*Séminaire Lotharingien de Combinatoire* **XX** (2026)
Article #54, 12 pp.

*Proceedings of the 38th Conference on Formal Power Series and Algebraic Combinatorics (Seattle)*

# UniTriSat: Unimodular Triangulations via SATISFIABILITY

Kyle Huang*[1] and Robert Lauff[2] and Charles Zhang[3]

[1]*Brandenburg University of Technology Cottbus-Senftenberg*
[2]*Technische Universität Berlin*
[3]*University of Vienna*

**Abstract.** Given a lattice polytope, it is natural to ask if it has a unimodular triangulation. Unimodular triangulations have connections to toric geometry, tropical geometry, and enumerative combinatorics, while also being an interesting property in their own right, for various classes of lattice polytopes. With the `Julia` package `UniTriSat`, we present a new algorithm for computing unimodular triangulations and regular unimodular triangulations, via translation to a SATISFIABILITY problem.

**Keywords:** lattice polytopes, unimodular triangulations, computer algebra, mathematical software, algorithms, optimization

## 1 Introduction & Motivation

A *unimodular triangulation* is a triangulation of a lattice polytope using only unimodular simplices–lattice simplices with least volume. Such a unimodular triangulation is *regular* if it can be induced as the lower convex hull of lifting the lattice points in an extra dimension.

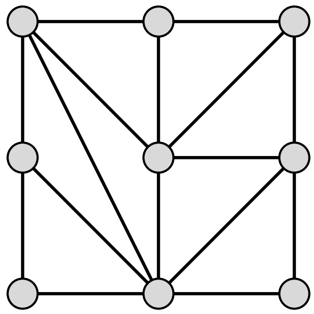
We are interested in algorithmically answering the following decision problems:

**Question 1.** *Given a lattice polytope $P$, does it have a (regular) unimodular*[1] *triangulation?*

In this software demonstration, we present the `UniTriSat` package, written in `Julia`, available at the GitHub repository `krhuang/UniTriSat`. To find unimodular triangulations, we encode simplices as boolean variables to translate this decision problem into a SAT problem. For *regular* unimodular triangulations, such triangulations can be checked for regularity via a linear program. The software also allows for finding and enumerating *all* (regular) unimodular triangulations.

*Kyle Huang is supported by the SPP 2458 "Combinatorial Synergies", funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 539867386

[1]We regularly write "(regular) unimodular" when the sentence can be read with and without "regular"



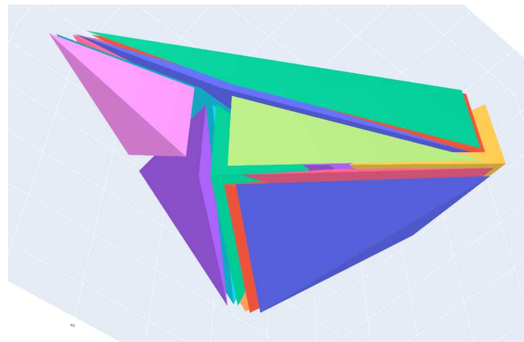


**Figure 1:** Two unimodular subdivisions of lattice polytopes. In dimension 2, every lattice polygon has a unimodular triangulation via Pick's Theorem. On the right, we visualize the smallest lattice polytope with a unimodular triangulation but no *regular* one. It has normalized volume 17 and two interior lattice points, and its coordinates are the columns of the matrix:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 2 & 1 & 2 \\ 0 & 0 & 2 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & -1 & -1 & 1 \end{bmatrix}$$

Previously, the smallest known example was produced by Ohsugi and Hibi via edge-polytopes and had dimension 9 [25]. We find many such lattice polytopes in dimension 3, 4, and 5; compare the last two columns of the tables in Appendix B.

## 1.1 Motivation

Unimodular triangulations are ubiquitous in toric geometry, tropical geometry, and combinatorics–having a unimodular triangulation is a "powerful" property in various senses.

A thorough survey of unimodular triangulations, their applications, and interesting open problems can be found at [18]. We attempt to give a small motivating taste. For toric geometers, unimodular triangulations correspond to crepant resolutions of the affine toric variety associated to the cone over the polytope. This desingularization is moreover projective if and only if the unimodular triangulation is regular; see the discussion in [18, 2]. Regular unimodular triangulations also give rise to Gröbner bases of toric ideals, with degree bounds that can be read off the combinatorics of the triangulation [18, Section 2.4].

In Ehrhart theory, Stanley showed that a unimodular triangulation, when it exists, gives a direct interpretation of the $h^*$-vector–it coincides with the $h$-vector of the unimodular triangulation [14]. Unimodular triangulations were also used in Igor Pak's combinatorial proof of the well-known hook-length formula [27], as well as Stanley's interpretation of Eulerian numbers as volumes of hypersimplices [14].

Unimodular triangulations are also an interesting property in their own right–active research has shown that certain classes of polytopes have regular unimodular triangulations [21, Lemma 2.4] [19, 20, 2], while the question remains an open conjecture for other classes of lattice polytopes, such as smooth polytopes and $s$-lecture-hall simplices [8].

## 1.2 Smooth Polytopes and Oda's Conjecture

We say a lattice polytope $P$ has the *Integer Decomposition Property (IDP)* if every lattice point in $kP$ can be written as a sum of $k$ lattice points of $P$, for all $k \in \mathbb{Z}_{\geq 1}$. It can be shown that having a (regular) unimodular triangulation implies the IDP–in fact they live far apart on a hierarchy of properties [17].

We say a lattice polytope is *smooth* if it is simple, and at every vertex the primitive edge directions form a lattice basis. The infamous Oda's conjecture, fast-approaching its third decade, concerns the relationship between smoothness and the IDP.

**Conjecture 1** (Oda's Conjecture [22, 24])**.** *Let $P$ be a smooth polytope. Then $P$ has the IDP.*

Some partial progress has been made on Oda's Conjecture: it is known for centrally symmetric polytopes in dimension three [4, 9], hollow smooth 3-polytopes via classification [12, Appendix A], and smooth, reflexive $d$-polytopes for $d \leq 7$ [28, 18]. In recent developments, Curtis presented a proof of Oda's Conjecture for combinatorial cubes at FPSAC 2025 [10].

However, a much-stronger conjecture, that smooth polytopes have a unimodular triangulation, or even a *regular* unimodular triangulation, remains open. Christian Haase and the first author recently produced a classification of all 12589 smooth 3-polytopes with $\leq$ 50 lattice points[2]. We used `UniTriSat` to verify that every smooth polytope in this database, as well as smooth polytopes with normalized volume $\leq 24, 20, 16$ in dimensions $4, 5, 6$, respectively (from Balletti's database [3]), have regular unimodular triangulations.

## 1.3 State of the Art

### 1.3.1 `TOPCOM`

The current state of the art for computing (regular) unimodular triangulations is `TOPCOM` (*Triangulations of Point Configurations and Oriented Matroids*) [30]. `TOPCOM` enumerates all triangulations by exploring the flip-graph of triangulations, and the flag `--unimodular` limits its output to unimodular triangulations.

[2]Not yet published, but the dataset is available at the GitHub repository `https://github.com/krhuang/SmoothGeneration`, under the filename "prune_output"

TOPCOM was used by Balletti to compute unimodular triangulations for his database, but reached prohibitively long runtimes. We extend these computations using UniTriSat–see the tables in Appendix B. It should be unsurprising that UniTriSat outperformed TOPCOM in finding a (regular) unimodular triangulation, as TOPCOM's intended use-case is enumerating *all* triangulations, a substantially harder problem. In fact, the --unimodular flag is a new feature not included in the original release of TOPCOM; see the remarks in the manual [31].

#### 1.3.2 polymake extension

To show that all smooth, reflexive polytopes with dimension $\leq 7$ have regular unimodular triangulations, Haase, Paffenholz, Piechnik, and Santos developed a heuristic project-and-lift approach, written as a polymake [1] extension[3]–a few cases were handled separately [18, 26]. However, we highlight that their heuristic procedure cannot prove the non-existence of a regular unimodular triangulation.

By passing regular=true to triangulate, users can use UniTriSat to conclusively check if a lattice polytope has a regular unimodular triangulation.

### 1.4 Final Remarks

UniTriSat is in active development, available at the GitHub repository krhuang/UniTriSat. The reader interested in using UniTriSat should see the QuickStart Guide in Appendix A. We look forward to any computational challenges that may arise, and hope you find UniTriSat useful to your research.

## 2 Algorithm Description

The algorithm builds on an integer programming approach by Firla and Ziegler [13] and relies on the extensive development of SAT-solvers [16, 23]. SAT (short for SATISFIABILITY) is a prototypical example of an NP-complete problem, with active development on several fast solvers.

Let $P$ be a $d$-dimensional lattice polytope and let $\mathcal{S}_P$ be the set of unimodular lattice simplices with vertices inside $P$:

$$\mathcal{S}_P = \{S \subset P \mid S \text{ is a unimodular lattice simplex}\}.$$

For a unimodular lattice simplex $F$ of dimension $d-1$ contained in the interior of $P$, let $\mathcal{S}_F$ be the set of unimodular simplices containing $F$ as a facet:

$$\mathcal{S}_F = \{S \in \mathcal{S}_P \mid F \text{ is a facet of } S, S \text{ is unimodular}\} \subseteq \mathcal{S}_P.$$

[3] Unfortunately, support for polymake extensions has been deprecated.

We wish to encode unimodular triangulations of $P$ as solutions to an appropriate boolean formula; a good manual is [29]. Consider the following SAT-formula $\Phi_P$ in conjunctive normal form (`AND` across clauses with only `OR`s), comprising of the following clauses on the variable set $\mathcal{S}_P$:

1. Intersection clauses: for $S \neq S' \in \mathcal{S}_P$, if $\mathrm{int}(S) \cap \mathrm{int}(S') \neq \emptyset$, we add the clause
$$\left(\neg S \ \vee \neg S'\right).$$
2. Face-covering clauses: For $S \in S_P$ and for every facet $F$ of $S$, if $F$ is not contained in the boundary of $P$, we add the clause
$$\left(\neg S \ \vee \bigvee_{S' \in \mathcal{S}_F \setminus \{S\}} S'\right).$$
3. Non-emptiness:
$$\bigvee_{S \in \mathcal{S}_P} S.$$

The non-emptiness clause can be replaced by a large `OR` over those simplices containing a sufficiently generic point. When combined with the intersection clauses, this `OR` can be recognized as an `XOR` in disguise, allowing us to split its solutions and parallelize the SAT-solver step–this is implemented via the `parallel_solving` flag.

We have the following Proposition as an easy consequence of [11, Corollary 4.5.20].

**Proposition 1.** *Given a lattice polytope $P$, the set of boolean solutions to $\Phi_P$ corresponds one to one to unimodular triangulations of $P$, where the boolean variables set to* `true` *correspond to unimodular simplices appearing in the triangulation.*

*Proof.* It's clear that a unimodular triangulation of $P$ gives rise to a solution to $\Phi_P$.

For the converse, let $P$ be a lattice polytope and $\mathcal{T}$ a set of simplices satisfying of $\Phi_P$. By the characterization [11, Corollary 4.5.20], we need to show

- *The Interior Covering Property*: For every interior facet $F$ of a simplex $S \in \mathcal{T}$, there are exactly two simplices $S, S'$ such that $F$ is a facet of $S, S'$ and defines a weakly separating hyperplane between $S, S'$
- *Interior Point Property*: There is a point $x \in P$ in general position contained in the convex hull of exactly one simplex

By the non-emptiness clause there exists at least one simplex $S \in \mathcal{T}$. For generic $x \in S$, it is contained in exactly one simplex of the triangulation $\mathcal{T}$ by the intersection clauses.

By the face-covering clause, for any interior facet $F$ of a simplex $S \in \mathcal{T}$ we must have another $S'$ with $F$ as a facet. Any two simplices on the same side of $F$ necessarily intersect on their interior, hence $F$ defines a weakly separating hyperplane between $S, S'$. □

Regularity of unimodular triangulations can then be checked via a simple linear program, with a real variable for each lattice point [11, Proposition 5.2.6].

# 3 Implementation

The algorithm is implemented in `UniTriSat` as the function `triangulate`. When given a polytope, the `triangulate` function follows the following pipeline:

1. Finding all lattice points in the convex hull of the given points.
2. Finding all unimodular simplices with vertices on those points.
3. Finding all internal $d-1$ dimensional simplices.

4a. Finding the set of pairs of unimodular simplices which intersect non-trivially and generating the associated clauses.

4b. Generating the face-covering clauses.

5. Invoking the chosen SAT-solver and passing the generated clauses.

Step 1 is done via precise rational arithmetic on polytopes, either using CDDLib [15] or Normaliz [7] as backends. The Normaliz backend is much faster, though we encountered some non-deterministic stability issues when using the Julia bindings.

Step 2 is done via a straightforward enumeration of simplices filtered by checking if the determinant of the difference matrix of lattice points has absolute value 1.

Step 3 is done by enumerating combinations of vertices and checking the boundary conditions derived from the defining hyperplanes of the polytope.

Step 4a often dominates the runtime of the algorithm. It is done by iterating over all pairs of simplices and checking if they intersect via the Separating Axis Theorem (see [32]). The candidate separating axes can be understood as normal vectors of the Minkowski sum $S + -S'$). To check whether a pair of simplices intersect via the Separating Axis Theorem, candidate separating axes are computed via the generalized cross-product before checking whether the 1-dimensional projections onto each span overlap.

Because this step dominates the runtime for some cases, special care was taken in the implementation to make this step as efficient as possible. We make heavy use of Julia meta-programming and type-aware JIT compilation to specialize the code to be fast for small dimensions and avoid excessive allocation.

Step 4b generates face-covering clauses by enumerating possible faces in the internal faces set generated in Step 3 and extracting covers which are supersets of those faces.

Step 5 is done with either `PicoSAT` [6] or `CaDiCaL` [5], depending on the user's choice. PicoSAT is readily available in the `Julia` ecosystem, whereas we provide a wrapper for `CaDiCaL`. As previously remarked, this step can also be parallelized by passing `parallel_solving=true` to `triangulate`.

To verify correctness, we validated our computations against independently published numbers for polytope datasets produced by Baletti, using `TOPCOM` [3, Appendix C]. We also provide an extensive test suite–see the QuickStart guide in Appendix A.

### 3.1 Robustness and Interoperability

We rely on external libraries for certain parts of the algorithm, even allowing for different implementations and backends for the same steps. For example, when the user does not have Normaliz available, a naive implementation via CDDLib is used as a fallback.

Polytopes can be constructed in Julia and fed to the exported `triangulate` function, which also provides for various optional auxiliary facilities such as logging, terminal output progress tracking, plotting. Users can also pass arguments to search for regular unimodular triangulations, output all triangulations, and parallelize the SAT-solver. For ease of use, the function is overloaded to allow the polytopes to be specified in various data formats, such as a matrix containing the polytope vertices as rows, a list of matrices, Julia-native Polyhedra objects, or a pathname to an on-disk file containing polytopes.

## Acknowledgements

We thank Christian Haase for the helpful discussions and enlightening pointers, as well as the endless supply of tea. We are also thankful to Yan Alves-Radke , Gabriele Balletti, Mónica Blanco, Giulia Codenotti, Anthony Della Vecchia, Michael Joswig, Niels Lindner, and Nikolas Rieke for their pointers and ideas. The first author is supported by the SPP 2458 "Combinatorial Synergies", funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 539867386.

## References

[1] B. Assarf, E. Gawrilow, K. Herr, M. Joswig, B. Lorenz, A. Paffenholz, and T. Rehn. “Computing convex hulls and counting integer points with polymake”. *Mathematical Programming Computation* **9** (2014), pp. 1–38. DOI.

[2] S. Backman and G. Liu. “A Regular Unimodular Triangulation of the Matroid Base Polytope”. *Séminaire Lotharingien de Combinatoire [electronic only]* **91B** (2024). Link.

[3] G. Balletti. “Enumeration of Lattice Polytopes by Their Volume”. *Discrete & Computational Geometry* **65**.4 (2020), pp. 1087–1122. DOI.

[4] M. Beck, C. Haase, A. Higashitani, J. Hofscheier, K. Jochemko, L. Katthän, and M. Michałek. "Smooth Centrally Symmetric Polytopes in Dimension 3 are IDP". *Annals of Combinatorics* **23**.2 (2019), pp. 255–262. DOI.

[5] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, and F. Pollitt. "CaDiCaL 2.0". *Computer Aided Verification*. Ed. by A. Gurfinkel and V. Ganesh. Cham: Springer Nature Switzerland, 2024, pp. 133–152.

[6] J. Bolewski, C. Lucibello, and M. Bouton. "PicoSAT.jl". PicoSAT.jl GitHub repository. Link.

[7] W. Bruns, B. Ichim, C. Söger, and U. von der Ohe. "Normaliz. Algorithms for rational cones and affine monoids". Link.

[8] J. B. Caicedo, M. Juhnke, and G. Poullot. "Ehrhart non-positivity and unimodular triangulations for classes of s-lecture hall simplices". 2025. arXiv:2508.18890.

[9] G. Codenotti and F. Santos. "Unimodular covers of 3-dimensional parallelepipeds and Cayley sums". *Combinatorial Theory* **3** (Dec. 2023). DOI.

[10] J. Curtis. "Smooth Combinatorial Cubes are IDP". Sept. 2025. arXiv:2509.02960.

[11] J. De Loera, J. Rambau, and F. Santos. *Triangulations. Structures for algorithms and applications*. Vol. 25. Jan. 2010. DOI.

[12] S. Di Rocco, C. Haase, B. Nill, and A. Paffenholz. "Polyhedral adjunction theory". *Algebra & Number Theory* **7**.10 (2013), pp. 2417–2446. DOI.

[13] R. T. Firla and G. M. Ziegler. "Hilbert Bases, Unimodular Triangulations, and Binary Covers of Rational Polyhedral Cones". *Discrete & Computational Geometry* **21** (Feb. 1999), pp. 205–216.

[14] D. Foata. "Distributions Euleriennes et Mahoniennes sur le Groupe des Permutations". *Higher Combinatorics*. Ed. by M. Aigner. Vol. 31. With a comment by Richard P. Stanley. Dordrecht: Springer Netherlands, 1977, pp. 27–49.

[15] K. Fukuda. "cddlib Reference Manual". Feb. 2021. Link.

[16] C. P. Gomes, H. Kautz, A. Sabharwal, and B. Selman. "Satisfiability Solvers". *Handbook of Knowledge Representation*. Ed. by F. van Harmelen, V. Lifschitz, and B. Porter. Vol. 3. Foundations of Artificial Intelligence. Amsterdam: Elsevier Science, 2008, pp. 89–134.

[17] C. Haase, T. Hibi, and D. Maclagan. "Mini-Workshop: Projective Normality of Smooth Toric Varieties". *Oberwolfach Report* **4** (2007). DOI.

[18] C. Haase, A. Paffenholz, L. Piechnik, and F. Santos. *Existence of unimodular triangulations - positive results*. Vol. 270. Memoirs of the American Mathematical Society. May 2014. DOI.

[19] G. Hamano. "Existence of a Regular Unimodular Triangulation of the Edge Polytopes of Finite Graphs". *COMMENTARII MATHEMATICI UNIVERSITATIS SANCTI PAULI* **65** (2 Sept. 2016). Link.

[20] T. Hibi, A. Higashitani, and K. Yoshida. "Existence of regular unimodular triangulations of dilated empty simplices". *European Journal of Combinatorics* **70** (2018), pp. 374–383. DOI.

[21] F. F. Knudsen. "Construction of nice polyhedral subdivisions". *Toroidal Embeddings I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1973, pp. 109–164. DOI.

[22] Mathematisches Forschungsinstitut Oberwolfach. "9744". *Workshop Reports* (1997). DOI.

[23] S. Möhle and A. Biere. "Backing Backtracking". *Theory and Applications of Satisfiability Testing – SAT 2019*. Ed. by M. Janota and I. Lynce. Cham: Springer International Publishing, 2019, pp. 250–266.

[24] T. Oda. "Problems on Minkowski sums of convex lattice polytopes". 2008. arXiv:0812.1418.

[25] H. Ohsugi and T. Hibi. "A Normal (0, 1)-Polytope None of Whose Regular Triangulations Is Unimodular". *Discrete & Computational Geometry* **21** (1999), pp. 201–204.

[26] A. Paffenholz. "Regular Unimodular Triangulations". Link.

[27] I. Pak. "Hook length formula and geometric combinatorics." *Séminaire Lotharingien de Combinatoire [electronic only]* **46** (2001), B46f, 13 p., electronic only–B46f, 13 p., electronic only. Link.

[28] L. Piechnik. "Smooth reflexive 4-polytopes have quadratic triangulations". *Undergraduate thesis, Duke University* (2004).

[29] S. Prestwich. "CNF Encodings". *Handbook of Satisfiability*. Ed. by A. Biere, M. Heule, H. van Maaren, and T. Walsh. 2nd ed. Vol. 336. Frontiers in Artificial Intelligence and Applications. Apr. 2021, pp. 75–100.

[30] J. Rambau. "TOPCOM: Triangulations of Point Configurations and Oriented Matroids". *Proceedings of the International Congress of Mathematical Software*. 2002. Link.

[31] J. Rambau. "TOPCOM User Manual, version 1.1.0". Sept. 2022. Link.

[32] J. Renders. "Proof of Separating Axis Theorem for polygons". Mathematics Stack Exchange. 2017. Link.

# A QuickStart Guide

First install `Julia`. To run the code on polytope data (e.g. from Baletti's database) run

```
git clone https://github.com/gabrieleballetti/small-lattice-polytopes
```

in a terminal. Then, in a `Julia` terminal, run the commands

```
using Pkg; Pkg.add(url="https://github.com/krhuang/UniTriSat/")

using UniTriSat

triangulate("small-lattice-polytopes/data/3-polytopes/v6.txt")
```

to find unimodular triangulations of lattice polytopes in Balletti's database.

To run the test suite, do `Pkg.test("UniTriSat")`.

The following example finds a unimodular triangulation of the 5-dimensional cross-polytope:

```
using UniTriSat

P = [ 1 0  0 0 0; -1 0 0 0 0; 0 1 0  0 0; 0 -1 0 0 0; 0 0 1 0  0;
      0 0 -1 0 0;  0 0 0 1 0; 0 0 0 -1 0; 0  0 0 0 1; 0 0 0 0 -1]

triangulate(P, terminal_output="running,table,final")
```

Running these commands yields the following summary:

```
----------------------------------------
Run Summary
----------------------------------------
Total Polytopes Processed:     1
Regularly Triangulatable:      0
Triangulatable:                1
Non-Triangulatable:            0
Total Run Time:                00:00:00
----------------------------------------
Step Name                           |    Time      |     Memory   |
-------------------------------------------------------------------
Compute all lattice points          | 0.012 s      | 4.94 MiB     |
Compute unimodular simplices        | 0.000 s      | 120.58 KiB   |
Compute internal faces              | 0.002 s      | 757.18 KiB   |
Compute intersecting pairs          | 0.001 s      | 582.37 KiB   |
Generate face-covering clauses      | 0.001 s      | 160.94 KiB   |
Solve SAT problem                   | 0.000 s      | 784 B        |
```

`triangulate` takes many optional input parameters controlling its functionality; see the GitHub repository for documentation.

# B Computational Results

Passing Balletti's database of lattice polytopes yielded the following tables. They should be compared with the tables in [3, Appendix C]–new entries are in **bold**.

| vol | total | IDP | UT | RUT |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | **1** |
| 2 | 3 | 2 | 2 | **2** |
| 3 | 6 | 5 | 5 | **5** |
| 4 | 17 | 14 | 14 | **14** |
| 5 | 19 | 15 | 15 | **15** |
| 6 | 54 | 43 | 43 | **43** |
| 7 | 59 | 47 | 47 | **47** |
| 8 | 154 | 125 | 125 | **125** |
| 9 | 181 | 135 | 135 | **135** |
| 10 | 368 | 290 | 290 | **290** |
| 11 | 414 | 323 | 323 | **323** |
| 12 | 961 | 746 | 745 | **745** |
| 13 | 1029 | 779 | 778 | **778** |
| 14 | 1929 | 1506 | 1506 | **1506** |
| 15 | 2409 | 1837 | 1835 | **1835** |
| 16 | 4254 | 3292 | 3288 | **3288** |
| 17 | 4983 | 3787 | **3784** | **3783** |
| 18 | 8586 | 6635 | **6627** | **6626** |
| 19 | 10186 | 7782 | **7771** | **7769** |
| 20 | 16708 | 12971 | **12957** | **12954** |
| 21 | 20487 | 15579 | **15551** | **15547** |
| 22 | 31163 | 24085 | **24055** | **24047** |
| 23 | 37779 | 29171 | **29140** | **29128** |
| 24 | 58906 | 45663 | **45588** | **45569** |
| 25 | 70057 | 53726 | **53634** | **53608** |
| 26 | 103117 | 80225 | **80121** | **80088** |
| 27 | 126507 | 97349 | **97197** | **97123** |
| 28 | 181732 | 141488 | **141294** | **141213** |
| 29 | 219325 | 169816 | **169549** | **169447** |
| 30 | 311917 | 242984 | **242630** | **242443** |
| 31 | 376303 | 291956 | **291526** | **291292** |
| 32 | 522559 | 408010 | **407435** | **407165** |
| 33 | 636394 | 494067 | **493353** | **492906** |
| 34 | 860937 | 673321 | **672389** | **671824** |
| 35 | 1043226 | 814161 | **813049** | **812353** |
| 36 | 1411304 | 1104038 | **1102568** | **1101557** |

**Figure 2:** Lattice polytope properties in dimension 3. The example in Figure 1 can be seen in row 17.

| vol | total | IDP | UT | RUT |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | **1** |
| 2 | 3 | 2 | 2 | **2** |
| 3 | 8 | 6 | 6 | **6** |
| 4 | 28 | 19 | 19 | **19** |
| 5 | 31 | 21 | 21 | **21** |
| 6 | 109 | 71 | 71 | **71** |
| 7 | 113 | 74 | 74 | **74** |
| 8 | 391 | 242 | 242 | **242** |
| 9 | 438 | 255 | 255 | **255** |
| 10 | 1019 | 618 | 618 | **618** |
| 11 | 1109 | 664 | 664 | **664** |
| 12 | 3251 | 1850 | 1849 | **1849** |
| 13 | 3123 | 1761 | 1760 | **1760** |
| 14 | 6863 | 3918 | **3918** | **3918** |
| 15 | 8506 | 4560 | **4558** | **4558** |
| 16 | 17309 | 9500 | **9494** | **9494** |
| 17 | 18861 | 10066 | **10062** | **10062** |
| 18 | 38061 | 20125 | **20115** | **20114** |
| 19 | 42067 | 21997 | **21981** | **21979** |
| 20 | 80578 | 42253 | **42227** | **42226** |
| 21 | 94373 | 47214 | **47171** | **47170** |
| 22 | 158030 | 81501 | **81454** | **81449** |
| 23 | 184646 | 92429 | **92371** | **92362** |
| 24 | 330776 | 165631 | **165490** | **165471** |

**Figure 3:** Lattice polytope properties in dimension 4. Entries in bold are newly computed.

| vol | total | IDP | UT | RUT |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | **1** |
| 2 | 4 | 2 | 2 | **2** |
| 3 | 10 | 6 | 6 | **6** |
| 4 | 38 | 21 | 21 | **21** |
| 5 | 42 | 25 | 25 | **25** |
| 6 | 169 | 86 | 86 | **86** |
| 7 | 163 | 90 | 90 | **90** |
| 8 | 659 | 322 | 322 | **322** |
| 9 | 707 | 344 | 344 | **344** |
| 10 | 1737 | 841 | 841 | **841** |
| 11 | 1743 | 869 | 869 | **869** |
| 12 | 6294 | 2791 | 2790 | **2790** |
| 13 | 5101 | 2392 | **2391** | **2391** |
| 14 | 12640 | 5756 | **5756** | **5756** |
| 15 | 15373 | 6655 | **6653** | **6653** |
| 16 | 34637 | 14870 | **14864** | **14864** |
| 17 | 32858 | 14314 | **14311** | **14311** |
| 18 | 77727 | 32160 | **32149** | **32149** |
| 19 | 75401 | 32272 | **32255** | **32254** |
| 20 | 167969 | 68488 | **68456** | **68454** |

**Table 1:** Lattice polytope properties in dimension 5. Entries in bold are newly computed.

| vol | total | IDP | UT | RUT |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | **1** |
| 2 | 4 | 2 | 2 | **2** |
| 3 | 11 | 6 | 6 | **6** |
| 4 | 48 | 22 | 22 | **22** |
| 5 | 51 | 27 | 27 | **27** |
| 6 | 228 | 94 | 94 | **94** |
| 7 | 204 | 97 | 97 | **97** |
| 8 | 961 | 362 | 362 | **362** |
| 9 | 970 | 392 | 392 | **392** |
| 10 | 2444 | 959 | 959 | **959** |
| 11 | 2249 | 964 | 964 | **964** |
| 12 | 9872 | 3362 | **3361** | **3361** |
| 13 | 6622 | 2676 | **2675** | **2675** |
| 14 | 18069 | 6684 | **6684** | **6684** |
| 15 | 21837 | 7828 | **7826** | **7826** |
| 16 | 53513 | 18005 | **17999** | **17999** |

**Table 2:** Lattice polytope properties in dimension 6. Entries in bold are newly computed.