution to calculate $H_1(\Gamma_0(71),\mathbb{Z})$ at all primes.

Example

```
gap> R:=FreeGResolution(KK,2);;
gap> S:=ResolutionFiniteSubgroup(R,N);;
gap> C:=TensorWithIntegers(S);;
gap> D:=ContractedComplex(C);;
gap> Homology(D,1);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
  4, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 16, 16, 16, 16,
  16, 16, 16, 16, 16, 16, 32, 96, 96, 672, 84717713505488213136786 24,
  12707657025823231970517936 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0 ]
```

The next example computes a CW complex Y with $H^*(Y,\mathbb{Z}) = H^*(\Gamma_0(23),\mathbb{Z})$ modulo the primes $2,3$ for $\Gamma_0(23) < SL_2(\mathscr{O}_{-2})$ and shows that a basis $b_1, b_2, \ldots b_9$ for $H^1(W,\mathbb{Z}) = \mathbb{Z}^9$ satisfies $b_i \cup b_j = 0$ except for $b_1 \cup b_6 = -b_6 \cup b_1 \neq 0$, $b_1 \cup b_7 = -b_7 \cup b_1 \neq 0$, $b_1 \cup b_9 = -b_9 \cup b_1 \neq 0$.

Example

```
gap> I:=QuadraticIdeal(OQ,23);;
gap> N:=HAP_CongruenceSubgroupGamma0(I);;
gap> Y:=GComplexToRegularCWComplex(KK,N);;
gap> Y:=SimplifiedComplex(Y);;
gap> Cohomology(Y,1);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> B:=IdentityMat(9);; # represents the 9 cohomology generators
gap> C:=NullMat(9,9);;
gap> cup:=CupProduct(Y);;
gap> for i in [1..9] do
> for j in [j+1..9] do
> if not IsZero(cup(1,1,B[i],B[j])) then C[i][j]:=1; C[j,i]:=1; fi;
> od;od;
gap> Display(C);
[ [  0,  0,  0,  0,  0,  1,  1,  0,  1 ],
  [  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  1,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  1,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  1,  0,  0,  0,  0,  0,  0,  0,  0 ] ]
```

THe following additional commands compute the 11-primary parts $H^2(\Gamma_0(23),\mathbb{Z})_{(11)} = \mathbb{Z}_{11}$, $H^n(\Gamma_0(23),\mathbb{Z})_{(11)} = 0$ for $2 \neq n > 0$. They then compute the cohomology ring $A = H^*(\Gamma_0(23),\mathbb{Z}_{11})$, a basis B of $H^1(\Gamma_0(23),\mathbb{Z}_{11}) = (\mathbb{Z}_{11})^{10}$, and procuce a matrix $C = (c_{ij})$ with $c_{ij} = 0$ if $b_i \cup b_j = 0$, $c_{ij} = 1$ if $b_i \cup b_j \neq 0$ for $b_i, b_j \in B$.

Example

```
gap> Cohomology(Y,2);
[ 2, 2, 2, 4, 4, 4, 4, 4, 44, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> A:=CohomologyRing(Y,11);
<algebra of dimension 20 over GF(11)>
gap> B:=Basis(A);;
gap> B:=Filtered(B,b->A!.degree(b)=1);;
gap> C:=NullMat(Length(B),Length(B));;
gap> for i in [1..Length(B)] do
> for j in [1..Length(B)] do
> if not IsZero(B[i]*B[j]) then C[i][j]:=1; fi;
> od;od;
gap> Display(C);
[ [  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 ],
  [  0,  0,  0,  0,  0,  0,  0,  1,  1,  1 ],
  [  0,  0,  0,  0,  0,  0,  0,  1,  1,  1 ],
  [  0,  0,  0,  0,  0,  1,  1,  1,  1,  1 ],
  [  0,  0,  0,  0,  0,  0,  0,  1,  1,  1 ],
  [  0,  0,  0,  1,  0,  0,  1,  1,  1,  1 ],
```

```
  [  0,  0,  0,  1,  0,  1,  0,  1,  1,  1 ],
  [  0,  1,  1,  1,  1,  1,  1,  0,  1,  1 ],
  [  0,  1,  1,  1,  1,  1,  1,  1,  0,  1 ],
  [  0,  1,  1,  1,  1,  1,  1,  1,  1,  0 ] ]
```

14.7 Computing a free resolution for $SL_2(\mathscr{O}_{-d})$

The above fundamental domains can be used to construct free resolutions for $SL_2(\mathscr{O}_{-d})$ and $PSL_2(\mathscr{O}_{-d})$. The following commands illustrate the computation of free resolutions for $SL_2(\mathscr{O}_{-43})$ and $SL_2(\mathscr{O}_{-10})$ and $SL_2(\mathscr{O}_{-14})$ and their integral homology (which in each case is periodic of period dividing 4 in degrees ≥ 3). The computation of fundamental domains uses exact arithmetic in the two field extensions $\mathbb{Q}(\sqrt{d})$ and $\mathbb{Q}(\sqrt{-d})$.

Example

```
gap> K:=BianchiGcomplex(-43);;
gap> R:=FreeGResolution(K,11);;
gap> C:=TensorWithIntegers(R);;
gap> List([0..10],n->Homology(C,n));
[ [ 0 ], [ 0, 0 ], [ 2, 2, 12, 0 ], [ 2, 2, 24 ], [ 2, 2 ], [ 2 ],
  [ 2, 2, 12 ], [ 2, 2, 24 ], [ 2, 2 ], [ 2 ], [ 2, 2, 12 ] ]

gap> K:=BianchiGcomplex(-10);;
gap> R:=FreeGResolution(K,11);;
gap> List([0..10],k->Homology(TensorWithIntegers(R),k));
[ [ 0 ], [ 2, 2, 0, 0, 0 ], [ 2, 2, 2, 12, 0, 0 ], [ 2, 2, 2, 24 ],
  [ 2, 4, 12 ], [ 2, 2, 2, 6 ], [ 2, 2, 2, 12 ], [ 2, 2, 2, 24 ],
  [ 2, 4, 12 ], [ 2, 2, 2, 6 ], [ 2, 2, 2, 12 ] ]

gap> K:=BianchiGcomplex(-14);;
gap> R:=FreeGResolution(K,11);;
gap> List([0..10],k->Homology(TensorWithIntegers(R),k));
[ [ 0 ], [ 6, 0, 0, 0, 0, 0 ], [ 2, 2, 2, 4, 12, 0, 0, 0, 0 ],
  [ 2, 2, 2, 2, 24 ], [ 2, 2, 2, 4, 12 ], [ 2, 2, 2, 2, 24 ],
  [ 2, 2, 2, 4, 12 ], [ 2, 2, 2, 2, 24 ], [ 2, 2, 2, 4, 12 ],
  [ 2, 2, 2, 2, 24 ], [ 2, 2, 2, 4, 12 ] ]
```

The following commands count the number of orbits of cusps (in addition to the orbit of ∞). They determine that there is precisely one element in the ideal class group of $\mathscr{O}_{-43}$ (i.e it is a principal ideal domain) and that there are precisely two elements in the ideal class group of $\mathscr{O}_{-10}$ and precisely four elements in the ideal class group of $\mathscr{O}_{-14}$.

Example

```
gap> K:=BianchiGcomplex(-43);;
gap> List([1..K!.dimension(0)],k->Order(K!.stabilizer(0,k)));
[ 24, 24, 6, 6, 4, 4, 12, 12 ]

gap> K:=BianchiGcomplex(-10);;
gap> List([1..K!.dimension(0)],k->Order(K!.stabilizer(0,k)));
[ 6, 6, 4, 4, 6, infinity ]
```

```
gap> K:=BianchiGcomplex(-14);;
gap> List([1..K!.dimension(0)],k->Order(K!.stabilizer(0,k)));
[ 6, 6, 2, 2, 2, infinity, infinity, 2, infinity, 6, 4 ]
```

14.8 Some sanity checks

There is ample scope for bugs in the implementation of the above method for computing resolutions of Bianchi groups. The following sanity checks lend confidence to the implementation.

14.8.1 Equivariant Euler characteristic

Let X be any cell complex with an action of a group G such that (i) X has finitely many G-orbits of cells, and (ii) the stabilizer subgroup in G for each cell is either finite or free abelian. One defines the *equivariant Euler characteristic* $$\chi_G(X) = \sum_e (-1)^{\dim e} / |\mathrm{Stab}_G(e)|$$ where e ranges over a set of representatives of the orbits of those cells with finite stabilizers. If G has a finite index torsion free subgroup and if the complex X is contractible then one can define the Euler characteristic of the group to be $\chi(G) = \chi_G(X)$. It is known that $\chi(SL_n(\mathscr{O})) = \chi(GL_n(\mathscr{O})) = 0$ for $\mathscr{O}$ the ring of integers of a number field [DSGG+16].

One easy test to make in our computations is to check that the equivariant Euler characteristic of the 2-complex is indeed zero. The following commands perform this test for the group $SL_2(\mathscr{O}_{-23})$.

Example

```
gap> K:=BianchiGcomplex(-23);;

gap> chi:=0;;
gap> for n in [0..2] do
> for k in [1..K!.dimension(n)] do
> g:=Order(K!.stabilizer(n,k));
> if g < infinity then chi:=chi + (-1)^n/g; fi;
> od;od;
gap> chi;
0
```

14.8.2 Boundary squares to zero

The signs in the boundary maps of the free resolution are delicate. Another easy test is to check that the boundary in the resolution squares to zero. The following commands perform this check for the group $SL_2(\mathscr{O}_{-23})$.

Example

```
gap> K:=BianchiGcomplex(-23);;
gap> R:=FreeGResolution(K,10);;
gap> n:=2;;List([1..R!.dimension(n)],k->ResolutionBoundaryOfWord(R,n-1,R!.boundary(n,k)));
[ [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ] ]
gap> n:=3;;List([1..R!.dimension(n)],k->ResolutionBoundaryOfWord(R,n-1,R!.boundary(n,k)));
[ [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
```

```
    [  ], [  ], [  ] ]
gap> n:=4;;List([1..R!.dimension(n)],k->ResolutionBoundaryOfWord(R,n-1,R!.boundary(n,k)));
[ [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ] ]
gap> n:=5;;List([1..R!.dimension(n)],k->ResolutionBoundaryOfWord(R,n-1,R!.boundary(n,k)));
[ [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ],
  [  ], [  ], [  ] ]
```

14.8.3 Compare different algorithms or implementations

Sebastian Schoennenbeck in his thesis work computed some contractible 2-complexes on which Bianchi groups act with finite stabilizers (even when the ideal class is greater than 1) using a different approach to that of Swan. These computed complexes are stored in HAP and provide an alternative way of computing cohomology for the stored groups. Alexander Rahm in his thesis work implemented Swan's approach and has provided some 2-complexes that are also stored in HAP in cases where the ideal class is equal to 1.

The following commands test that Sebastian Schoennenbeck's 2-complex for $SL_2(\mathcal{O}_{-23})$ yields the same integral homology as the above HAP implementation. Both computations use HAP's implementation of Wall's perturbation technique for computing the resolution from the 2-complex.

Example

```
gap> K:=ContractibleGcomplex("SL(2,O-23)");;
gap> R:=FreeGResolution(K,10);;
gap> List([0..9],n->Homology(TensorWithIntegers(R),n));
[ [ 0 ], [ 12, 0, 0, 0 ], [ 2, 2, 12, 0, 0 ], [ 2, 2, 12 ], [ 2, 2, 12 ],
  [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ] ]

gap> K:=BianchiGcomplex(-23);;
gap> R:=FreeGResolution(K,10);;
gap> List([0..9],n->Homology(TensorWithIntegers(R),n));
[ [ 0 ], [ 12, 0, 0, 0 ], [ 2, 2, 12, 0, 0 ], [ 2, 2, 12 ], [ 2, 2, 12 ],
  [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ], [ 2, 2, 12 ] ]
```

14.8.4 Compare geometry to algebra

The number of cusps (i.e. the number of orbits of vertices with infinite stabilizer subgroup) must be precisely one less than the number of elements in the ideal class group of $\mathcal{O}_{-d}$. The following commands check this for $SL_2(\mathcal{O}_{-23})$ where $\mathcal{O}_{-23}$ is known to have class number 3. (This class number is easily computed from a formula in Swan's paper.)

Example

```
gap> K:=BianchiGcomplex(-23);;
gap> List([1..K!.dimension(0)],k->Order(K!.stabilizer(0,k)));
[ 6, 2, 2, 4, infinity, infinity ]
```

A visualization of the fundamental domain tells us a certain amount about the algebra. In the case of $SL_2(\mathcal{O}_{-23})$

a fundamental domain for the action on $\mathbb{C}$ by the translation subgroup generated by the matrices $$ \left(\begin{array}{ll} 1 &1\\ 0 &1\end{array}\right), \left(\begin{array}{ll} 1 &\omega\\ 0 &1\end{array}\right) $$ $\omega = (1+\sqrt{-23})/2$ is indicated by the white rectangle. From this we see that under the action of $SL_2(\mathcal{O}_{-23})$ there are at most 11 orbits of 2-cells, the central decagon and ten quadrilaterals. However, the matrix $$ \left(\begin{array}{rr} 0 &-1\\ 1 &0\end{array}\right) $$ maps $(z,0)$ to $(-1/z,0)$ and fixes $(0,1)$. This isometry identifies points on the boundary of the decagon pairwise. These observations are consistent with the above listing of the six orbit stabilizers and the following algebraic information on the boundaries of the 2-cells in the Bianchi 2-complex.

Example

```
gap> K:=BianchiGcomplex(-23);;
gap> List([1..K!.dimension(2)],k->Length(K!.boundary(2,k)));
[ 10, 4, 4, 4, 4, 4 ]
```

14.9 Group presentations

Swan's reason for studying fundamental domains was to obtain explicit group presentations for $SL_2(\mathcal{O}_{-d})$ for various values of d. The following commands obtain a presentation for $SL_2(\mathcal{O}_{-23})$.

Example

```
gap> K:=BianchiGcomplex(-23);;
gap> R:=FreeGResolution(K,2);;
gap> P:=PresentationOfResolution(R);
gap> G:=SimplifiedFpGroup(P!.freeGroup/P!.relators);
<fp group on the generators [ k, r, s, w, x ]>
gap> RelatorsOfFpGroup(G);
[ w^-1*k*w*k^-1, s^-1*r*s*r^-1, k^6, x^-1*k^-3*x*k^-3, s^-1*k^-3*s*k^-3,
  r^-1*w*x^-1*s*r*w^-1*x*s^-1, r^-1*k^-3*r*k^-3,
  x*k^-2*r^-1*x*r^-1*s^-1*k^-1*s^-1, x^-1*k^3*s*r*x^-1*s*r ]

gap> #Next we identify the generators as matrices
gap> GeneratorsOfGroup(P!.freeGroup);
[ k, m, n, p, q, r, s, t, u, v, w, x, y, z ]
gap> P!.gens;
[ 19, 6, 6, 20, 6, 21, 22, 6, 52, 53, 2, 50, 1, 4 ]

gap> k:=R!.elts[19];
[ [ 1, 1 ],
  [ -1, 0 ] ]
gap> r:=R!.elts[21];
[ [ 3, 3 + -1 Sqrt(-23) ],
  [ -3/2 + -1/2 Sqrt(-23), -5 ] ]
gap> s:=R!.elts[22];
[ [ 2 + 1 Sqrt(-23), 13/2 + 1/2 Sqrt(-23) ],
  [ 5/2 + -1/2 Sqrt(-23), -1 Sqrt(-23) ] ]
gap> w:=R!.elts[2];
[ [ 3/2 + 1/2 Sqrt(-23), -3/2 + 1/2 Sqrt(-23) ],
  [ 3/2 + -1/2 Sqrt(-23), 3 ] ]
gap> x:=R!.elts[50];
[ [ 11/2 + 1/2 Sqrt(-23), 15/2 + -1/2 Sqrt(-23) ],
```

```
    [ -1 Sqrt(-23), -4 + -1 Sqrt(-23) ] ]
```

14.9.1 Swan's generators

The five generators in the above presentation for $SL_2(\mathscr{O}_{-23})$ are obtained by constructing the low dimensions of a CW-classifying space, forming a maximal tree in its 1-skeleton, constructing a presentation whose generators correspond to edges not in the tree, and then applying Tietze operations to simplify this presentation. Swan's approach instead uses generators corresponding to the "face pairings" of the 3-dimensional fundamental domain. For $d \neq 1,3$ the number of these face-pairing generators in $SL_2(\mathscr{O}_{-d})$ is equal to 3 plus the number of orbits of 2-cells in the equivariant 2-complex K. The three extra generators are the two translations corresponding to the vertical faces of the 3-dimensional fundamental domain which get homotopied to edges in K, plus the element $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ which acts trivially on $\mathfrak{H}^3$. For $d = 1,3$ there is one further generator due to the existence of non-trivial units in $\mathscr{O}(\mathbb{Q}(\sqrt{-1}))$ and $\mathscr{O}(\mathbb{Q}(\sqrt{-3}))$.

The following commands show that for $SL_2(\mathscr{O}_{-23})$ there are 6 orbits of 2-cells in K and then lists Swan's 9 generators.

Example

```
gap> K:=BianchiGcomplex(-23);;
gap> K!.dimension(2);
6
gap> for g in GeneratorsOfGroup(K!.group) do
> Print(g,"\n");
> od;
[ [ -1, 0 ], [ 0, -1 ] ]
[ [ 1, 1 ], [ 0, 1 ] ]
[ [ 1, 1/2 + 1/2 Sqrt(-23) ], [ 0, 1 ] ]
[ [ -1, 0 ], [ 1, -1 ] ]
[ [ 5 + 2 Sqrt(-23), -11 ], [ 3/2 + -1/2 Sqrt(-23), 3/2 + 1/2 Sqrt(-23) ] ]
[ [ 208 + 25 Sqrt(-23), -225 + 25 Sqrt(-23) ], [ 3/2 + 1/2 Sqrt(-23), -3 ] ]
[ [ 18 + 91 Sqrt(-23), -437 ], [ 5/2 + 1/2 Sqrt(-23), -5/2 + 1/2 Sqrt(-23) ] ]
[ [ 1 + -6 Sqrt(-23), -29 ], [ 5/2 + -1/2 Sqrt(-23), -5/2 + -1/2 Sqrt(-23) ] ]
[ [ 1 Sqrt(-23), -6 ], [ 4, 1 Sqrt(-23) ] ]
```

14.10 Finite index subgroups

The following commands compute the integral homology of a congruence subgroup G of index 24 in $SL_2(\mathscr{O}_{-23})$. They also compute a presentation for G with 13 generators and 24 relators.

Example

```
gap> OQ:=RingOfIntegers(QuadraticNumberField(-23));;
gap> I:=QuadraticIdeal(OQ,[Sqrt(-23)]);
ideal of norm 23 in O(Q(Sqrt(-23)))
gap> G:=HAP_CongruenceSubgroupGamma0(I);
<group of 2x2 matrices in characteristic 0>
gap> IndexInSL2O(G);
24
```

```
gap> K:=BianchiGcomplex(-23);;
gap> R:=FreeGResolution(K,11);;
gap> R:=QuadraticToCyclotomicCoefficients(R);;
gap> S:=ResolutionFiniteSubgroup(R,G);;
gap> List([0..10],n->Homology(TensorWithIntegers(S),n));
[ [ 0 ], [ 2, 0, 0, 0, 0, 0, 0, 0, 0 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0 ], [ 2, 2, 2, 2, 2, 2, 2, 2 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2 ], [ 2, 2, 2, 2, 2, 2, 2, 2 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2 ], [ 2, 2, 2, 2, 2, 2, 2, 2 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2 ], [ 2, 2, 2, 2, 2, 2, 2, 2 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2 ] ]

gap> P:=PresentationOfResolution(S);;
gap> H:=SimplifiedFpGroup(P!.freeGroup/P!.relators);
<fp group on the generators [ f8, f10, f15, f70, f86, f125, f132, f138, f182,
  f187, f191, f273, f279 ]>
gap> Length(RelatorsOfFpGroup(H));
24
```

14.11 Totally real quadratic fields / Hilbert modular group

For square free $d > 0$ the group $PSL_2(\mathcal{O}_d)$ is a subgroup of $PSL_2(\mathbb{R})$ but not a discrete subgroup. It embeds as a discrete subgroup of $PSL_2(\mathbb{R}) \times PSL_2(\mathbb{R})$ but a fundamental domain for its action on $\mathfrak{h}^2 \times \mathfrak{h}^2$ is not implemented in HAP. One can obtain a contractible $GL_2(\mathcal{O}_d)$-equivariant complex using Sebastian Schoennenbeck's Magma implementation of Voronoi's algorithm to produce text files that can be read into HAP. A few of these text files are stored in HAP. The following commands use one such text file to compute integral homology groups of $GL_2(\mathcal{O}(\mathbb{Q}(\sqrt{14})))$. In this example the contractible complex is 4-dimensional and has 33 orbits of 4-dimensional cells.

Example

```
gap> K:=ContractibleGcomplex("GL(2,O14)");
Non-free resolution in characteristic 0 for matrix group .
No contracting homotopy available.

gap> List([0..5],K!.dimension);
[ 144, 463, 532, 234, 33, 0 ]

gap> R:=FreeGResolution(K,6);;
gap> List([0..5],n->Homology(TensorWithIntegers(R),n));
[ [ 0 ], [ 2, 2, 2, 0 ], [ 2, 2, 2, 2, 2, 2, 0, 0, 0, 0 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 24, 24, 0, 0, 0, 0 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 12, 24 ],
  [ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4 ] ]
```

14.12 Calling Magma's Voronoi algorithm directly

If

- the MAGMA computational algebra system can be started using the line command `magma` and
- Sebastian Schoennenbeck's Magma implementation of Voronoi's algorithm for imaginary quadratic number fields is copied to ~pkg/Voronoi/VMH-ImaginaryQuadraticNumberFields and
- Sebastian Schoennenbeck's Magma implementation of Voronoi's algorithm for totally real quadratic number fields is copied to ~pkg/Voronoi/VMH-TotallyRealNumberFields

then one can use the GAP commands `VoronoiGL(n,d)` and `VoronoiSL(n,d)` to call the MAGMA implementation of the Voronoi algorithm directly. For example, the following commands do this to calculate the homology groups $H^4(GL_2(\mathcal{O}_{-5}), P_{\mathcal{O}_{-5}}(8)) = (\mathbb{Z}_2)^{17} \oplus (\mathbb{Z}_4)^2 \oplus (\mathbb{Z}_8)^2$ and $H_4(GL_2(\mathcal{O}(\mathbb{Q}(\sqrt{5})), \mathbb{Z}) = (\mathbb{Z}_2)^7 \oplus \mathbb{Z}_{60}$ and $H_n(GL_3(\mathcal{O}_{-5}), \mathbb{Z})$ for $n = 1, 2$.

Example

```
gap> V:=VoronoiComplexSL(2,-5);; #makes a call to Magma
gap> R:=FreeGResolution(V,5);;
gap> M:=HomogeneousPolynomials(R!.group,8);;
gap> C:=HomToIntegralModule(R,M);;
gap> Cohomology(C,4);
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 8, 8 ]

gap> V:=VoronoiComplexGL(2,5);; #makes a call to Magma
gap> R:=FreeGResolution(V,5);;
gap> Homology(TensorWithIntegers(R),4);
[ 2, 2, 2, 2, 2, 2, 60 ]

gap> #V:=VoronoiComplexGL(3,-5);; #makes a call to Magma (requires much time)
gap> V:=ContractibleGcomplex("SL(3,O-5)");; #loads stored data (is faster)
gap> R:=FreeGResolution(V,4);;
gap> Homology(TensorWithIntegers(R),1);
[ 2 ]
gap> Homology(TensorWithIntegers(R),2);
[ 2, 2, 2, 4 ]
```

Chapter 15

Parallel computation

15.1 An embarassingly parallel computation

The following example creates fifteen child processes and uses them simultaneously to compute the second integral homology of each of the 2328 groups of order 128. The final command shows that

$H_2(G,\mathbb{Z}) = \mathbb{Z}_2^{21}$

for the 2328-th group G in GAP's library of small groups. The penulimate command shows that the parallel computation achieves a speedup of 10.4 .

Example

```
gap> Processes:=List([1..15],i->ChildProcess());;
gap> fn:=function(i);return GroupHomology(SmallGroup(128,i),2);end;;
gap> for p in Processes do
> ChildPut(fn,"fn",p);
> od;
gap> Exec("date +%s");L:=ParallelList([1..2328],"fn",Processes);;Exec("date +%s");
1716105545
1716105554
gap> Exec("date +%s");L1:=List([1..2328],fn);;Exec("date +%s");
1716105586
1716105680

gap> speedup:=1.0*(680-586)/(554-545);
10.4444

gap> L[2328];
[ 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 ]
```

The function `ParallelList()` is built from HAP's six core functions for parallel computation.

15.2 A non-embarassingly parallel computation

The following commands use core functions to compute the product $A = M \times N$ of two random matrices by distributing the work over two processors.

Example

```
gap> M:=RandomMat(10000,10000);;
gap> N:=RandomMat(10000,10000);;
```

```
gap>
gap> s:=ChildProcess();;
gap>
gap> Exec("date +%s");
1716109418
gap> Mtop:=M{[1..5000]};;
gap> Mbottom:=M{[5001..10000]};;
gap> ChildPut(Mtop,"Mtop",s);
gap> ChildPut(N,"N",s);
gap> NextAvailableChild([s]);;
gap> ChildCommand("Atop:=Mtop*N;;",s);;
gap> Abottom:=Mbottom*N;;
gap> A:=ChildGet("Atop",s);;
gap> Append(A,Abottom);;
gap> Exec("date +%s");
1716110143

gap> AA:=M*N;;Exec("date +%s");
1716111389

gap> speedup:=1.0*(111389-110143)/(110143-109418);
1.71862
```

The next commands compute the product $A = M \times N$ of two random matrices by distributing the work over fifteen processors. The parallelization is very naive (the entire matrices M and N are communicated to all processes) and the computation achieves a speedup of 7.6.

Example

```
gap> M:=RandomMat(15000,15000);;
gap> N:=RandomMat(15000,15000);;
gap> S:=List([1..15],i->ChildCreate());;

gap> Exec("date +%s");
1716156583
gap> ChildPutObj(M,"M",S);
gap> ChildPutObj(N,"N",S);
gap> for i in [1..15] do
> cmd:=Concatenation("A:=M{[1..1000]+(",String(i),"-1)*1000}*N;");
> ChildCommand(cmd,S[i]);
> od;
gap> A:=[];;
gap> for i in [1..15] do
>  C:=ChildGet("A",S[i]);
>  Append(A,C);
> od;
gap> Exec("date +%s");
1716157489

gap> AA:=M*N;;Exec("date +%s");
1716164405

gap> speedup:=1.0*(64405-57489)/(57489-56583);
```

```
7.63355
```

15.3 Parallel persistent homology

Section 5.8 illustrates an alternative method of computing the persitent Betti numbers of a filtered pure cubical complex. The method lends itself to parallelisation. However, the following parallel computation of persistent Betti numbers achieves only a speedup of 1.5 due to a significant time spent transferring data structures between processes. On the other hand, the persistent Betti function could be used to distribute computations over several computers. This might be useful for larger computations that require significant memory resources.

Example

```
gap> file:=HapFile("data247.txt");;
gap> Read(file);;
gap> F:=ThickeningFiltration(T,25);;
gap> S:=List([1..15],i->ChildCreate());;
gap> N:=[0,1,2];;
gap> Exec("date +%s");P:=ParallelPersistentBettiNumbers(F,N,S);;Exec("date +%s");
1717160785
1717161285

gap> Exec("date +%s");Q:=PersistentBettiNumbersAlt(F,N);;Exec("date +%s");
1717161528
1717162276
gap> speedup:=1.0*(1717162276-1717161528)/(1717161285-1717160785);
1.496
```

15.4 Parallelism not a substitute for careful serial implementation

Here is a story of naive implementation design causing a serial algorithm to run at least 170 times slower than it should!

The following snippet begins with the construction of a regular CW-complex Y with 157539 cells. It then constructs the cellular chain complex $C_* = C_*(Y)$. Computing the homology of Y directly from C_* would be time consuming due to the large number of cells in Y. So, an elementary procedure is used to eliminate redundant generators from Y in order to obtain a smaller quasi-isomorphic chain complex D_* and this smaller complex is used to quickly obtain the homology. However, the construction of D_* takes over 170 seconds.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)s");;
gap> G:=HAPCongruenceSubgroupGamma0(3,101);;
gap> Y:=GComplexToRegularCWComplex(K,G);;

gap> C:=ChainComplexOfRegularCWComplex(Y);
Chain complex of length 3 in characteristic 0 .

gap> List([0..3],C!.dimension);
[ 3679, 37565, 75083, 41212 ]
```

```
gap> D:=ContractedComplex(C);time;
Chain complex of length 3 in characteristic 0 .

170547

gap> List([0..3],D!.dimension);
[ 1, 27, 78, 67 ]

gap> List([0..3],n->Homology(D,n));time;
[ [ 0 ], [ 25 ], [ 2, 2, 2, 2, 2 ],
  [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ]
26
```

The above procedure for reducing the number of generators in a chain complex of free abelian groups uses the same elementary technique for "generator removal" as a slightly more elaborate implementation programmed some years earlier. The earlier implementation is (unnecessarily) restricted to chain complexes arising as cellular chain complexes of regular CW-complexes while the later implemetation is aimed at arbitrary chain complexes of free abelian groups. The earlier implementation returns chain homotopy equivalences between the original and the simplified complexes. To ease the the coding of the later implementation it was decided to output just the simplified chain complex and forget about the chain maps yielding the chain equivalence.

The following snippet illustrates the earlier implementation. It again begins with the construction of the regular CW-complex Y with 157539 cells. It then takes just over a second to construct a pair of chain equivalences $f_1 : C_*(Y) \to C_*(Y')$ and $f_2 : C_*(Y') \to C_*(Y)$ with Y' a homotopy equivalent non-regular CW-complex with fewer cells. It uses $D_* = C_*(Y')$ to quickly compute the homology of Y.

Example

```
gap> K:=ContractibleGcomplex("SL(3,Z)s");;
gap> G:=HAPCongruenceSubgroupGamma0(3,101);;
gap> Y:=GComplexToRegularCWComplex(K,G);;
gap> f:=ChainComplexEquivalenceOfRegularCWComplex(Y);time;
[ Chain Map between complexes of length 3 .
    , Chain Map between complexes of length 3 .
    ]
1192

gap> C:=Source(f[1]);
Chain complex of length 3 in characteristic 0 .

gap> List([0..3],C!.dimension);
[ 3679, 37565, 75083, 41212 ]

gap> D:=Target(f[1]);
Chain complex of length 3 in characteristic 0 .

gap> List([0..3],D!.dimension);
[ 4, 15, 29, 33 ]

gap> List([0..3],n->Homology(D,n));time;
```

```
[ [ 0 ], [ 25 ], [ 2, 2, 2, 2, 2 ],
  [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ]
211
```

The takeaway from this story is that careful design of the datatypes to be used in the implementation of an algorithm can significantly impact the speed of the implementation! This is a well established principle, illustrated forcefully by the story. The decision to output chain maps $C_* \leftrightarrow D_*$ leads to an implementation where all redundant generators in C_* first have to be identified, and enough information has to be retained to produce the chain maps. The later decision to output just the complex D_* leads to an easier to implement process of eliminating each redundant generator in a boundary matrix as soon at it is found. This is analogous to a gaussian elimination procedure for producing echelon form that updates a boundary matrix each time a suitable pivot is found; it is easy to code but less efficient than the earlier implementation.

A new implementation of `ContractedComplex(C)` is on the HAP to-do list.

Chapter 16

Regular CW-structure on knots (written by Kelvin Killeen)

16.1 Knot complements in the 3-ball

While methods for endowing knot complements with CW-structure already exist in HAP (see section 2.1), they often result in a large number of cells which can make computing with them taxing. The following example shows how one can obtain a comparatively small 3-dimensional regular CW-complex corresponding to the complement of a thickened trefoil knot from an arc presentation. Recall that an arc presentation is encoded in HAP as a list of integer pairs corresponding to the position of the endpoints of each horizontal arc in a grid.

Example

```
gap> k_:=PureCubicalKnot(3,1);
prime knot 1 with 3 crossings

gap> arc:=ArcPresentation(k_);
[ [ 2, 5 ], [ 1, 3 ], [ 2, 4 ], [ 3, 5 ], [ 1, 4 ] ]
gap> k_:=RegularCWComplex(PureComplexComplement(k_));
Regular CW-complex of dimension 3

gap> Size(k_);
13291
gap> k:=KnotComplement(arc);
Regular CW-complex of dimension 3

gap> Size(k);
395
```

An optional argument of "rand" in the KnotComplement function randomises the order in which 2-cells are added to the complex. This allows for alternate presentations of the knot group.

Example

```
gap> arc:=ArcPresentation(PureCubicalKnot(3,1));
[ [ 2, 5 ], [ 1, 3 ], [ 2, 4 ], [ 3, 5 ], [ 1, 4 ] ]
gap> k:=KnotComplement(arc,"rand");
Random 2-cell selection is enabled.
Regular CW-complex of dimension 3
```

```
gap> g:=FundamentalGroup(k); RelatorsOfFpGroup(g);
#I  there are 2 generators and 1 relator of total length 6
<fp group of size infinity on the generators [ f1, f2 ]>
[ f2^-1*f1*f2^-1*f1^-1*f2*f1^-1 ]
gap> k:=KnotComplement(arc,"rand");
Random 2-cell selection is enabled.
Regular CW-complex of dimension 3

gap> g:=FundamentalGroup(k); RelatorsOfFpGroup(g);
#I  there are 2 generators and 1 relator of total length 7
<fp group of size infinity on the generators [ f1, f2 ]>
[ f1*f2^-2*f1*f2*f1^-1*f2 ]
```

It is often useful to obtain an inclusion of regular CW-complexes $\iota : \partial(N(K)) \hookrightarrow B^3 \backslash N(K)$ from the boundary of a tubular neighbourhood of some knot $N(K)$ into its complement in the 3-ball $B^3 \backslash N(K)$. The below example does this for the first prime knot on 11 crossings.

Example

```
gap> arc:=ArcPresentation(PureCubicalKnot(11,1));
[ [ 2, 9 ], [ 1, 3 ], [ 2, 6 ], [ 4, 7 ], [ 3, 5 ], [ 6, 10 ], [ 4, 8 ],
  [ 9, 11 ], [ 7, 10 ], [ 1, 8 ], [ 5, 11 ] ]
gap> k:=KnotComplementWithBoundary(arc);
Map of regular CW-complexes

gap> Size(Source(i));
616
gap> Size(Target(i));
1043
```

Note that we can add n-cells to regular CW-complexes by specifying the $(n-1)$-cells in their boundaries and $(n+1)$-cells in their coboundaries.

Example

```
gap> k:=KnotComplement([[1,2],[1,2]])!.boundaries;;
gap> Homology(RegularCWComplex(k),0);
[ 0 ]
gap> AddCell(k,0,[0],[]);
gap> Homology(RegularCWComplex(k),0);
[ 0, 0 ]
```

16.2 Tubular neighbourhoods

Let Y denote a CW-subcomplex of a regular CW-complex X and let $N(Y)$ denote an open tubular neighbourhood of Y. Given an inclusion of regular CW-complexes $f : Y \hookrightarrow X$, this algorithm describes a procedure for obtaining the associated inclusion $f' : \partial C \hookrightarrow C$ where $C = X \backslash N(Y)$ and ∂C denotes the boundary of C. The following is also assumed:

Let e^n denote a cell of $X \backslash Y$ of dimension n with $\bar{e}^n$ denoting its closure. For each n-cell, all of the connected components of the subcomplex $\bar{e}^n \cap Y$ are contractible.

Some additional terminology and notation is needed to describe this algorithm. The output regular CW-complex $X\backslash N(Y)$ consists of the cell complex $X\backslash Y$ as well as some additional cells to maintain regularity. A cell of $X\backslash N(Y)$ is referred to as *internal* if it lies in $X\backslash Y$, it is *external* otherwise. Let $\bar{e}^n$ denote the closure in X of an internal cell e^n. Note that $\bar{e}^n$ is a CW-subcomplex of X and so is the intersection $\bar{e}^n \cap Y$ which can be expressed as the union

$\bar{e}^n \cap Y = A_1 \cup A_2 \cup \cdots \cup A_k$

of its path components A_i all of which are CW-subcomplexes of Y. For each n-cell of $X\backslash Y$ there is one internal n-cell e^n of $X\backslash N(Y)$. For $n \geq 1$ there is also one external $(n-1)$-cell $f^{e^n}_{A_i}$ for each path component A_i of $\bar{e}^n \cap Y$. Lastly, we need a method for determining the homological boundary of the internal and external cells:

• The boundary of an internal n-cell e^n consists of all those internal $(n-1)$-cells of $\bar{e}^n$ together with all external $(n-1)$-cells $f^{e^n}_{A_i}$ where A_i is a path component of $\bar{e}^n \cap Y$.

• The boundary of an external $(n-1)$-cell $f^{e^n}_{A_i}$ consists of all those external $(n-2)$-cells $f^{e^{n-1}}_{B_j}$ where e^{n-1} is an $(n-1)$-cell of $\bar{e}^n$ and $B_j \subseteq A_i$ is a path component of A_i.

The following three steps comprise the algorithm.

(1) For each internal n-cell $e^n \subset X\backslash Y$, compute the CW-complex $\bar{e}^n \cap Y$ as a union of path components $A_1 \cup A_2 \cup \cdots \cup A_k$. This information can be used to determine the number of cells of $X\backslash N(Y)$ in each dimension.

(2) Create a list $B = [\,[\,],[\,],\ldots,[\,]\,]$ of length $\dim X + 1$.

(3) For $0 \leq n \leq \dim X$ set $B[n+1] = [b_1, b_2, \ldots, b_{\alpha_n}]$ where α_n is the number of n-cells in $X\backslash N(Y)$ and b_i is a list of integers describing the $(n-1)$-cells of the i^{th} n-cell of $X\backslash N(Y)$. The internal cells will always be listed before the external cells in each sublist. Return B as a regular CW-complex.

The following example computes the tubular neighbourhood of a 1-dimensional subcomplex of a 3-dimensional complex corresponding to the Hopf link embedded in the closed 3-ball.

Example

```
gap> arc:=[[2,4],[1,3],[2,4],[1,3]];;
gap> f:=ArcPresentationToKnottedOneComplex(arc);
Map of regular CW-complexes

gap> comp:=RegularCWComplexComplement(f);
Testing contractibility...
151 out of 151 cells tested.
The input is compatible with this algorithm.
Regular CW-complex of dimension 3
```

Note that the output of this algorithm is just a regular CW-complex, not an inclusion map. The function `BoundaryMap` can be employed to obtain the boundary of a pure complex. This results in three path components for this example: two corresponding to the boundary of the knotted tori and the other corresponding to the boundary of the 3-ball in which the link was embedded. These path components can be obtained as individual CW-subcomplexes if desired. A CW-subcomplex is represented in HAP as a list $[X, s]$ where X is a regular CW-complex and s is a list of length n whose i^{th} entry lists the indexing of each $(i-1)$-cell of the n-dimensional subcomplex of X. CW-subcomplexes and CW maps can be converted between each other interchangeably. This next example obtains the inclusion detailed in the above algorithm, finds the path components of the source of said inclusion, shows that they are in fact disjoint, and then obtains the first four integral homology groups of each component.

Example

```
gap> f_:=BoundaryMap(comp);
Map of regular CW-complexes

gap> f_:=RegularCWMapToCWSubcomplex(f_);;
gap> paths:=PathComponentsCWSubcomplex(f_);
[ [ Regular CW-complex of dimension 3
        ,
      [ [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 18, 19, 20 ],
          [ 1, 2, 3, 4, 5, 6, 13, 14, 15, 16, 17, 18, 33, 34, 35, 46, 47, 48
            ], [ 11, 12, 13, 14, 15, 16, 35, 36 ] ] ],
  [ Regular CW-complex of dimension 3
      , [ [ 21, 24, 25, 27, 30, 31, 32, 37, 38, 39, 40, 43, 45, 46, 48 ],
        [ 49, 51, 53, 56, 57, 59, 61, 63, 65, 67, 69, 71, 73, 74, 76, 79,
            82, 83, 86, 87, 90, 91 ], [ 37, 39, 41, 44, 45, 47, 49 ] ] ],
  [ Regular CW-complex of dimension 3
      , [ [ 22, 23, 26, 28, 29, 33, 34, 35, 36, 41, 42, 44, 47, 49, 50 ],
        [ 50, 52, 54, 55, 58, 60, 62, 64, 66, 68, 70, 72, 75, 77, 78, 80,
            81, 84, 85, 88, 89, 92 ], [ 38, 40, 42, 43, 46, 48, 50 ] ] ] ]
gap> paths:=List(paths,CWSubcomplexToRegularCWMap);
[ Map of regular CW-complexes
    , Map of regular CW-complexes
    , Map of regular CW-complexes
    ]
gap> List([1..3],x->List(Difference([1..3],[x]),y->IntersectionCWSubcomplex(paths[x],paths[y])));
[ [ [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ], [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ] ], [ [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ], [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ] ], [ [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ], [ Regular CW-complex of dimension 3
            , [ [  ], [  ], [  ] ] ] ] ]

gap> List(paths,x->List([0..3],y->Homology(Source(x),y)));
[ [ [ 0 ], [  ], [ 0 ], [  ] ], [ [ 0 ], [ 0, 0 ], [ 0 ], [  ] ],
  [ [ 0 ], [ 0, 0 ], [ 0 ], [  ] ] ]
```

As previously mentioned, for the tubular neighbourhood algorithm to work, we require that no external cells yield non-contractible path-components in their intersection with the subcomplex. If this is ever the case then we can subdivide the offending cell to prevent this from happening. We have implemented two subdivision algorithms in HAP, one for barycentrically subdividing a given cell, and the other for subdividing an n-cell into as many n-cells as there are $(n-1)$-cells in its boundary. Barycentric subdivision is integrated into the `RegularCWComplexComplement` function and will be performed automatically as required. The following example shows this automatic subdivision running via the complement of a tubular neighbourhood of the unknot, then obtains an inclusion map from the closure of an arbitrary 3-cell of this complex and then compares the difference in size of the two different subdivisions of a 2-cell in the boundary of this 3-cell.

Example

```
gap> arc:=[[1,2],[1,2]];;
gap> unknot:=ArcPresentationToKnottedOneComplex(arc);
```

```
Map of regular CW-complexes

gap> f:=RegularCWComplexComplement(unknot);
Testing contractibility...
79 out of 79 cells tested.
Subdividing 3 cell(s):
100% complete.
Testing contractibility...
145 out of 145 cells tested.
The input is compatible with this algorithm.
Regular CW-complex of dimension 3

gap> f:=Objectify(HapRegularCWMap,rec(source:=f,target:=f,mapping:={i,j}->j));
Map of regular CW-complexes

gap> closure:=ClosureCWCell(Target(f),3,1);
[ Regular CW-complex of dimension 3
    ,
  [ [ 1, 2, 3, 4, 7, 8, 9, 10, 11, 13, 14, 20, 21, 22, 23, 25 ],
      [ 1, 2, 3, 7, 8, 9, 10, 11, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 28, 55, 58, 59,
          60, 63 ], [ 1, 4, 7, 8, 9, 13, 14, 15, 18, 52 ], [ 1 ] ] ]
gap> Size(Target(f));
195
gap> Size(Target(BarycentricallySubdivideCell(f,2,1)));
231
gap> Size(Target(SubdivideCell(f,2,1)));
207
```

16.3 Knotted surface complements in the 4-ball

A construction of Satoh's, the tube map, associates a ribbon torus-knot to virtual knot diagrams. A virtual knot diagram differs from a knot diagram in that it allows for a third type of crossing, a virtual crossing. The image of such a crossing via the tube map is two tori which pass through each other. An arc diagram is a triple of lists `[arc,cross,cols]` that encode virtual knot diagrams. `arc` is an arc presentation. `cross` is a list of length the number of crossings in the knot associated to the arc presentation whose entries are $-1, 0$ or 1 corresponding to an undercrossing (horizontal arc underneath vertical arc), a virtual crossing (depicted by intersecting horizontal and vertical arcs) and an overcrossing (horizontal arc above vertical arc) respectively. `cols` is a list of length the number of 0 entries in `cross` and its entries are $1, 2, 3$ or 4. It describes the types of 'colourings' we assign to the virtual crossings. We interpret each integer as the change in 4-dimensional height information as represented by a colour scale from blue (lower down in 4-space), to green (0 level), to red (higher up in 4-space). Without loss of generality, we impose that at each virtual crossing, the vertical arc passes through the horizontal arc. Thus, 1 corresponds to the vertical bar entering the horizontal bar as blue and leaving as blue, 2 corresponds to entering as blue and leaving as red, 3 corresponds to entering as red and leaving as blue and 4 corresponds to entering and leaving as red. A coloured arc diagram can be visualised using the `ViewColouredArcDiagram` function.

Example

```
gap> arc:=ArcPresentation(PureCubicalKnot(6,1));
[ [ 5, 8 ], [ 4, 6 ], [ 3, 5 ], [ 2, 4 ], [ 1, 3 ], [ 2, 7 ], [ 6, 8 ], [ 1, 7 ] ]
gap> cross:=[0,0,1,-1,-1,0];;
gap> cols:=[1,4,3];;
gap> ViewArc2Presentation([arc,cross,cols]);
convert-im6.q16: pixels are not authentic '/tmp/HAPtmpImage.txt' @ error/cache.c/QueueAuthenticPi
```

Towards obtaining a regular CW-decomposition of ribbon torus-knots, we first begin by embedding a self-intersecting knotted torus in the 3-ball. The function `ArcDiagramToTubularSurface` inputs a coloured arc diagram and outputs an inclusion from the boundary of some (potentially self-intersecting) torus in the 3-ball. By inputting just an arc presentation, one can obtain an inclusion identical to the `KnotComplementWithBoundary` function. By additionally inputting a list of -1s and 1s, one can obtain an inclusion similar to `KnotComplementWithBoundary` but where there is extra freedom in determining whether or not a given crossing is an under/overcrossing. If one inputs both of the above but includes 0 entries in the `cross` list and includes the list of colours, the output is then an inclusion from an embedded self-intersecting torus into the 3-ball where each 2-cell (the top-dimensional cells of the self-intersecting surface) is assigned a colour.

Example

```
gap> tub:=ArcDiagramToTubularSurface(arc);
Map of regular CW-complexes

gap> tub:=ArcDiagramToTubularSurface([arc,cross]);
Map of regular CW-complexes

gap> tub:=ArcDiagramToTubularSurface([arc,cross,cols]);
Map of regular CW-complexes

gap> List([1..Length(Source(tub)!.boundaries[3])],x->tub!.colour(2,tub!.mapping(2,x)));
[ [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ],
  [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ],
  [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ],
  [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ],
  [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ -1 ], [ -1 ],
  [ 0 ], [ 0 ], [ -1 ], [ -1 ], [ -1 ], [ -1 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 1 ], [ 1 ],
  [ 0 ], [ 0 ], [ 1 ], [ 1 ], [ 1 ], [ 1 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 1 ], [ 1 ], [ 0 ],
  [ 0 ], [ -1 ], [ -1 ], [ 1 ], [ -1 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ -1 ], [ -1 ], [ 0 ],
  [ 1 ], [ 1 ], [ 0 ], [ 0 ], [ 0 ], [ 0 ], [ 1 ], [ -1 ], [ 0 ] ]
```

From this self-intersecting surface with colour, we can lift it to a surface without self-intersections in $\mathbb{R}^4$. We do this by constructing a regular CW-complex of the direct product $B^3 \times [a,b]$ where B^3 denotes the 3-ball, a is 1 less than the smallest integer assigned to a cell by the colouring, and b is 1 greater than the largest integer assigned to a cell by the colouring. The subcomplex of the direct product corresponding to the surface without intersection can be obtained using the colouring with additional care taken to not lift any 1-cells arising as double-point singularities. The following example constructs the complement of a ribbon torus-link embedded in $\mathbb{R}^4$ obtained from the Hopf link with one virtual crossing and then calculates some invariants of the resulting space. We compare

the size of this complex, as well as how long it takes to obtain the same invariants, with a cubical complex of the same space. As barycentric subdivision can massively increase the size of the cell complex, the below method sequentially obtains the tubular neighbourhood of the entire subcomplex by obtaining the tubular neighbourhood of each individual 2-cell. This has yet to be optimised so it currently takes some time to complete.

Example

```
gap> arc:=[[2,4],[1,3],[2,4],[1,3]];;
gap> tub:=ArcDiagramToTubularSurface([arc,[0,-1],[2]]);
Map of regular CW-complexes

gap> tub:=LiftColouredSurface(tub);
Map of regular CW-complexes

gap> Dimension(Source(tub));
2
gap> Dimension(Source(tub));
4
gap> map:=RegularCWMapToCWSubcomplex(tub);;
gap> sub:=SortedList(map[2][3]);;
gap> sub:=List(sub,x->x-(Position(sub,x)-1));;
gap> clsr:=ClosureCWCell(map[1],2,sub[1])[2];;
gap> seq:=CWSubcomplexToRegularCWMap([map[1],clsr]);;
gap> tub:=RegularCWComplexComplement(seq);
Testing contractibility...
3501 out of 3501 cells tested.
The input is compatible with this algorithm.
gap> for i in [2..Length(sub)] do
>     clsr:=ClosureCWCell(tub,2,sub[i])[2];;
>     seq:=CWSubcomplexToRegularCWMap([tub,clsr]);;
>     tub:=RegularCWComplexComplement(seq);
> od;
Testing contractibility...
3612 out of 3612 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
3693 out of 3693 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
3871 out of 3871 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
3925 out of 3925 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4084 out of 4084 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4216 out of 4216 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4348 out of 4348 cells tested.
The input is compatible with this algorithm.
```

```
Testing contractibility...
4529 out of 4529 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4688 out of 4688 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4723 out of 4723 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
4918 out of 4918 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5107 out of 5107 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5269 out of 5269 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5401 out of 5401 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5548 out of 5548 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5702 out of 5702 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
5846 out of 5846 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6027 out of 6027 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6089 out of 6089 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6124 out of 6124 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6159 out of 6159 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6349 out of 6349 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6467 out of 6467 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6639 out of 6639 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
```

```
6757 out of 6757 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
6962 out of 6962 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7052 out of 7052 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7242 out of 7242 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7360 out of 7360 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7470 out of 7470 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7561 out of 7561 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7624 out of 7624 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7764 out of 7764 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7904 out of 7904 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
7979 out of 7979 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8024 out of 8024 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8086 out of 8086 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8148 out of 8148 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8202 out of 8202 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8396 out of 8396 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8534 out of 8534 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8625 out of 8625 cells tested.
```

```
The input is compatible with this algorithm.
Testing contractibility...
8736 out of 8736 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8817 out of 8817 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
8983 out of 8983 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9073 out of 9073 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9218 out of 9218 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9323 out of 9323 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9442 out of 9442 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9487 out of 9487 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9538 out of 9538 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9583 out of 9583 cells tested.
The input is compatible with this algorithm.
Testing contractibility...
9634 out of 9634 cells tested.
The input is compatible with this algorithm.
gap> Size(tub);
9685
gap> total_time_1:=0;;
gap> List([0..4],x->Homology(tub,x)); total_time_1:=total_time_1+time;;
[ [ 0 ], [ 0, 0 ], [ 0, 0, 0, 0 ], [ 0, 0 ], [  ] ]
gap> c:=ChainComplexOfUniversalCover(tub);; total_time_1:=total_time_1+time;;
gap> l:=Filtered(LowIndexSubgroups(c!.group,5),g->Index(c!.group,g)=5);; total_time_1:=total_time
gap> inv:=Set(l,g->Homology(TensorWithIntegersOverSubgroup(c,g),2)); total_time_1:=total_time_1+t
[ [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
 ]
gap> total_time_1;
3407
gap> hopf:=PureComplexComplement(HopfSatohSurface());;
gap> hopf:=RegularCWComplex(hopf);;
gap> Size(hopf);
4508573
gap> total_time_2:=0;;
gap> c_:=ChainComplexOfUniversalCover(hopf);; total_time_2:=total_time_2+time;;
```

```
gap> l_:=Filtered(LowIndexSubgroups(c_!.group,5),g->Index(c_!.group,g)=5);; total_time_2:=total_t
gap> inv_:=Set(l_,g->Homology(TensorWithIntegersOverSubgroup(c_,g),2));; total_time_2:=total_time
gap> total_time_2;
1116000
gap> inv_=inv;
true
```

References

[AL70] A. Atkin and J. Lehner. Hecke operators on $\gamma_0(m)$. *Math. Ann.*, 185:134--160, 1970. 180

[BCNS15] O. Braun, R. Coulangeon, G. Nebe, and S. Schoennenbeck. Computing in arithmetic groups with voronoï's algorithm. *J. Algebra*, 435:263--285, 2015. 183

[BE14] A. Bui and G. Ellis. The homology of $sl_2(F[1/m])$ for small m. *Journal of Algebra*, 408:102--108, 2014. 174

[Ber00] E. Berkove. *On the Mod-2 Cohomology of the Bianchi Groups*. Trans. of the AMS., 352, no. 10, 2000. 185

[Ber06] E. Berkove. *The integral Cohomology of the Bianchi Groups*. Trans. of the AMS., 358, no. 3, 2006. 185

[Ber16] N. Bergeron. *Torsion homology growth in arithmetic groups*. EuropeanMathematical Society, European Congress of Mathematicians, July 18-22, 2016. 195

[BL87] R. Brown and J.-L. Loday. Van kampen theorems for diagrams of spaces. *Topology*, 26:311--335, 1987. 162

[BLR20] E. Berkove, G. Lakeland, and A. Rahm. *The mod2 cohomology rings of congruence subgroups in the Bianchi groups*. J. Algebraic Combin. 52, 2020. 185

[Bro60] E. Brody. *The topological classification of the lens spaces*. Ann. of Math. 71, 163–184, 1960. 49

[CKL14] D. Coeurjolly, B. Kerautret, and J.-O. Lachaud. *Extraction of Connected Region Boundary in Multidimensional Images*. Image Processing On Line, 2014. 75

[CLV25] M. Condor, G. Liversidge, and M. Vsemirnov. *Generating paits for SL(n,Z)*. Journal of Algebra, Volume 662, pages 123--137, 2025. 196

[DJKV24] A. Dooms, E. Jespers, O. Konovalov, and H. Verrill. *Congruence, Congruence subgroups of SL(2,Integers), Version 1.2.7*. (GAP package), https://gap-packages.github.io/congruence, 2024. 173

[DPR91] R. Dijkgraaf, V. Pasquier, and P. Roche. *Quasi-Hopf algebras, group cohomology and orbifold models*. Nuclear Phys. B Proc. Suppl. 18B, 60-72, 1991. 115

[DSGG+16] M. Dutour Sikiric, H. Gangl, P. Gunnells, J. Hanke, A. Schuermann, and D. Yasaki. *On the cohomology of linear groups over imaginary quadratic fields.* J. Pure and Applied Algebra, volume 220, issue 7, 2016. 207

[EHS06] G. Ellis, J. Harris, and E. Skoldberg. Polytopal resolutions for finite groups. *J. Reine Angew. Math.*, 598:131--137, 2006. 171

[Eic57] M. Eichler. Eine verallgemeinerung der abelschen integrale. *Mathematische Zeitschrift*, 67:267--298, 1957. 169

[GM15] D. Goncalves and S. Martins. *Diagonal approximation and the cohomology ring of the fundamental groups of surfaces*. European Journal of Mathematics, 1, pp122--137, 2015. 19

[Gre13] J. Greene. *The lens space realization problem*. Annals of Mathematics 177, pages 449-511, 2013. 52

[Hat01] A. Hatcher. *Algebraic Topology*. Available online, 2001. 161

[Hor00] K. Horadam. *An introduction to cocyclic generalised Hadamard matrices*. Discrete Applied Math, 102, 115-130, 2000. 116

[IO01] K. Igusa and K. E. Orr. *Links, pictures and the homology of nilpotent groups*. Topology, Volume 40, Issue 6, pp-1125--1166, 2001. 107

[Joh16] F. Johnson. *Syzygies and dihedral resolutions for dihedral groups*. Communication in Algebra 44(5), pp 2034-2047, 2016. 145

[KFM08] L. H. Kauffman and J. Faria Martins. Invariants of welded virtual knots via crossed module invariants of knotted surfaces. *Compos. Math.*, 144(4):1046--1080, 2008. 38

[Kho01] I. Kholodna. *Low-dimensional homotopical syzygies*. PhD Thesis, National University of Ireland Galway, 2001. 145

[KS98] Y. V. Kuz'min and Y. S. Semenov. *On the homology of a free nilpotent group of class 2*. Mat. Sb. 189, no. 4, pp 49--82, 1998. 107

[Kso00] R. Ksontini. *Proprietes homotopiques du complexe de Quillen du groupe symetrique*. These de doctorat, Universitet de Lausanne, 2000. 9

[Kul91] R. Kulkarni. An arithmetic-geometric method in the study of the subgroups of the modular group. *American Journal of Mathematics*, 113, No. 6:1053--1133, 1991. 172

[LY24a] C. Liu and W. Ye. Crystallography, group cohomology, and lieb-schultz-mattis constraints. *https://arxiv.org/abs/2410.03607/*, 2024. 133, 134

[LY24b] C. Liu and W. Ye. Space group cohomology and lsm -- a github repository. *https://github.com/liuchx1993/Space-Group-Cohomology-and-LSM*, 2024. 133

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*. Proc. 8th Int'l Conf. Computer Vision, 2, pp 416--423, 2001. 74

[Mil58] J. Milnor. *On simply connected 4-manifolds*. International symposium on algebraic topology, Universidad Nacional Autonoma de Mexico and UNESCO, Mexico City, 1958. 18

[Moi52] E. Moise. *Affine structures in 3-manifolds V. The triangulation theorem and Hauptvermu- tung*. Annals of Math. 56, 96--114, 1952. 49

[Mos71] L. Moser. *Elementary surgery along a torus knot*. Pacific Journal of Mathematics, Vol. 38, No. 3, 1971. 52

[PY03] J. Przytycki and A. Yasukhara. *Symmetry of links and classification of lens spaces*. Geom. Dedicata 98, 57--61, 2003. 49

[Rah10] A. Rahm. *Cohomologies and K-theory of Bianchi groups using computational geometric models*. These de doctorat, Universite Joseph-Fourier -- Grenoble I, 2010. 183

[Rah13a] A. Rahm. *Higher torsion in the Abelianization of the full Bianchi groups*. LMS J. Comput. Math. 16, 2013. 185

[Rah13b] A. Rahm. *The homological torsion of PSL2 of the imaginary quadratic integers*. Trans. Amer. Math. Soc. 365, 2013. 185

[Rei35] K. Reidemeister. *Homotopieringe und Linsenraume*. Abh. Math. Sem. Univ. Hamburg 11 , 102–109, 1935. 49

[RF13] A. Rahm and M. Fuchs. *The integral homology of PSL2 of imaginary quadratic integers with nontrivial class group*. J. Pure Appl. Algebra 215, 2013. 185, 186

[RHM+13] A. Romero, J. Heras, G. Mata, J. Rubio, and F. Sergereart. *Spectral sequences for computing persistent homology of digital images*. Proceedings ACA 2013, 2013. 142

[Sat00] S. Satoh. Virtual knot presentation of ribbon torus-knots. *J. Knot Theory Ramifications*, 9(4):531--542, 2000. 38

[Sen11] M. H. Sengun. On the integral cohomology of bianchi groups. *Experimental Mathematics*, 20(4):487--505, 2011. 184

[Shi59] G. Shimura. Sur les integrales attachees aux formes automorphes. *Journal of the Mathematical Society of Japan*, 67:291--311, 1959. 169

[SK11] J. Spreer and W. Khuenel. *Combinatorial properties of the K3 surface: Simplicial blowups and slicings*. Experimental Mathematics Volume 20 Issue 2, 2011. 10, 15

[Ste07] W. Stein. *Modular forms, a computational approach*, volume 79. AMS Graduate Studies in Mathematics, 2007. 170, 178, 180

[SV83] J. Schwermer and K. Vogtmann. *The Integral Homology of SL2 and PSL2 of Euclidean Imaginary Quadratic Integer*. Comment. Math. Helvetica 58, 1983. 185

[Swa60] R. Swan. *Periodic resolutions for finite groups*. Annals of Mathematics 72, pages 267-291, 1960. 145

[Swa71a] R. Swan. Generators and relations for certain general linear groups. *Advances in Mathematics*, 6:1--77, 1971. 183

[Swa71b] R. Swan. *Generators and relations for certain Special Linear Groups*. Advances in Mathematics 6, 1--77, 1971. 200, 201

[Sym10] P. Symmonds. *ON THE CASTELNUOVO-MUMFORD REGULARITY OF THE COHOMOLOGY RING OF A GROUP*. Journal of the Americal Mathematical Society, Volume 23, 2010. 125, 126

[Thu02] W. Thurston. *The Geometry and Topology of Three-Manifolds*. http://www.msri.org/publications/books/gt3m/, 2002. 65

[TZ08] S. Tomoda and P. Zvengrowski. *Remarks on the cohomology of finite fundamental groups of 3-manifolds*. Geometry and Topology Monographs 14, 519-556, 2008. 115

[Vog85] K. Vogtmann. *Cohomology of Bianchi Groups, Math.* Math. Ann. 272, 1985. 185

[Web87] P. Webb. Subgroup complexes. *Proceedings of Symposia in Pure Mathematics, Volume 47*, 1987. 118

[Wie78] G. Wieser. *Computational arithmetic of modular forms*. Universitat Duisburg-Essen, 2007/8. 170

[Wue92] M. Wuestner. *An example of a nonsolvable Lie algebra*. Seminar Sophus Lie 2, 57-58, 1992. 123