

# polyDB: A Database for Polytopes and Related Objects



Andreas Paffenholz

**Abstract** polyDB is a database for discrete geometric objects independent of a particular software. The database is accessible via web and an interface from the software package polymake. It contains various datasets from the area of lattice polytopes, combinatorial polytopes, matroids and tropical geometry.

In this short note we introduce the structure of the database and explain its use with a computation of the free sums and certain skew bipyramids among the class of smooth Fano polytopes in dimension up to 8.

**Keywords** Database • polyDB • Discrete geometry • Lattice polytopes • Reflexive polytopes • Smooth polytopes

**Subject Classifications** 52-04, 52B20

## 1 Introduction

In recent years availability of computational classifications of mathematical objects has proven to be an important and valuable tool to obtain new results, to check new ideas and to experiment with the objects to obtain insight into their structure and directions for further research.

We know the full list of smooth Fano polytopes (up to lattice equivalence) up to dimension 9 by an algorithm of Øbro [33], whose availability within the software package polymake has been the foundation e.g. for counter-examples to a conjecture of Batyrev and Selivanova [32] or the classification of simplicial, terminal, and reflexive polytopes with many vertices by Assarf et al. [5]. Availability of the same data in Magma [37] lead to the study of the poset of blowups by Higashitani [16] or the study of reflexive polytopes of higher index by Kasprzyk and

---

A. Paffenholz (✉)

TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany

e-mail: [paffenholz@opt.tu-darmstadt.de](mailto:paffenholz@opt.tu-darmstadt.de); [paffenholz@mathematik.tu-darmstadt.de](mailto:paffenholz@mathematik.tu-darmstadt.de)

NilI [24]. The classification of 0/1-polytopes up to dimension 6 by Aichholzer [2] was used in the study of permutation polytopes by Baumeister et al. [6].

We also know classifications of small oriented matroids by Miyata et al. [12], polytopes [21, 39], and reflexive polytopes up to dimension 4 by Kreuzer and Skarke [3, 26]. The symbolic data project by Gräbe et al. [13] aims to collect data from computer algebra and make it accessible in a structured and searchable form on their web page. The library *MIPLIB* by Koch et al. [25] collects discrete optimization problems for benchmarking of algorithms.

Most of these collections, however, cannot easily be used in a software package. Sometimes the data is only available in text format or, if searchable via a database, is connected to a specific software package or lacks a proper interface at all. For example, the small oriented matroids [12], polytopes [27, 39], or 0/1-polytopes [2] are available as text files, while access to the small groups library [7] is linked to GAP [36]. Altman et al. [3] have created a database for the reflexive polytopes up to dimension 4 computed by Kreuzer and Skarke [26], but it is currently not accessible at the link given in the paper.

On the other hand, the Graded Rings Database [8] project has a more general approach and provides data in a format both searchable via a web interface and accessible via a programmatic interface that can be used in software packages. It currently has a focus on data from combinatorial commutative algebra and toric geometry.

The new database `polyDB` aims to provide searchable data from a wide range of areas at a permanent location in an application independent format. It allows download in text format and access from any software package that provides an interface to the data. It is also searchable via a web interface at [db.polymake.org](http://db.polymake.org). Currently, one interface to a software package is implemented, in the software package `polymake` [4, 22]. The current collection of data is thus still inspired by the range of applications of `polymake` with data from combinatorial geometry, matroid theory, toric geometry and combinatorial topology.

In the following two sections we explain the concept of the database and introduce the interface implemented in `polymake` to access the data. The last section shows one application of the database and the interface. We will show that in dimensions up to 8 more than 80% of the smooth Fano polytopes arise from lower dimensional ones as a free sum of two lower dimensional smooth Fano polytopes or a certain skew sum construction of a smooth Fano polytope and a simplex. We give the count of polytopes decomposable in this way in Table 1. With a simple extension of the scripts one can also obtain the list of possible decompositions for each polytope.

**Table 1** Free sums, skew bipyramids and generalized smooth simplex sums among the smooth Fano polytopes

Dimension	2	3	4	5	6	7	8
Smooth Fano polytopes	5	18	124	866	7622	72,256	749,892
Free sums	1	5	28	176	1361	11,760	112,285
Skew bipyramids	1	9	57	489	4323	43,777	466,770
sg simplex-1 sums	2	13	66	556	4700	47,076	495,092
sg simplex-2 sums	1	3	31	232	2403	25,157	284,249
sg simplex-3 sums	–	1	4	52	515	6635	83,730
sg simplex-4 sums	–	–	1	5	81	961	14,598
sg simplex-5 sums	–	–	–	1	6	114	1609
sg simplex-6 sums	–	–	–	–	1	7	155
sg simplex-7 sums	–	–	–	–	–	1	8
sg simplex-8 sums	–	–	–	–	–	–	1
Total sg simplex sums	3	16	93	708	6283	61,961	657,380
Total decomposable	3	16	96	712	6346	62,331	660,792

The rows denoted by *sg simplex-n sums* for  $n$  between 1 and 8 count the simplex sums with a simplex of dimension  $n$ . The row denoted by *total sg simplex sums* gives the number of different generalized smooth simplex sums with a simplex of any dimension. The row *total decomposable* counts the number of different polytopes among the free sums and the smooth generalized simplex sums

## 2 polyDB

In this section we briefly introduce the structure of the database polyDB and the data sets already contained in it.

The database polyDB for discrete geometric objects is based on the open source NoSQL database MongoDB [31]. It has been set up at [db.polymake.org](http://db.polymake.org). The database stores its data as plain JSON documents grouped into *collections* and *databases* (To avoid confusion with this and the abstract database polyDB we will refer to this technical term introduced by MongoDB as a *collection group*). We use this to group collections from the same area of discrete geometry into a common collection group. E.g., the collection group *Objects in Tropical Geometry* currently contains two collections of such objects, the small *tropical oriented matroids* classified by Horn [17] and the *polytropes* classified by Kulas [27] and Tran [39]. polyDB stores data in a plain JSON format independent of any particular software package. See Fig. 1 for an example of an entry in the collection of smooth reflexive polytopes.

Each document contains one special entry *polyDB* (besides its *\_id*, which is required by MongoDB). Apart from this all other entries and their tags can be chosen freely depending on the data. The entry *polyDB* may specify format restrictions for the data and import or export specifications for various software packages, separated by subfields naming the software. This section may contain, e.g., information on the required version, authors of the data, and the method to load the data into the

```

{
  "_id" : "F.2D.3",
  "DIM":2,
  "FACETS" : [[1,0,1],[1,0,-1],[1,1,0],[1,-1,-1],[1,-1,0]],
  "VERTICES": [[1,-1,-1],[1,-1,1],[1,0,1],[1,1,0],[1,1,-1]],
  "F_VECTOR" : [5,5],
  "EHRHART_POLYNOMIAL_COEFF": ["1", "7/2", "7/2"],
  "H_STAR_VECTOR": [1,5,1],
  "CENTROID": ["1", "-2/21", "-2/21"],
  "N_LATTICE_POINTS":8,
  "NORMAL" : "true",
  "VERY_AMPLE" : "true",
  "LATTICE_VOLUME":7,
  "polyDB" : {
    [...]
  }
}

```

**Fig. 1** An entry in the collection of smooth Fano polytopes. Naming of the fields is in this example taken from standard properties of objects in `polymake`. However, there are no restrictions on field names

particular software package. Each collection group also has a separate collection *type\_information* that specifies the format of an entry in a collection and allows to store information applicable to all data sets in this collection, e.g., methods for import and export of the data. The web interface at [db.polymake.org](http://db.polymake.org) allows independent and searchable access to all data sets in `polyDB`.

There are currently five collections, grouped into four collection groups contained in `polyDB`. We give a brief introduction to each of the collections.

- The collection group *Lattice Polytopes* has the collection *Smooth Reflexive Polytopes* that contains low dimensional smooth reflexive polytopes based on the algorithm of Øbro [33]. Øbro used his algorithm to compute the data up to dimension 8. Later, dimension 9 was computed with an improved implementation of the algorithm by Lorenz and the author. There are 9,060,505 such polytopes.
- The collection group *Objects in Tropical Geometry* has two collections. The collection *Tropical Oriented Matroids* contains a list of 71 known non-realizable tropical oriented matroids. This data was provided by Horn [17]. The collection *Full-dimensional Polytopes in TP3* contains all 1013 polytopes in 3-dimensional tropical projective space. The collection was generated by Constantin Fischer from data of Joswig and Kulas [21] and Tran [39]. See [20] for a description.
- The collection group *Special Polytopes* has the collection *Faces of Birkhoff Polytopes* which contains all 5371 combinatorial types of faces up to dimension 8 of the Birkhoff polytope in any dimension [34].

- The collection group *Matroids* has the collection *Matroids on at most 12 elements*. This collection contains all 32,401,446 small matroids as computed by Miyata et al. [11, 12, 30].

Further collections are in preparation.

### 3 The `polymake` Interface to `polyDB`

The initiative for `polyDB` was started in 2013 by Silke Horn and the author as an *extension* for the software package `polymake` [19] with associated database. With the latest version 3.1 of `polymake` [38], released in March 2017, the interface to the database has been turned into a *bundled extension* for `polymake` that is directly delivered with the software and the database has been set up as an independent project.

However, the software package `polymake` currently provides the only interface for import of data into the database and methods to access and use it for computations. Given a search query, i.e. a list of restrictions on the properties of an object, MongoDB allows the retrieval of a single object satisfying the query, an array with all objects satisfying the query or a cursor that returns objects from the result set one after another. All three methods are also implemented in `polymake`. The implementation is based on the perl MongoDB driver [1]. With

```
polytope > db_info();
DATABASE: LatticePolytopes
This database contains various classes of lattice polytopes.

Collection: SmoothReflexive
A complete collection of smooth reflexive lattice polytopes
in dimensions up to 9, up to lattice equivalence. [...]
```

we can query which collection groups are available. The collection group and collection we want to use for our search are then specified with the keywords `db` and `collection` in any access function. The query itself is given as a perl hash. The query is not processed by `polymake` but directly handed over to MongoDB, so it allows all queries specified in the MongoDB query language. A specification of the full query language and its use from within perl can be found in the documentation of MongoDB [31] and the perl driver for it [1].

Here is an example returning an array of results.

```
polytope > $parray=db_query({"DIM"=>3, "N_FACETS"=>5},
polytope(2) > db=>"LatticePolytopes",
polytope(3) > collection=>"SmoothReflexive");
polytope > print $parray->size;
4
```

This shows that there are four polytopes in the collection *SmoothReflexive* that have dimension 3 and 5 facets. Using a loop over this array or a database cursor we can

check properties of each object returned. For example

```
polytope > $cursor=db_cursor({"DIM"=>3, "N_FACETS"=>5},
polytope (2) > db=>"LatticePolytopes",
polytope (3) > collection=>"SmoothReflexive");
polytope > while ( !$cursor->at_end() ) {
polytope (2) > $p=$cursor->next();
polytope (3) > print $p->N_LATTICE_POINTS, " ";
polytope (4) > }
34 30 31 30
```

defines a cursor over the collection *SmoothReflexive* successively returning all polytopes that satisfy the restrictions given in the query, i.e., that have five facets in dimension 3. Here it tells us that among the four polytopes found above, two have 30, one has 31 and one has 34 lattice points.

## 4 Decomposing Smooth Fano Polytopes

We illustrate the use of `polyDB` and its interface to `polymake` with a computation that uses the collection *SmoothReflexive* in the collection group *LatticePolytopes* to compute decompositions of smooth Fano polytopes in dimensions 1 to 8. With our computations we start a new statistics that counts how many of the smooth Fano polytopes can be generated from lower dimensional smooth Fano polytopes with some simple known polytope construction method that preserves both smoothness and reflexivity of the polytope. We consider three methods in this paper and determine how many of the smooth Fano polytopes in these dimensions are

- free sums of two smooth Fano polytopes
- a smooth skew bipyramid over a smooth Fano polytope as defined in [5], or
- a generalized simplex sum of a smooth Fano polytope with a smooth simplex.

This new construction method will be defined below.

All smooth skew bipyramids and many of the free sums are also generalized smooth simplex sums. We will also provide the total number of smooth Fano polytopes that can be decomposed with at least one of these constructions. The results are collected in Table 1.

We briefly explain the relevant notions. More background can, e.g., be found in the book of Ewald [10]. Let  $P \subseteq \mathbb{R}^d$  be a polytope with vertices  $v_1, \dots, v_r \in \mathbb{R}^d$ , i.e.,

$$P := \text{conv}(v_1, \dots, v_r) \tag{1}$$

is the convex hull of these points and none of the  $v_i$  can be omitted in the definition. We assume that  $P$  is full dimensional, i.e., the affine hull of  $P$  is  $\mathbb{R}^d$  (otherwise we can pass to a subspace). A polytope can equally be given as the intersection of a

finite number of half-spaces in the form

$$P = \{ x \mid Ax \leq b \} \tag{2}$$

for some  $A \in \mathbb{R}^{s \times d}$  and  $b \in \mathbb{R}^s$ . We can again assume that no inequality is redundant in this definition. In this case the rows of  $A$  are the *facet normals* of  $P$ . A *facet*  $F$  of  $P$  is the set of all  $x \in P$  that satisfy one of the inequalities in (2) with equality. A *face* of  $P$  is the common intersection in  $P$  of a subset of the facets (this may be empty). The vertices, which are the faces of dimension 0, are in the common intersection of at least  $d$  facets.

If  $0$  is strictly contained in the interior of  $P$ , then the *polar* or *dual polytope* is defined as

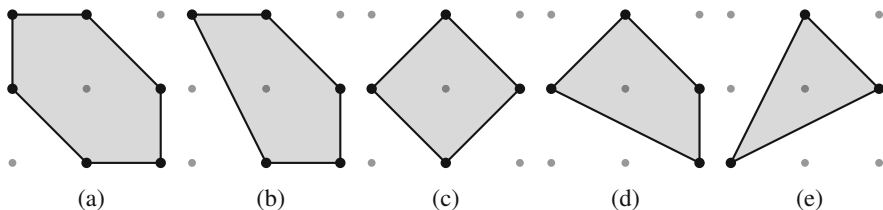
$$P^\vee := \{ v \mid \langle v, x \rangle \leq 1 \text{ for all } x \in P \}.$$

In fact, a finite subset of the inequalities in this definition suffice to define  $P^\vee$  (those corresponding to the vertices of  $P$ ), so that  $P^\vee$  is again a polytope. Further, we have  $(P^\vee)^\vee = P$ .

A lattice  $\Lambda$  is the integral span of a linearly independent set of vectors in  $\mathbb{R}^n$ . Up to a linear transformation we can assume that  $\Lambda$  is the integer lattice  $\mathbb{Z}^d \subset \mathbb{R}^n$ , and by passing to a subspace we can assume that  $n = d$ . With these assumptions a polytope  $P$  is a *lattice polytope* if all its vertices are in  $\mathbb{Z}^d$ .

In this case we can assume that both  $A$  and  $b$  are integral in (2), and that the greatest common divisor of the entries of each row of  $A$  (i.e., of the entries of each facet normal) is 1. A lattice polytope  $P$  is *reflexive* if  $P^\vee$  is again a lattice polytope. In this case  $b = \mathbf{1}$  in (2), and for both  $P$  and  $P^\vee$  the origin is the unique interior lattice point.  $P$  is *smooth* if the vertices of any facet of  $P$  are a lattice basis of  $\mathbb{Z}^d$ . In this case  $0$  is a strictly interior point of  $P$  and each facet has exactly  $d$  vertices, so  $P$  is *simplicial*. Moreover, the polar polytope is again a lattice polytope (in the dual lattice) whose vertices are the facet normals (the rows of  $A$ ), so  $P$  is also reflexive. Note that in the literature sometimes the polytopes polar to the ones defined here are called smooth.

It follows from a result of Hensley [15] and Lagarias and Ziegler [28] that there are only finitely many smooth reflexive polytopes in each dimension up to lattice equivalence (affine transformations preserving  $\mathbb{Z}^d$ ), as reflexive polytopes have exactly one interior lattice point. See Fig. 2 for the list of such polytopes in dimension 2. The complete list is contained in polyDB for  $d \leq 9$  in the collection *SmoothReflexive* of the database *LatticePolytopes*. Note however, that in the database we follow the above mentioned alternative definition and list the duals of the ones defined here. Yet, for the purpose of the following constructions it is easier to work with the definition given above, so we will use that one in the following. This requires that in the scripts we use for our computations below we polarize the polytopes obtained from the database. Sometimes this is, however, only done implicitly. This saves computation time, as it follows from the design of `polymake` that for reflexive polytopes the facets of the polytope are the vertices



**Fig. 2** The five 2-dimensional smooth Fano polytopes. (a)  $P_6$ . (b)  $P_5$ . (c)  $P_{4a}$ . (d)  $P_{4b}$ . (e)  $P_3$

of its dual. Also, as we will see below, most constructions can also be given for the duals of the polytopes.

We introduce several methods to construct a smooth Fano polytope from smaller ones. The most well known construction is the *free sum* of two polytopes  $P \subseteq \mathbb{R}^a$  and  $Q \subseteq \mathbb{R}^b$  that both contain the origin in their interior. This is the polytope

$$P \oplus Q := \text{conv}(\{(v, 0) \in \mathbb{R}^{a+b} \mid v \in P\} \cup \{(0, w) \in \mathbb{R}^{a+b} \mid w \in Q\}) .$$

We can also define this on the dual side. The *product* of polytopes  $P$  and  $Q$  is the polytope

$$P \times Q := \{ (x, y) \mid x \in P, y \in Q \} .$$

Then, if  $P$  and  $Q$  contain the origin in their interior,

$$P \oplus Q = (P^\vee \times Q^\vee)^\vee . \tag{3}$$

We will use this dual definition for the detection of free sums among the smooth Fano polytopes. See Fig. 3a for an example.

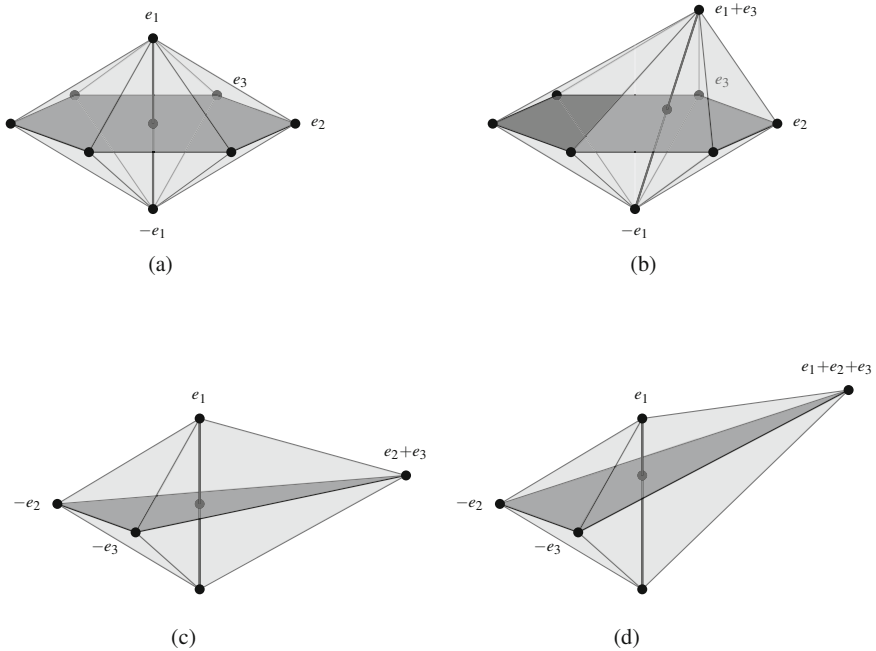
A *bipyramid* over a polytope  $P$  is the free sum of  $P$  with a segment  $S$  containing the origin in the interior. More generally, we say that  $Q$  is a *skew bipyramid* over  $P$  if  $Q$  has the same combinatorial type (the same face lattice) as a bipyramid over  $P$ . The two vertices coming from vertices of  $S$  are the two *apices* of  $Q$ .

If  $P$  is a smooth  $d$ -dimensional Fano polytope then we call the free sum with the segment  $[-1, 1]$  the *smooth bipyramid* over  $P$ . Let  $v$  be a vertex of  $P$  and  $\bar{v}$  its embedding into  $\mathbb{R}^{d+1}$  by adding a 0 at the end. Then the *smooth skew bipyramid* for vertex  $v$  as defined by Assarf et al. [5] is the polytope

$$\text{SBipy}(P, v) := \text{conv}(P \times \{0\} \cup \{-e_{d+1}, \bar{v} + e_{d+1}\}) .$$

Figure 3b shows an example of this definition. More generally, we say that  $Q$  is a *smooth generalized skew bipyramid* over  $P$  if  $Q$  is a skew bipyramid over  $P$  such that the two apices have lattice distance 1 from  $P$ . This class contains all smooth bipyramids and smooth skew bipyramids. The following proposition is an extension of Lemmas 1, 2 and 3 of [5]. The proof easily carries over into this more general setting.





**Fig. 3** Polytope constructions. (a) The free sum of a hexagon and a segment. This is at the same time also a proper bipyramid over the hexagon. (b) A skew bipyramid of a hexagon. The top apex has been shifted to  $e_1 + e_3$ . (c) A generalized simplex sum of a segment and a triangle. (d) Another generalized simplex sum of a segment with a triangle

**Proposition 4.1** *Let  $P$  and  $Q$  be smooth Fano polytopes. Then the free sum  $P \oplus Q$ , the smooth bipyramid and any smooth (generalized) skew bipyramid over  $P$  are again smooth Fano polytopes.*  $\square$

We further generalize this construction. Let  $P \subseteq \mathbb{R}^a$  be a smooth Fano polytope and  $Q \subseteq \mathbb{R}^b$  a smooth Fano simplex (this is unique up to lattice equivalence). Let  $v$  be a vertex of  $Q$ . Then  $R := P \oplus Q$  is a smooth Fano polytope and also any polytope  $R'$  obtained from  $R$  by replacing  $v$  with a lattice point  $v'$  in the hyperplane  $\mathbb{R}^a + v \subseteq \mathbb{R}^{a+b}$ , as long as  $R$  and  $R'$  have the same combinatorial type. This is again a simple extension of the proposition above. We call those polytopes *smooth generalized simplex sums*. Figure 3c, d shows two examples. Observe that any smooth (generalized skew) bipyramid is a simplex sum.

We can use `polymake` and `polyDB` to detect all free sums and smooth generalized simplex sums among the smooth Fano polytopes. Clearly, these two constructions overlap in various ways. Any proper bipyramid over all polytope  $P$  is also the free sum of a  $P$  with a segment, and polytopes may have more than one possible decomposition into a free sum. Many of the various possibilities to place the vertex  $v'$  for a generalized smooth simplex sum are lattice equivalent. Our

approach to detect all different instances is as follows: For a fixed dimension  $d$  we consider all possible splits of  $d$  as a sum of dimensions  $a$  and  $b$  and compute all free sums of smooth Fano polytopes in these two dimensions and all simplex sums of an  $a$ -dimensional smooth Fano polytope with a  $b$ -dimensional simplex. For each such polytope we run through the list of  $d$ -dimensional smooth Fano polytopes, check for lattice equivalence and store the name of the polytope we have found. We could also store the way we obtained it alongside, so that in the end we have a list of all possible splits for a given  $d$ -dimensional smooth Fano polytope.

We did the computation up to dimension 8. The results are given in Table 1. The free sums can be obtained with the small scripts given in Fig. 4. The first script `identify_smooth_polytope` takes a smooth Fano polytope, identifies it in the database and returns its name. The identification is based on the `polymake` function `lattice_isomorphic_smooth_polytopes`, that reduces the check whether two lattice polytopes are lattice isomorphic to a colored graph isomorphism problem (which is solved using `bliss` [23] or `nauty` [29]). Note that there is also the extension `LatticeNormalization` [18] to `polymake` that computes the lattice normal form of a lattice polytope (see [14] for a definition), but the reduction to colored graph isomorphism is more efficient for smooth polytopes. The simpler problem of checking combinatorial isomorphisms (i.e., graph isomorphism) can also be done with the `polymake`-function `canonical_hash` (also based on `bliss` or `nauty`). The second function `all_free_sums_in_dim` computes all possible free sums that lead to a  $d$ -dimensional smooth Fano polytope. As the database contains the polytopes dual to the ones we consider we use (3) and compute products instead of sums to avoid explicit dualization. For each product the function calls `identify_smooth_polytope` to identify it in the database. The function returns a list of all names (`_ids`) found in this way. If `splitinfo` is set to 1 it also returns all pairs of summands.

For the computation of the smooth generalized simplex sums we used the function `all_skew_simplex_sums_in_dim` available at [35]. For each combination of an  $a$ -dimensional smooth Fano polytope and a  $b$ -dimensional simplex with  $d = a + b$  we compute all possible lattice points for the shifted vertex  $v'$ , construct the polytope and again use `identify_smooth_polytope` to identify it in the database. Computation of all possible  $v'$  requires the computation of all lattice points in the hyperplane  $\mathbb{R}^a + v \subseteq \mathbb{R}^{a+b}$  that lead to a lattice polytope with the same combinatorial type as the proper free sum. This can be reduced to enumerating lattice points in the interior of a polytope, which is done in `polymake` via the interface to `Normaliz` [9]. As above the function returns a list of `ids`, and also all possible decompositions into a simplex sum if `splitinfo` is set to 1. You can save the scripts to a file in the current folder and load this into `polymake` via

```
polytope> script(<filename>);
```

```

use application "polytope";

sub identify_smooth_fano_in_polydb {
  my $p = shift;
  my $d = $p->DIM;
  my $nlp = new Int($p->N_LATTICE_POINTS);
  my $parray=db_query({"DIM"=>$d, "N_VERTICES"=>$p->N_VERTICES, "
    N_FACETS"=>$p->N_FACETS,
    "N_LATTICE_POINTS"=>$nlp, }, db=>"LatticePolytopes",
    collection=>"SmoothReflexive");
  foreach my $c ( @$parray ) {
    if ( lattice_isomorphic_smooth_polytopes($c,$p) ) { return $c
      ->name; }
  }
  die "polytope not found\n";
}

sub all_free_sums_in_dim {
  my ($d,%options) = @_;
  my $list;
  if ( $options{"splitinfo"} ) {
    $list = new Map<String,Set<Pair<String,String> > >;
  } else {
    $list = new Set<String>;
  }
  my $cur_options = { db=>"LatticePolytopes", collection=>"
    SmoothReflexive" };
  foreach my $n (1..$d/2) {
    my $curl=db_cursor({"DIM"=>$n}, $cur_options);
    while ( !$curl->at_end() ) {
      my $c1 = $curl->next();
      my $cur2=db_cursor({"DIM"=>$d-$n}, $cur_options );
      while ( !$cur2->at_end() ) {
        my $c2 = $cur2->next();
        my $name = identify_smooth_fano_in_polydb(product($c1,$c2))
          ;
        if ( $options{"splitinfo"} ) {
          my $split = new Pair<String,String>($c1->name,$c2->name);
          $list->{$name} += $split;
        } else {
          $list += $name;
        }
      }
    }
  }
  return $list;
}

```

**Fig. 4** A function to detect all free sums among the smooth Fano polytopes. This is a shortened version of the script given at [35]

Then the classification, e.g. in dimension 4, is obtained with

```

polytope > $fs = all_free_sums_in_dim(4);
polytope > print $fs->size;
28
polytope > $sb = all_skew_bipyramids_in_dim(4);
polytope > print $sb->size;
57
polytope > $s = new Set<String>;
polytope > foreach (1..4) {
polytope(2) > $st = skew_simplex_sums_in_dim(4,$_);
polytope(3) > print $st->size, " ";
polytope(4) > $s += $st;
polytope(5) > }
66 31 4 1
polytope > print $s->size;
93
polytope > print (($fs+$s)->size);
96
    
```

The scripts containing the functions `all_skew_bipyramids_in_dim` for skew bipyramids and `skew_simplex_sums_in_dim` for generalized smooth simplex sums are available from [35] and allow to store the possible decompositions. Note that the computation time for the decompositions grows quickly in the dimension. While dimension 4 runs in a few minutes on an Intel Xeon E5-4650, computations in dimension 8 took over a month.

From these computations we can, e.g., see that we have three different decompositions of the dual of the 5-dimensional polytope with index  $F.5D.0116$ . Its vertices are the rows of the matrix in Table 2a. We can decompose this into three different simplex sums. One is over dual of the 3-dimensional polytope  $P_3$  with index  $F.3D.0112$ . This is shown in Table 2b, where the shaded part corresponds to the vertices of  $P_3$ . The shifted vertex of the triangle is  $[0, 0, -2, 1, 1]$ . Note that the vertices are given as obtained by dualization from the database. Hence, the equality

**Table 2** Simplex sums leading to the dual of  $F.5D.0116$

0 0 0 0 1	0 1 1 0 0	-1 0 0 0 0	-1 0 0 0 0
0 0 1 0 -1	-1 0 0 0 0	0 0 0 -1 0	0 0 0 -1 0
0 0 -1 0 0	0 0 1 0 0	0 1 0 1 0	0 1 0 -1 0
0 0 0 -1 0	0 -1 0 0 0	0 -1 0 0 0	0 -1 0 0 0
0 0 0 0 -1	0 0 -1 0 0	0 0 0 1 0	0 0 0 1 0
-1 0 0 0 0	1 0 -1 0 0	0 0 -1 0 0	0 0 -1 0 0
0 -1 0 0 0	0 0 0 -1 0	1 0 1 2 0	1 0 1 2 0
0 1 0 0 1	0 0 0 0 -1	0 0 0 0 -1	0 0 0 0 -1
1 0 0 1 2	0 0 -2 1 1	0 0 0 -1 1	0 0 0 1 1

(a) Dual of  $F.5D.0116$

(b) Dual of  $F.3D.0112$  extended with skew triangle

(c) Dual of  $F.4D.0008$  extended with segment

(d) Dual of  $F.4D.0019$  extended with segment

is not directly visible from the vertices, as the two polytopes differ by a lattice isomorphism. We can check this with `polymake`.

```
polytope > $p3_ext = new Polytope(VERTICES=>
polytope (2) > [[1,0,1,1,0,0], [1,-1,0,0,0,0], [1,0,0,1,0,0],
polytope (3) > [1,0,-1,0,0,0], [1,0,0,-1,0,0], [1,1,0,-1,0,0],
polytope (4) > [1,0,0,0,-1,0], [1,0,0,0,0,-1],
polytope (5) > [1,0,0,-2,1,1]);
polytope > $p5 = new Polytope(VERTICES=>
polytope (2) > [[1,0,0,0,0,1], [1,0,0,1,0,-1], [1,0,0,-1,0,0],
polytope (3) > [1,0,0,0,-1,0], [1,0,0,0,0,-1], [1,-1,0,0,0,0],
polytope (4) > [1,0,-1,0,0,0], [1,0,1,0,0,1], [1,1,0,0,1,2]);
polytope > print lattice_isomorphic_smooth_polytope (
polytope (2) > polarize($p3_ext),polarize($p5));
1
```

Here, the variable `$p3_ext` contains the polytope  $P_3$  and `$p5` is  $P_5$ . As above we need to dualize for the isomorphism check. The check returns 1, which is the true-value for `polymake`.

The other two decompositions are over the 4-dimensional polytopes  $P_4^1$  and  $P_4^2$  with index F.4D.0008 and F.4D.0019. Those are shown in Table 2c and d. Again, the vertices of  $P_4^1$  and  $P_4^2$  are shaded. The shifted vertices of the 1-dimensional simplex are in the last line.

With this simple computation we have seen that over 80% of the smooth Fano polytopes can be obtained from at least one of the constructions considered here. Hence, for a structural description of all smooth Fano polytopes it suffices to look at the remaining less than 20%.

## References

1. A perl driver for MongoDB (2016). Available at [search.cpan.org/perldoc?MongoDB](http://search.cpan.org/perldoc?MongoDB)
2. O. Aichholzer, Extremal properties of 0/1-polytopes of dimension 5, in *Polytopes—Combinatorics and Computation (Oberwolfach, 1997)*, vol. 29, ed. by G. Kalai, G.M. Ziegler, DMV Seminar (Birkhäuser, Basel, 2000), pp. 111–130, [http://www.ist.tugraz.at/staff/aichholzer/research/rp/rcs/info01poly](http://www.ist.tugraz.at/staff/aichholzer/research/rp/rcs/info01poly/www.ist.tugraz.at/staff/aichholzer/research/rp/rcs/info01poly)
3. R. Altman, J. Gray, Y.H. He, V. Jejjala, B.D. Nelson, A Calabi-Yau database: threefolds constructed from the Kreuzer-Skarke list. *J. High Energy Phys.* (2) **158**, front matter+48 (2015)
4. B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, T. Rehn, `Polymake` in linear and integer programming. *Math. Program. Comput.* (2014, to appear). Available at [arxiv:1408.4653](http://arxiv.org/abs/1408.4653)
5. B. Assarf, M. Joswig, A. Paffenholz, Smooth Fano polytopes with many vertices. *Discrete Comput. Geom.* **52**, 153–194 (2014)
6. B. Baumeister, C. Haase, B. Nill, A. Paffenholz, On permutation polytopes. *Adv. Math.* **222**(2), 431–452 (2009)
7. H.U. Besche, B. Eick, E. O’Brien, Small groups library (2002). Available at [www.icm.tu-bs.de/ag\\_algebra/software/small/](http://www.icm.tu-bs.de/ag_algebra/software/small/)
8. G. Brown, A. Kasprzyk, The graded rings database. Available at [www.grdb.co.uk](http://www.grdb.co.uk)

9. W. Bruns, B. Ichim, T. Römer, R. Sieg, C. Söger, Normaliz. Algorithms for rational cones and affine monoids. *J. Algebra* **324**, 1098–1113 (2010). Available at [www.normaliz.uni-osnabrueck.de](http://www.normaliz.uni-osnabrueck.de)
10. G. Ewald, *Combinatorial Convexity and Algebraic Geometry*. Graduate Texts in Mathematics, vol. 168 (Springer, New York, 1996)
11. L. Finschi, K. Fukuda, Combinatorial generation of small point configurations and hyperplane arrangements, in *Discrete and Computational Geometry*. Algorithms and Combinatorics, vol. 25 (Springer, Berlin, 2003), pp. 425–440
12. K. Fukuda, H. Miyata, S. Moriyama, Complete enumeration of small realizable oriented matroids. *Discrete Comput. Geom.* **49**(2), 359–381 (2013)
13. H.G. Gräbe, Semantic-aware fingerprints of symbolic research data, in *Mathematical Software ICMS 2016*, vol. 9725, ed. by G.M. Greuel, T. Koch, P. Paule, A. Sommese. Theoretical Computer Science and General Issues (Springer, Cham, 2016)
14. R. Grinis, A. Kasprzyk, Normal forms of convex lattice polytopes (2013). Available at [arxiv:1301.6641](https://arxiv.org/abs/1301.6641)
15. D. Hensley, Lattice vertex polytopes with interior lattice points. *Pac. J. Math.* **105**(1), 183–191 (1983)
16. A. Higashitani, Equivalence classes for smooth Fano polytopes (2015). Available at [arxiv:1503.06434](https://arxiv.org/abs/1503.06434)
17. S. Horn, Tropical oriented matroids and cubical complexes. Ph.D. thesis, 2012
18. S. Horn, A. Paffenholz, Lattice normalization: a polymake extension for lattice isomorphism checks (2014). Available at [github.com/apaffenholz/lattice\\_normalization](https://github.com/apaffenholz/lattice_normalization)
19. S. Horn, A. Paffenholz, polyDB: a database extension for polymake (2014). Available at [github.com/solros/poly\\_db](https://github.com/solros/poly_db)
20. M. Joswig, Polytopes in the tropical projective 3-torus (2016). Available at [homepages.math.tu-berlin.de/~joswig/polymake/data/](http://homepages.math.tu-berlin.de/~joswig/polymake/data/)
21. M. Joswig, K. Kulas, Tropical and ordinary convexity combined. *Adv. Geom.* **10**(2), 333–352 (2010)
22. M. Joswig, B. Müller, A. Paffenholz, polymake and lattice polytopes, in *21st International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2009)*, ed. by C. Krattenthaler, V. Strehl, M.N. Kauers. Discrete Mathematics and Theoretical Computer Science Proceedings, AK (Association of Discrete Mathematics and Theoretical Computer Science, Nancy, 2009), pp. 491–502
23. T. Junttila, P. Kaski, bliss: a tool for computing automorphism groups and canonical labelings of graphs (2011). Available at [www.tcs.hut.fi/Software/bliss/](http://www.tcs.hut.fi/Software/bliss/)
24. A.M. Kasprzyk, B. Nill, Reflexive polytopes of higher index and the number 12 (2011). Available at [arxiv:1107.4945](https://arxiv.org/abs/1107.4945)
25. T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R.E. Bixby, E. Danna, G. Gamrath, A.M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, D. Salvagnin, D.E. Steffy, K. Wolter, MIPLIB 2010. *Math. Program. Comput.* **3**(2), 103–163 (2011). Available at [mpc.zib.de/index.php/MPC/article/view/56/28](http://mpc.zib.de/index.php/MPC/article/view/56/28)
26. M. Kreuzer, H. Skarke, Calabi-yau data (1997). Available at [hep.itp.tuwien.ac.at/~kreuzer/CY/](http://hep.itp.tuwien.ac.at/~kreuzer/CY/)
27. K. Kulas, Coarse types of tropical matroid polytopes (2010). Available at [arxiv:1012.3053](https://arxiv.org/abs/1012.3053)
28. J.C. Lagarias, G.M. Ziegler, Bounds for lattice polytopes containing a fixed number of interior points in a sublattice. *Can. J. Math.* **43**(5), 1022–1035 (1991). <https://doi.org/10.4153/CJM-1991-058-4>
29. B. McKay, nauty 2.5 (2013). Available at [cs.anu.edu.au/~bdm/nauty/](http://cs.anu.edu.au/~bdm/nauty/)
30. H. Miyata, S. Moriyama, K. Fukuda, Classification of oriented matroids (2013). Available at [www.imai.is.s.u-tokyo.ac.jp/~hmiyata/oriented\\_matroids/](http://www.imai.is.s.u-tokyo.ac.jp/~hmiyata/oriented_matroids/)
31. MongoDB v3.2 (2016). Available at [www.mongodb.com](http://www.mongodb.com)
32. B. Nill, A. Paffenholz, Examples of Kähler-Einstein toric Fano manifolds associated to non-symmetric reflexive polytopes. *Beitr. Algebra Geom.* **52**(2), 297–304 (2011). <https://doi.org/10.1007/s13366-011-0041-y>

33. M. Øbro, Classification of Smooth Fano Polytopes. Ph.D. thesis, University of Aarhus, 2007. Available at [pure.au.dk/portal/files/41742384/imf\\_phd\\_2008\\_moe.pdf](http://pure.au.dk/portal/files/41742384/imf_phd_2008_moe.pdf)
34. A. Paffenholz, Faces of Birkhoff polytopes. *Electron. J. Combin.* **22**(1), Paper 1.67, 36 (2015)
35. A. Paffenholz, `polymake_smooth_fano`: scripts for decompositions of Fano polytopes (2016). Available at [github.com/apaffenholz/polymake\\_smooth\\_fano](https://github.com/apaffenholz/polymake_smooth_fano)
36. The GAP Group, GAP – Groups, Algorithms, and Programming, Version 4.4.12. The GAP Group (2008). Available at [www.gap-system.org](http://www.gap-system.org)
37. The Magma Group, Magma computational algebra system (2017). Available at [magma.maths.usyd.edu.au/magma/](http://magma.maths.usyd.edu.au/magma/)
38. The `polymake`-Team, `polymake` release 3.1 (2017). Available at [github.com/polymake/polymake](https://github.com/polymake/polymake)
39. N.M. Tran, Polytopes and tropical eigenspaces: cones of linearity. *Discrete Comput. Geom.* **51**(3), 539–558 (2014)