Contemporary Mathematics

Geometric Algorithms for Resolutions for Bieberbach Groups

Marc Röder

Abstract. We present a geometry-based method for calculating free resolutions for Bieberbach groups. An implementation is compared to a more standard algebraic method and as an example, we calculate the additive integral homology of all 7-dimensional generalised orientable Hantzsche-Wendt manifolds.

1. Outline

Let $G \leq \mathrm{O}(n) \ltimes \mathbb{R}^n$ be a discrete and cocompact group. Then G is called a *crystallographic group*.
If G is torsion free (i.e. it does not contain elements of finite order), it is called a *Bieberbach group*.

A crystallographic group $G \leq \mathrm{O}(n) \ltimes \mathbb{R}^n$ is said to be *n-dimensional*. Bieberbach proved in [**Bie11, Bie12**] that the Bieberbach groups of dimension n correspond to the compact, flat, riemannian manifolds of dimension n.

In this paper, we compute the additive integral homology of all orientable generalised Hantzsche-Wendt manifolds of dimension 7. We achieve this by constructing, for the corresponding group of the manifold, a free $\mathbb{Z}G$-resolution of $\mathbb{Z}$ (with trivial action of G) from the natural action of G on $\mathbb{R}^n$. The calculations are done on a computer using the algorithms given in sections 3–5 as implemented in the `GAP` package HAPcryst.

Section 3 provides methods for dealing with orbits and stabilisers of the natural representation of G on $\mathbb{R}^n$. In Section 4 we give a method for finding a Dirichlet domain for G and Section 5 contains an algorithm that generates a free $\mathbb{Z}G$-resolution from such a polytope.

The performance of the algorithm is illustrated in section 6 and is compared to the algebraic methods for computing free resolutions of crystallographic groups implemented in the `GAP` package HAP. Furthermore, this section (6.3) contains the main results of this paper, namely the additive integral homology of the orientable Hantzsche-Wendt manifolds of dimension 7 mentioned above.

2000 *Mathematics Subject Classification.* 18G10, 20H15, 20J06.
Supported by Marie Curie Transfer of Knowledge grant MTKD-CT-2006-042685.

2. Preliminaries

As we discuss geometric methods, we will only consider the natural action of G as a group of affine transformations on $\mathbb{R}^n$. Any element $g \in G$ can be written as At with $A \in \mathrm{O}(n)$ and $t \in \mathbb{R}^n$ and we refer to A as the *linear part* of g and to t as the *translational part* of g.

We will use right-multiplication (in particular, we will use xd rather than $d(x)$) and denote by x^g the image of $x \in \mathbb{R}^n$ under the group element $g \in G$ while we will write h^g for conjugation in G with $h, g \in G$. Furthermore, we write G_x for the stabiliser of $x \in \mathbb{R}^n$ in G and x^G for the orbit of $x \in \mathbb{R}^n$ under G.

THEOREM 1 (Bieberbach, [**Bie11, Bie12**]). *Let G be a Bieberbach group acting naturally on $\mathbb{R}^n$. Then G contains a free Abelian normal subgroup T of rank n and finite index (pure translations).*

In other words: G contains a lattice of maximal rank. The finite group G/T is called the *point group* of G.

THEOREM 2 (Dirichlet-Voronoi construction). *Let $x \neq y \in \mathbb{R}^n$ and* $\mathrm{H}(x, y) := \{ a \in \mathbb{R}^n \mid \|x - a\| \leq \|y - a\| \}$.
Let G be a crystallographic group acting naturally on $\mathbb{R}^n$ and $x \in \mathbb{R}^n$ with $G_x = 1$. Define the Dirichlet domain *to be*

$$\mathrm{D}_G(x) := \bigcap_{y \in x^G} \mathrm{H}(x, y).$$

Then $D_G(x)$ is a polytope and contains a dense system of representatives of G-orbits (i.e. it is a fundamental domain).

Let $\mathfrak{P} = \mathrm{D}_G(x) \subseteq \mathbb{R}^n$ be the Dirichlet domain of the Bieberbach group G with respect to $x \in \mathbb{R}^n$. The orbit of $\mathfrak{P}$ under G is a tessellation of $\mathbb{R}^n$. Let $\mathfrak{P}_i$ be the free abelian group on the faces of dimension i of this tessellation.

Our aim is to calculate a chain complex from this tessellation with the natural boundary map by imposing some orientation on the faces:

$$0 \to \mathfrak{P}_n \to \cdots \to \mathfrak{P}_1 \to \mathfrak{P}_0 \to 0$$

As G is torsion free, we can then identify faces with group elements and get a free $\mathbb{Z}G$ resolution of $\mathbb{Z}$:

$$0 \to (\mathbb{Z}G)^{n_n} \to \cdots \to (\mathbb{Z}G)^{n_1} \to (\mathbb{Z}G)^{n_0}$$

where n_i is the number of orbits of G on the i-dimensional faces (notice that $n_n = 1$).

This construction can also be read in a more topological way. As Bieberbach groups correspond to compact, flat riemannian manifolds, the above construction can be viewed as the cellular chain complex of $\mathbb{R}^n$, where $\mathbb{R}^n$ is given a cell structure corresponding to that of the universal cover of a cellular manifold.

In the following sections, we will discuss algorithms for calculating $\mathrm{D}_G(x)$ and the corresponding chain complex.

3. Orbit-Stabiliser Type Algorithms

In this section we look at algorithms that will be needed during the calculation of Dirichlet domains and the resolutions defined by the associated tessellation.

Algorithm 1 calculates the point group of an n-dimensional crystallographic group G and a system of representatives for it in G. We assume the existence of a function LIN that returns the linear part of an element from G.

Algorithm 1 Calculating The Point Group And Representatives For It

```
1: procedure PointGroupRepresentatives(G)
2:     Set γ = R = ∅
3:     function recursion(h, Γ, P, R)
4:         for g ∈ Γ do
5:             if Lin(hg) ∉ P then
6:                 add hg to R
7:                 add Lin(hg) to P
8:                 recursion(hg, Γ, P, R)
9:             end if
10:        end for
11:    end function
12:    Let Γ be a set of generators of G
13:    Let P := ∅; R := ∅
14:    recursion(1, Γ, P, R)
15:    return point group P with representatives R
16: end procedure
```

The orbits of an n-dimensional crystallographic group G are infinite. So for computational purposes, the orbit of $x \in \mathbb{R}^n$ is represented by the set $[0,1)^n \cap x^G$ (the intersection with a half-open unit cube centred in $(1/2)^n$). So two vectors are conjugate under G if and only if their orbit parts are equal.

Algorithm 2 calculates this orbit-part of x under G. Note that the algorithm requires the vector x to be contained in $[0,1)^n$. Of course, this restriction can be removed by applying an appropriate translation. The reason for keeping it is that otherwise the orbit representatives of x do not necessarily contain x. An algorithm for orbit and stabiliser on point sets is just a slight variation of Algorithm 2. In this case, the orbit representatives are all images that contain a point of $[0,1)^n$.

Algorithm 3 calculates all elements of a crystallographic group G that map a vector x into a given polytope $\mathfrak{P}$.

4. Calculating a Dirichlet Domain

Algorithm 4 calculates the Dirichlet domain $\mathrm{D}_G(x)$ of the n-dimensional Bieberbach group G with translation subgroup $T \trianglelefteq G$ with respect to a given starting point $x \in \mathbb{R}^n$.

NOTE. Some remarks on algorithm 4 and its implementation:

- The algorithm is designed to work with a system where group computation and convex-hull computations are done separately and there is some overhead involved in changing from one sort of computation to the other. The algorithm was implemented in `GAP` using an interface to polymake for the convex hull computations [**GAP07, Röd08b, Röd08a, GJ**]. For a different algorithm, see [**Eng81**].

Algorithm 2 Orbit(part)-Stabiliser

Require: crystallographic group G, vector $x \in [0,1)^n$
1: Let P be a set of point group representatives of G (alg. 1)
2: Let $O := \emptyset$, $S := \emptyset$
3: **for** $g \in P$ **do**
4: $\quad i := x^g \mod 1$
5: $\quad$ **if** $i \notin O$ **then**
6: $\quad\quad$ Add i to O
7: $\quad$ **else if** $x^g = x$ **then**
8: $\quad\quad$ Add i to S
9: $\quad$ **else if** $i = x$ **then**
10: $\quad\quad$ Let t be the translation mapping x^g to x
11: $\quad\quad$ Add gt to S
12: $\quad$ **end if**
13: **end for**
14: **return** orbit part O, stabiliser $\langle S \rangle$

Algorithm 3 Mapping a Point Into a Polytope

Require: G crystallographic group, $x \in \mathbb{R}^n$, $\mathfrak{P} \subseteq \mathbb{R}^n$ polytope
1: **procedure** MapsToPolytope(G,x,$\mathfrak{P}$)
2: $\quad M := \emptyset$
3: $\quad$ Let P be a set of point group representatives (alg. 1)
4: $\quad$ Choose a point v in the interior of $\mathfrak{P}$
5: $\quad$ Let $d := \max\{|v - y| \mid y \in \mathrm{vert}(\mathfrak{P})\}$
6: $\quad$ **for** $g \in P$ **do**
7: $\quad\quad i := x^g$
8: $\quad\quad$ Let $\mathfrak{T}$ be the set of translations mapping i into $v + [-d/2, d/2]^n$
9: $\quad\quad$ **for** $t \in \mathfrak{T}$ **do**
10: $\quad\quad\quad$ **if** $i^t \in \mathfrak{P}$ **then**
11: $\quad\quad\quad\quad$ Add gt to M
12: $\quad\quad\quad$ **end if**
13: $\quad\quad$ **end for**
14: $\quad$ **end for**
15: $\quad$ **return** M
16: **end procedure**

- In line 1 the initial polytope can be chosen to be any polytope of the form $\mathrm{D}_S(x)$ with suitable $S \subseteq G$.
- For the calculation of vertices and facets in line 4 a convex hull computation is performed. The implementation of this part uses the computational geometry package polymake [**GJ**] as a black box.
- In line 15, the value of b is chosen by experiments. It balances the load between convex hull computations and the translation calculations of line 13
- The same algorithm can also be used to produce a fundamental domain for a crystallographic (non-Bieberbach) group. For this, x has to be a point with trivial stabiliser in G.

Algorithm 4 Calculating a Dirichlet domain

calculate the initial polytope $\mathfrak{P} = \mathrm{D}_T(x)$
Set $I = V_n = \emptyset$
repeat
  Calculate vertices $\mathrm{vert}(\mathfrak{P})$ and facets $\mathrm{fac}(\mathfrak{P})$ of $\mathfrak{P}$
  $V := \mathrm{vert}(\mathfrak{P}) \setminus V_n$
  if $V = \emptyset$ **then**
    return $\mathfrak{P}$
  end if
  Set $N = I := \emptyset$
  repeat
    choose random $v \in V$ and remove it from V
    Add v to V_n
    $N := \{\, g \in G \mid v^g \in \mathfrak{P} - \mathrm{vert}(P) \}$ ▷ algorithm 3
    Append $\left\{ H_x(x^g) \mid g \in N \cup N^{-1} \right\}$ to I
  until $V = \emptyset$ or $|I| > b$ for some suitable bound b
  Redefine $\mathfrak{P}$ by the inequalities $\mathrm{fac}(\mathfrak{P}) \cup I$
until $I = \emptyset$
return $\mathfrak{P}$

LEMMA 3. *Let G be an n-dimensional Bieberbach group and $x \in \mathbb{R}^n$. Then algorithm 4 terminates in finitely many steps and the returned object is the Dirichlet domain with respect to x.*

PROOF. Let $\mathfrak{P}$ be a polytope from Algorithm 4 approximating the Dirichlet domain. Suppose that $N = \{\, g \in G \mid v^g \in \mathfrak{P} - \mathrm{vert}(\mathfrak{P}) \} \neq \emptyset$. Then $\mathfrak{P} \cap \mathfrak{P}^g \neq \emptyset$ either contains a non-trivial open set for some $g \in N$ or $\mathfrak{P}$ is already a fundamental domain, as it contains one. So let's suppose we are in the first case.

Then $\mathfrak{P}' \subsetneq \mathfrak{P}$ where P' is the polytope defined by the inequalities $\mathrm{fac}(\mathfrak{P}) \cup \{H_x(x^g), H_x(x^{g^{-1}})\}$. In particular, $\mathfrak{P}'$ is smaller than $\mathfrak{P}$ and we still have $\mathfrak{P} \supseteq D_G(x)$. But as $|G/T| < \infty$, this process stops after finitely many steps.

The polytope $\tilde{\mathfrak{P}}$ in the final step has the property that $\tilde{\mathfrak{P}} \cap \tilde{\mathfrak{P}}^g$ has Lebesgue measure 0 for all $g \in G$. Hence $\tilde{\mathfrak{P}} = \mathrm{D}_G(x)$. □

5. Building a Resolution

Let G be a crystallographic group acting naturally on $\mathbb{R}^n$ and $\mathfrak{P} \subseteq \mathbb{R}^n$ a convex polytope which is a fundamental domain for G. From the face lattice of $\mathfrak{P}$, we get a mapping

$$\partial\colon \quad \mathfrak{P}_n \to \mathfrak{P}_{n-1} \to \cdots \to \mathfrak{P}_0 \tag{1}$$

where $\mathfrak{P}_i$ are the free abelian groups generated by the faces of $\mathfrak{P}$ in dimension i. This mapping extends to the full tessellation of $\mathbb{R}^n$ given by $\mathfrak{P}^G$.

Now let G be a Bieberbach group. This group acts freely on the faces of the tesselation $\mathfrak{P}^G$ (of any given dimension). So by identifying faces with group elements, we get a sequence of $\mathbb{Z}G$-modules

$$\partial\colon (\mathbb{Z}G)^{n_n} \to (\mathbb{Z}G)^{n_{n-1}} \to \cdots \to (\mathbb{Z}G)^{n_0} \tag{2}$$

where n_i is the number of orbits of G on $\mathfrak{P}_i$.

The aim of this section is to calculate a free resolution from (2). Observe that (2) is not an exact sequence of modules. In order to get an exact sequence, we will replace the map ∂ in (2) with a module homomorphism d which satisfies the boundary condition $d^2 = 1$.

LEMMA 4. *Let G be a crystallographic group, $\mathfrak{P} \subseteq \mathbb{R}^n$ a Dirichlet domain, and $\{\mathfrak{P}_n^G, \dots, \mathfrak{P}_0^G\}$ the corresponding face lattice of the tessellation. Let $F \in \mathfrak{P}_i^G$ be an i-face of this tessellation. Then every $(i-1)$-face in F is contained in exactly two i-faces of F.*

The proof of this lemma is a simple dimension argument.

NOTATION. Let $f \in (\mathbb{Z}G)^{n_k}$ and $B = \{g_1, \dots, g_{n_k}\}$ be a basis of $(\mathbb{Z}G)^{n_k}$. We say that f is a *face* if its expression in B is of the form $f = g_i h$ for some $h \in G$ and $1 \leq i \leq n_k$.

In Algorithm 5, we calculate the image of a generator in dimension k under the boundary d from the boundary map in dimension $k-1$. In the view of Lemma 4, we call faces in dimension $k-1$ "lines" and faces in dimension $k-2$ "points". Also, by abuse of notation, we will identify the images of a face under the mapping ∂ (from (2)) with a set of faces.

The general idea of algorithm 5 is as follows:

(1) The boundary of the 1-faces is chosen such that for each 1-face one of the 0-faces contained in it has $+1$ as a sign and the other one has -1.
(2) For a given face b_i^k, calculate its points and lines. For every point of b_i^k find the lines of b_i^k incident with it (exactly two, by Lemma 4).
(3) Now assign signs to the lines $b_i^k\partial$ to get a boundary map d satisfying $d^2 = 0$. Start with any point p_0 and a line l containing p_0. Set the sign of l to $+1$.
(4) Let l_2 be the second line containing p. If the sign of p in $l_2 d$ is the same as in ld, set the sign of l_2 in $b_i^k d$ to -1. Note that d is already known on the lines by induction.
(5) Do the same with all other points in $l, l_1, \dots$ Because of connectedness, all lines will eventually be processed.

The algorithm assumes an ordering on the generators of every module of the sequence (2). This is used to determine the orientation of the faces in dimension 1 and again to find the line l from which the sign- calculation is started. This is just done for convenience as it ensures the same output in every run and makes debugging easier.

6. Performance of the Implementaion and Examples

The algorithms described in the previous sections are implemented in `GAP` as part of a package called HAPcryst [**GAP07, Röd08a**]. The `GAP` package HAP [**Ell08**] has methods for the calculation of resolutions that can be applied to crystallographic groups as well, so it is natural to compare the two.

It is, however, worth noting that the algorithms used by HAP are very different from the ones in HAPcryst and the returned resolutions can be quite different. The most obvious difference is that HAPcryst only calculates resolutions for Bieberbach groups whereas HAP handles a much larger class of groups. On the other hand,

Algorithm 5 Boundary of Generator

Require: Bieberbach group G, for every module $(\mathbb{Z}G)^{n_k}$ from (2): a basis $B^k = \left\{ b_i^k \;\middle|\; 1 \le i \le n_k \right\}$, mappings ∂ from (2)

```
procedure BoundaryOfGenerator(b_i^k, ∂)
    if k=1 then
        write b_i^1 ∂ := b_{ℓ_1}^0 g_1 + b_{ℓ_2}^0 g_2 with b_{ℓ_1}^0, b_{ℓ_2}^0 ∈ B^0 and g_1, g_2 ∈ G
        if ℓ_1 > ℓ_2 then
            Define b_i^1 d := b_{ℓ_1}^0 g_1 − b_{ℓ_2}^0 g_2
        else
            Define b_i^1 d := b_{ℓ_1}^0 g_1 + b_{ℓ_2}^0 g_2
        end if
    else
        Let L be the set of faces in b_i^k ∂                ▷ The "lines" to consider
        Choose l ∈ L and define λ := {l}
        π := ∅                                              ▷ The "points" that are already done
        repeat
            λ_n := ∅
            for l ∈ λ do
                P := l∂ − π
                for p ∈ P do
                    Add p to π
                    Let s be the sign of p in ld
                    Let l_2 ≠ l be the second (k − 1)-face in b_i^k ∂ containing p
                    if l_2 ∈ L then
                        Remove l_2 from L
                        Let s_2 be the sign of p in l_2 d
                        if s = s_2 then
                            Define the sign of l_2 in b_i^k d to be −1
                            Add −l_2 to λ_n
                        else
                            Define the sign of l_2 in b_i^k d to be +1
                            Add l_2 to λ_n
                        end if
                    end if
                end for
            end for
            overwrite λ with the value of λ_n
        until λ = ∅
    end if
    return b_i^k d
end procedure
```

HAP calculates only finitely many terms of a resolution with infinitely many non-trivial terms while the resolution obtained from HAPcryst has only finitely many non-trivial terms (all of which are calculated).

We consider a couple of example cases to illustrate the performance of the two algorithms in different areas.

	timing h:mm:ss			
G	HAPcryst	HAP (9)	HAP (5)	dmin – dmax
1	0:00:07	0:00:38	0:00:04	1 – 70
2	0:03:55	0:01:22	0:00:07	18 – 601
3	0:06:00	0:01:43	0:00:09	27 – 1 031
4	0:14:21	0:01:51	0:00:12	34 – 1 750
5	0:00:08	0:01:01	0:00:05	1 – 70
6	0:19:16	0:10:57	0:00:26	45 – 1 937
7	28 hours	12:33:10	0:05:34	142 – 22 122
8	0:41:10	0:48:39	0:00:59	44 – 2 689
9	1:39:57	1:34:19	0:01:38	68 – 4 473
10	0:00:26	0:01:42	0:00:06	4 – 155
11	0:00:31	0:06:18	0:00:12	4 – 155

TABLE 1. Results for Resolutions of 8-dimensional Hyperkähler Manifolds

The resolutions calculated by HAPcryst depend on the calculation of a Dirichlet domain with respect to some point. In all cases, this point was chosen to be the origin.

Note that the times for HAPcryst are not CPU time, but "wall time" while the times for HAP are CPU time. It is also woth noting that HAPcryst does some calculations in random order which can lead to a slight variation in runtimes. All tests were done on an ordinary desktop computer.

6.1. Low Dimensional Groups. The fact that the implementation in HAPcryst produces quite some overhead by calling polymake for convex hull computations leads to bad performance in low dimensional cases. For the group `SpaceGroup(3,165)` which has point group $\mathbb{Z}_6$, calculating a resolution with HAPcryst takes almost 2 seconds. `ResolutionAlmostCrystGroup` calculates a free resolution with 4 terms in 0.2 seconds and a resolution with 10 terms in about 1.2 seconds.

6.2. Small Point Groups: Hyperkähler Manifolds. As an example for groups with small point group, we use Hyperkähler manifolds of dimension 8. In [**DM05**], the Poincaré polynomials and the structure of the cohomolgy rings of these manifolds are calculated (theoretically). Hyperkähler manifolds are compact, flat, and riemannian so they correspond to Bieberbach groups. The point groups of these Bieberbach groups are cyclic of order $2, 4, 3$, and 6 (in cases 1–4, 5–7, 8–10 and 11, enumeration as in [**DM05**]).

We calculate 9 terms of a resolution using HAP (Table 1). As Hyperkähler manifolds are orientable, the corresponding groups are Poincaré duality groups and hence 5 terms of a resolution are enough to calculate all (additive) homology and cohomology groups. So the timing for calculating 5 terms in HAP is added as well. The minimum and maximum of the dimensions of the modules of the resolution HAPcryst calculated are given in the last column (where the minimum is taken over the terms 0 to 7, the 8th term is one-dimensional by construction). The dimensions of HAP's resolutions are 1, 9, 37, 93, 163, 219, 247, 255, 256 in all cases.

6.3. Large Point Groups in Dimension 7: Hantzsche-Wendt Groups. For an example of groups with large point group, we look at the orientable generalised Hantzsche-Wendt groups in dimension 7. These are Bieberbach groups with

Group number [**MR99**]	H_0	H_1	H_2	H_3	H_4	H_5	H_6	H_7
1, 30, 44, 60	$\mathbb{Z}$	$\mathbb{Z}_2^6$	$\mathbb{Z}_2^8$	$\mathbb{Z}_4^6$	$\mathbb{Z}_2^8$	$\mathbb{Z}_2^6$	0	$\mathbb{Z}$
2, 3, 5, 6, 11, 13, 15, 17, 23, 24, 35, 39	$\mathbb{Z}$	$\mathbb{Z}_2^6$	$\mathbb{Z}_2^9$	$\mathbb{Z}_4^2 \oplus \mathbb{Z}_2^{10}$	$\mathbb{Z}_2^9$	$\mathbb{Z}_2^6$	0	$\mathbb{Z}$
4, 7, 10, 16, 19, 20, 22, 25, 27, 31, 32, 33, 34, 36, 37, 38, 40, 42, 43, 47, 48, 50, 52, 53, 55, 56, 57, 58, 59, 61, 62	$\mathbb{Z}$	$\mathbb{Z}_2^6$	$\mathbb{Z}_2^8$	$\mathbb{Z}_4^4 \oplus \mathbb{Z}_2^4$	$\mathbb{Z}_2^8$	$\mathbb{Z}_2^6$	0	$\mathbb{Z}$
8, 9, 12, 14, 18, 21, 26, 28, 29, 41, 45, 46, 49, 51, 54	$\mathbb{Z}$	$\mathbb{Z}_2^6$	$\mathbb{Z}_2^8$	$\mathbb{Z}_4^2 \oplus \mathbb{Z}_2^8$	$\mathbb{Z}_2^8$	$\mathbb{Z}_2^6$	0	$\mathbb{Z}$

TABLE 2. Integral Homology of 7-Dimensional Hantzsche-Wendt Manifolds

an elementary abelian point group of order 2^6. Up to isomorphism, there are 64 groups of this type [**MR99**].

Calculating resolutions with HAPcryst for all of these 62 groups takes about 41 minutes in total. The pointwise minimum and maximum of the module-dimensions in the resulting resolutions are (3, 33, 113, 183, 157, 71, 15) and (3, 142, 490, 742, 630, 280, 42), and for both bounds there is actually a group with a resolution having these dimensions.

From these resolutions, the integral homology groups can be calculated by HAP. The results of this calculation are listed in Table 2 (enumeration as in [**MR99**]).

HAP is able to calculate resolutions of length 3 for all 62 groups in 26 minutes while calculating 4 terms takes many hours for a single group.

6.4. The General Case in Dimensions 5 and 6. The Bieberbach groups up to dimension 6 are known [**CS01**] and available in various electronic formats from the website of the CARAT project [**CAR**].

There are 1060 Bieberbach groups in dimension 5 and calculating free resolutions for all of them with HAPcryst takes just under one hour in total. Whereas HAP needs 41 hours (median: 2.5 seconds) to calculate resolutions with only 4 terms for all groups.

Now consider an example of a Bieberbach group in dimension 6 with large non-abelian point group. Up to isomorphism, there is exactly one 6-dimensional Bieberbach group with point group $\mathbb{Z}_6 \times \mathrm{Alt}_4$. Using HAPcryst, we get a free resolution with modules of dimension 20, 102, 194, 176, 79, 16, 1 in 30 seconds of which 5 seconds are spent for convex hull computations. HAP calculates 3 terms of a free resolution in 6 minutes and 4 terms in 11 days (dimensions $1, 8, 31, 81, 169$).

Calculating free resolutions for the 176 Bieberbach groups in dimension 6 which have a point group of size at least 36 takes a little more than 12 hours (average per group 4 minutes, median 14 seconds).

7. Concluding Remarks

The geometric approach described in this paper provides a generic method for calculating free resolutions for Bieberbach groups. In contrast to existing algebraic methods, the geometric method finds a resolution with only finitely many non-trivial terms and calculates all of them.

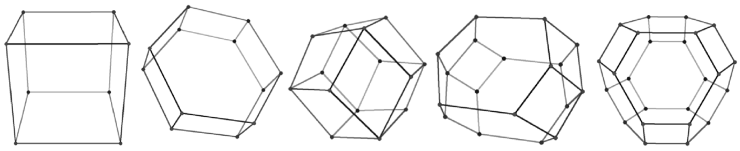

FIGURE 1. Combinatorial types of Dirichlet domains for group IT_4

We have shown in section 6.4 that the method can be used to calculate free resolutions of groups of dimension 5 (and probably 6) in reasonable time. Experiments suggest that calculations in higher dimensional groups are possible at least in cases where the group is "well-behaved" (i.e. has a Dirichlet domain with not too many faces). As seen in 6.1 and 6.2, the use of the algebraic methods of HAP can, however, be much faster than the geometric approach for groups of low dimension or small point group when only few terms of a resolution are needed. This is the case when calculating the homology of groups for Bieberbach groups with Poincaré duality, for example.

7.1. Possible Extensions and Further Work. The construction of a free resolution as a geometric object suggests a number of further applications. It seems possible to calculate a contracting homotopy on this resolution in a geometric way. This could then be used to calculate the cohomology rings for Bieberbach groups. Another possibility is the construction of resolutions of manifolds from others, if the new manifold is constructed terms of fundamental domains (this applies to direct sums, for example).

While experiments suggest that for groups given in the notation of `GAP`'s crystallographic groups library and the CARAT catalogue [**CAR**], the origin is a good choice, it would be very desirable to have a method to find a suitable starting point for resolution calculations.

EXAMPLE (Some Dirichlet domains in dimension 3). Experiments show that there are at least 136 combinatorial types of Dirichlet domains for the 10 Bieberbach groups in dimension 3. Consider the following small example:

The group $IT_4 := \left\langle \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \right\rangle$ acts as a 3-dimensional Bieberbach group on the space of affine row-vectors of the form $(x_1, x_2, x_3, 1)$ by multiplication from the right. Experiments show that this group admits at least 5 different combinatorial types of Dirichlet domains (see Figure 1 for illustrations).

Figure 2 shows the combinatorial type of $\mathrm{D}_G((x, 0, y))$ for $0 \leq x, y < 1$. Every shade of gray represents one combinatorial type. There are very light gray points at the intersections of the dark-gray lines and the black circles – this is probably hard to see in print. Note that it is enough to know $\mathrm{D}_G(v)$ in this region because of symmetry. For more geometric models of Dirichlet domains for 3-dimensional Bieberbach groups see [**Röd**].

8. Acknowledgements

The author would like to thank Graham Ellis for helpful suggestions, discussions, and for pointing out this problem.

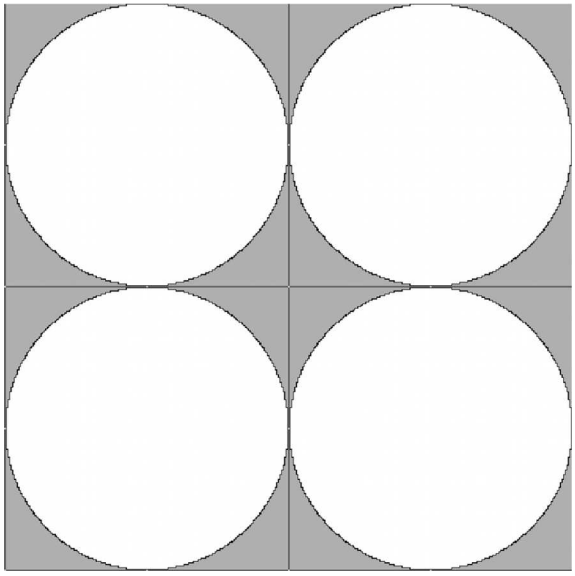

FIGURE 2. Location of combinatorial types in $(x, 0, z)$, $0 \leq x, z < 1$

References

[Bie11] L. Bieberbach, *Über die Bewegungsgruppen der euklidischen Räume. (Erste Abh.)*, Math. Ann. **70** (1911), 297–336 (German).

[Bie12] ———, *Über die Bewegungsgruppen der euklidischen Räume. (Zweite Abhandlung.) Die Gruppen mit einem endlichen Fundamentalbereich.*, Math. Ann. **72** (1912), 400–412 (German).

[CAR] *Torsion free space group database*, `http://wwwb.math.rwth-aachen.de/carat/bieberbach.html`.

[CS01] Carlos Cid and Tilman Schulz, *Computation of five and six dimensional bieberbach groups*, Experimental Mathematics **10** (2001), no. 1, 109–115.

[DM05] I.G. Dotti and R.J. Miatello, *On the cohomology ring of flat manifolds with a special structure*, Revista de la Unión Matemática Argentina. **46** (2005), no. 2, 133–147.

[Ell08] Graham Ellis, *HAP, version 1.8.6*, `http://hamilton.nuigalway.ie/Hap/www/index.html`, 2008.

[Eng81] Peter Engel, *Über Wirkungsbereichsteilungen von kubischer Symmetrie*, Zeitschrift für Kristallographie **154** (1981), no. 3-4, 199–215 (German).

[GAP07] *GAP – Groups, Algorithms, and Programming, Version 4.4.10*, `http://www.gap-system.org`, 2007.

[GJ] Ewgenij Gawrilow and Michael Joswig, *polymake*, `http://www.math.tu-berlin.de/polymake/`.

[MR99] R.J. Miatello and J.P. Rossetti, *Hantzsche-Wendt manifolds of dimension 7.*, Proceedings of the 7th international conference, DGA 98, and satellite conference of ICM in Berlin, Brno, Czech Republic (Ivan Kolář, ed.), Masaryk University, 1999, pp. 379–390.

[Röd] Marc Röder, *Geometric models for dirichlet domains of 3-dimensional bieberbach groups*, submitted to `http://www.eg-models.de`.

[Röd08a] ———, *HAPcryst, version 0.1.6*, `http://www.gap-system.org/Packages/hapcryst.html`, 2008.

[Röd08b] ———, *polymaking, version 0.7.3*, `http://www.gap-system.org/Packages/polymaking.html`, 2008.

E-mail address: `roeder.marc@gmail.com`

DEPARTMENT OF MATHEMATICS, NATIONAL UNIVERSITY OF IRELAND, GALWAY