VOLUME COMPUTATION FOR POLYTOPES: STRATEGIES AND PERFORMANCES

Polytopes and volume. A *convex polytope* P in dimension d can be described in two ways: First, it is the convex hull of a finite set of points $\{v_1, \ldots, v_n\} \subseteq \mathbf{R}^d$. Second, it is the bounded intersection of a finite number of half spaces, i.e. given as $\{x : Ax \leq b\}$ for a matrix $A \in \mathbf{R}^{m \times d}$ and a vector $b \in \mathbf{R}^m$. In the first case, the polytope is given by its *$\mathcal{V}$-representation*, in the second case by its *$\mathcal{H}$-representation*.

Only full dimensional polytopes, i.e. polytopes which are not contained in any hyperplane of $\mathbf{R}^d$, are of interest in the context of volume computation. If P is such a polytope, then its volume $\mathrm{Vol}(P) = \mathrm{Vol}_d(P)$, as usual defined by the d-dimensional Lebesgue measure, is not zero. A minimal $\mathcal{V}$-representation of P is unique and given by the vertices of P. Also a minimal $\mathcal{H}$-representation is unique (up to multiplication of rows of A and the corresponding entries of b by positive scalars), and the sets $P \cap \{x : a_i x = b\}$, where a_i is the i-th row of A in such a minimal representation, define the facets of P.

There are numerous applications of polytope volume computation, ranging from estimating the size of the solution space of a linear program to counting the number of roots of a system of complex polynomial equations. To date, most practical applications concern low dimensions. This is partially due to the fact that the complexity of volume computation algorithms increases exponentially when the dimension grows, partially to limited experience with these algorithms.

This contribution is concerned with exact algorithms for computing polytope volumes which have been implemented successfully. Randomized approximation algorithms as described in [9] are omitted from the discussion because so far they have not been programmed.

Complexity results. Simple examples show that the minimal $\mathcal{V}$-representation of a polytope can have exponential size with respect to its minimal $\mathcal{H}$-representation, and vice versa. Hypercubes, for instance, have $2d$ facets and 2^d vertices in dimension d, and the converse holds for their duals, the *cross polytopes* (see below for more about polytope duality). Hence it is not surprising that the complexity of volume computation depends on the polytope representation. The complexity results of this section are valid for polytopes represented over $\mathbf{Z}$ or $\mathbf{Q}$ and are formulated with respect to the bit length of the input. Detailed proofs can be found in [7].

For some classes of polytopes there are polynomial time algorithms computing their volumes. An important such class is formed by the polytopes in fixed dimension, given either in $\mathcal{H}$- or $\mathcal{V}$-representation, which reflects that geometric 2- or 3-dimensional problems can be solved efficiently in practice.

Moreover, there are polynomial algorithms for polytopes P satisfying the following condition: P is given in $\mathcal{H}$-representation, and there is a constant δ such that each facet of P contains at most δ vertices (the *near-simplicial* case). Or, P is given in $\mathcal{V}$-representation, and there is a constant δ such that each vertex of P is contained in at most δ facets (the *near-simple* case). In particular, the case $\delta = d$ covers *simplicial* $\mathcal{H}$- and *simple* $\mathcal{V}$-polytopes. However, this result seems to be of little practical value. For instance, simple polytopes are usually given by their $\mathcal{H}$-representation, and their $\mathcal{V}$-representation already tends to be exponential. Examples for this phenomenon are provided by hypercubes or polytopes constructed from randomly chosen halfspaces.

convex polytope→ *convex polytope*
$\mathcal{V}$-representation→ *$\mathcal{V}$-representation*
$\mathcal{H}$-representation→ *$\mathcal{H}$-representation*
cross polytopes→ *cross polytope*
near-simplicial→ *near-simpliciality*
near-simple→ *near-simplicity*
simplicial→ *simpliciality*
simple→ *simplicity*

All known algorithms are exponential in d resp. δ. This is inevitable for $\mathcal{H}$-polytopes, since the binary size of the volume of a polytope may be exponential in its $\mathcal{H}$-representation. Some of the triangulation methods described below (boundary triangulation, for instance) show that the volume of a polytope has polynomial size with respect to the $\mathcal{V}$-representation. However, even in this case volume computation is not an easy task. Indeed, the problem is #P-hard for polytopes in either $\mathcal{H}$- or $\mathcal{V}$-representation (see [5]).

The complexity of the volume computation problem is unknown when both representations are given. This problem might even be solvable in polynomial time with one of the known algorithms.

Basic approaches and duality. All deterministic volume computation methods decompose a given polytope into polytopes whose volumes are easier to compute. Especially apt for volume computation purposes are *simplices*, i.e. d-dimensional polytopes with $d+1$ vertices. Their volume is given by a determinant, precisely,

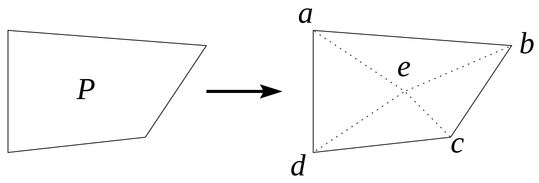
$$\mathrm{Vol}(\mathrm{conv}(\{v_0,\ldots,v_d\})) = \frac{|\det(v_1-v_0,\ldots,v_d-v_0)|}{d!}.$$

Depending on the decomposition into simplices, two basic classes of algorithms can be distinguished.

Triangulations. A *triangulation* of a polytope P is a finite collection $\{\Delta_i : i=1,\ldots,s\}$ of simplices such that $P=\bigcup_{i=1}^{s}\Delta_i$ and the intersection of any two simplices is their common face. (This is the usual definition of a triangulation; in fact, for volume computation purposes it would be enough to demand that no two simplices share an interior point.) Then clearly

$$\mathrm{Vol}(P) = \sum_{i=1}^{s}\mathrm{Vol}(\Delta_i).$$

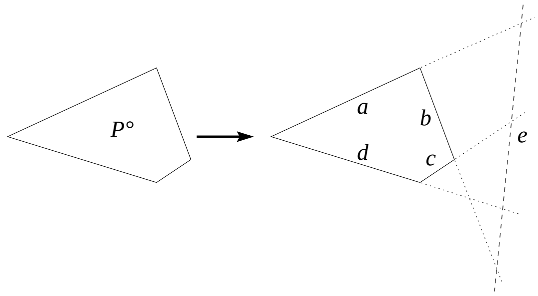
For example, if P is simplicial, i.e., all its facets are $d-1$-dimensional simplices, then linking these facets to a fixed interior point as illustrated in the figure yields a triangulation of the whole polytope, called *boundary triangulation*. Other algorithms are presented below.



Signed decompositions. The restriction that no two simplices may share an interior point is in fact not necessary. If it is violated, then some parts of the polytope are covered more than once, which must be corrected by subtracting their respective volumes. A *signed decomposition* of P is thus a finite collection $\{(\Delta_i,\sigma_i) : i=1,\ldots,s\}$ of simplices Δ_i and signs $\sigma_i\in\{\pm1\}$ with the following property: If a point of P does not lie on the boundary of any simplex, then the number of positive simplices containing it exceeds the number of negative simplices containing it by 1. A point outside P is contained in as many positive as negative simplices. It follows that

$$\mathrm{Vol}(P) = \sum_{i=1}^{s}\sigma_i\,\mathrm{Vol}(\Delta_i).$$

An example for a signed decomposition is given by `J. Lawrence`'s algorithm [11], see the figure. Let P be simple, and introduce an additional hyperplane e not parallel to any edge of the polytope. The signed decomposition consists of one simplex for each vertex, which is formed by e and the hyperplanes incident with this vertex. How the sign is determined will be described below. In the figure, the signed decomposition is $\{(ade,+),(abe,-),(cde,-),(bce,+)\}$.



simplices→ *simplex*
triangulation→ *triangulation*
boundary triangulation→ *boundary triangulation*
signed decomposition→ *signed decomposition*
`J. Lawrence`

In fact, triangulations are special cases of signed decompositions with all signs being +1. Due to duality results a distinction between these two classes is reasonable nevertheless.

Duality. Assume that the origin lies in the interior of P, which may be obtained by a translation into any interior point. Then the *polar dual* of P is defined to be $P^\circ = \left\{x \in \mathbf{R}^d : y^T x \leq 1 \text{ for all vertices } y \text{ of } P\right\}$. Hence the vertices of P are in bijection with the facets of P°, and the converse also holds. This means that the $\mathcal{V}$-representation of P corresponds to the $\mathcal{H}$-representation of P° (with the right hand side b normalized to the all one vector), and vice versa. For instance, the polytopes of the two figures above are dual to each other.

Polarity reverses inclusion, so that the duals of simple polytopes are simplicial and vice versa, and the combinatorial structure of P° depends only on the combinatorial structure of P. The exact geometry of P° and its volume, however, depend on the location of the origin in P. A result due to P. Filliman [6] shows that to each triangulation of P corresponds a signed decomposition of P°, if the origin is located conveniently. Precisely, let $\mathcal{T}$ be a triangulation of P such that the origin does not lie on any hyperplane spanned by the vertices of the triangulation. Then the following procedure yields a signed decomposition of P°: For a simplex $\Delta \in \mathcal{T}$ let its *separation number* s be the number of its facets separating it from the origin. Let Δ° be the simplex bounded by the hyperplanes corresponding by polarity to the vertices of Δ, and let its sign be $\sigma = (-1)^s$.

In this sense, Lawrence's decomposition of P° in the second figure is induced by the boundary triangulation of P in the first figure.

Suppose now that the complexity of a decomposition is measured by the number of simplices it generates (which is a simplified model since it does not take the sizes of the occurring numbers into account). Let $\mathcal{P}$ be a class of polytopes containing the origin in their interiors, and let $\mathcal{P}^\circ$ be composed of the duals of the elements in $\mathcal{P}$. Then for each triangulation algorithm on $\mathcal{P}$ (working with the $\mathcal{V}$-, $\mathcal{H}$- or both representations), there is a signed decomposition algorithm on $\mathcal{P}^\circ$ (working with the $\mathcal{H}$-, $\mathcal{V}$- or both representations) with the same complexity. This explains why most of the complexity results above are symmetric with respect to duality, and that asymmetry may only occur when the binary lengths of the numbers involved are taken into account.

Algorithms. In this section, some volume computation algorithms are presented in more detail. While the following list is not exhaustive, all of the described methods have been implemented and tested on different examples (see [3]).

Delaunay triangulation. The classical *Delaunay triangulation* is obtained by lifting the polytope vertices onto the surface of a $d+1$-dimensional convex body. Precisely, let $f : \mathbf{R}^d \to \mathbf{R}$ be a strictly convex function. (Traditionally, $f(x) = \sum_{i=1}^d x_i^2$.) Lift each vertex v to $(v, f(v))$. Then the lifted vertices will usually be in *general position*, i.e., their convex hull will be simplicial. Interpreting these simplicial facets in terms of the original vertices yields a triangulation. If the lifted points are not in general position — which is frequent for the traditional lifting function on 0-1-polytopes — then perturbation or the choice of a different lift resolves the problem. The algorithm needs the $\mathcal{V}$-representation. Its time and space complexity and numerical behavior depend on the convex hull algorithm used for computing the facets of the lifted polytope.

Boundary triangulation. The algorithm for simplicial polytopes has been described above. If P is not simplicial itself, then perturbation of its vertices leads to a simplicial polytope, whose facets can be determined by any convex hull algorithm. Identifying the perturbed vertices with the original ones leads to a triangulation of P.

polar dual→ *polar duality*
P. Filliman
separation number→ *separation number*
Delaunay triangulation→ *Delaunay triangulation*
general position→ *general position*

Again, only the $\mathcal{V}$-representation is needed, and the behavior of the algorithm is determined by the underlying method for computing the convex hull.

Triangulation by Cohen and Hickey. An obvious improvement of any boundary triangulation is obtained by using a vertex instead of an interior point, thus dropping all simplices containing this vertex. If in the first figure above, for instance, e were translated into a, then only the two simplices abc and acd would be left over. Together with a very efficient scheme of handling non-simpliciality, this observation is the basis of the triangulation algorithm described by `J. Cohen` and `T. Hickey` [4]. Suppose that $e_1, \ldots, e_m$ are the facets of P, given by the sets of vertices they contain. (This requirement is equivalent to the knowledge of both the $\mathcal{V}$- and the $\mathcal{H}$-representation of P.) Fix a vertex $v(P)$. Then the set $\{\mathrm{conv}(v(P), e_i) : v(P) \notin e_i\}$ is a decomposition of P into pyramids with bases e_i and apex v. If the polytope is simplicial, then the pyramids are in fact simplices. Otherwise their bases are triangulated recursively, and the resulting $d-1$-dimensional simplices, together with the fixed vertex $v(P)$, form a triangulation of P. More precisely, let b be such a base, again represented by a set of vertices. Fix a vertex $v(b) \in b$. As above, $\{\mathrm{conv}(v(b), e) : v(b) \notin e\}$, where e varies over the facets of b, i.e. the $\dim b - 1$-dimensional faces of P contained in b, forms a decomposition of b into pyramids. The recursion continues with the e's until a simplicial face is reached, which is the case at the latest in dimension 1. While the e's are not known a priori for $b \neq P$, they are exactly the $\dim b - 1$-dimensional sets within $\{b \cap e_i : v(b) \notin e_i\}$. Instead of testing the dimensions of these sets, which turns out to be too costly in practice, one can just continue the algorithm with all of them. If the dimension of $b \cap e_i$ is lower than needed, then this recursion branch will end prematurely with an empty face and not contribute a simplex to the triangulation.

Since Cohen and Hickey's triangulation scheme is purely combinatorial — it uses only incidence information and not the vertex coordinates to obtain a triangulation — it raises no numerical problems. Storing faces as sets of their vertices facilitates the detection of simplicial faces, as a face in (supposed) dimension d' is simplicial if and only if it has $d'+1$ vertices. Similarly, faces with supposed dimension d' and less than $d'+1$ vertices can be dropped, as their dimension is in fact lower than d'.

Lawrence's signed decomposition. Assume first that P is simple. To apply the scheme presented in the section about signed decompositions, it must be known how to compute the volume of the simplex associated with a vertex v of P. `J. Lawrence` [11] derives the following formula. Let the additional hyperplane be $\{x : c^T x = 0\}$, where $x \mapsto c^T x$ is non-constant on any edge of P. Denote by A_v the $d \times d$-matrix corresponding to the facets containing v. (This matrix can be derived if both the $\mathcal{H}$- and the $\mathcal{V}$-representation of P are known.) Then A_v is invertible, $\gamma^v = (A_v^T)^{-1} c$ is well defined up to permutations of its entries, and none of its entries is zero. Finally,

$$\mathrm{Vol}(P) = \sum_v \frac{(c^T v)^d}{d! \, |\det A_v| \prod_{i=1}^d \gamma_i^v}.$$

If P is not simple, then a vertex v is contained in more than d facets, and A_v is not square any more. In this case one perturbs the $\mathcal{H}$-representation lexicographically and runs a vertex enumeration algorithm. This results in possibly several new simple vertices for each vertex of the original polytope. Each new vertex corresponds to a lexicographically feasible cobasis A_v of the original vertex and contributes a term to the sum above. The same procedure must be applied if only the $\mathcal{H}$-representation of P is known.

An unresolved problem is the choice of the additional hyperplane, i.e. of c. For rational polytopes it is possible to choose c of size polynomial in the $\mathcal{H}$-representation, so that each term of

---

`J. Cohen`
`T. Hickey`
`J. Lawrence`

the sum has polynomial size. However, the non-trivial denominators often result in prohibitively large numbers after a few terms are summed up. On the other hand, using floating point arithmetic often leads to severe numerical instabilities when the denominator is close to zero in some vertices, so that numbers of drastically different magnitudes are summed up.

Lasserre's signed decomposition. Like Cohen and Hickey's triangulation, `J.B. Lasserre`'s algorithm [10] is based on a decomposition of the polytope into pyramids and proceeds by recursion. However, it uses only the $\mathcal{H}$-representation. Let a_i be the i-th row of A, b_i the i-th entry of b and $e_i = \{x \in P : a_i x = b_i\}$. Suppose that if e_i is parallel to e_j for some $i \neq j$, then b_i and b_j have different signs. (This is no restriction, since otherwise one of them is redundant, but has to be tested in an implementation.) Then

$$\mathrm{Vol}(P) = \frac{1}{d} \sum_{i=1}^{m} \frac{b_i}{\|a_i\|} \mathrm{Vol}_{d-1}(e_i),$$

where $|b_i| \,/\, \|a_i\|$ is the distance of the origin from e_i. A negative sign occurs for negative b_i, i.e., when the hyperplane corresponding to e_i separates the origin from the polytope. This formula does not yet allow a recursive implementation, since the $d-1$-dimensional volumes of facets embedded into d-dimensional space are needed. Thus e_i is projected onto a suitable subspace of $d-1$ coordinates. Let $a_{ij} \neq 0$. Substituting $x_j = (b_i - \sum_{k \neq j} a_{ik} x_i)/a_{ij}$ in the system of linear inequalities $Ax \leq b$ yields a new system $A^{(i)} x^{(i)} \leq b^{(i)}$ with $m-1$ inequalities in $d-1$ variables, which describes the projection of e_i onto the coordinate subspace $\{x : x_j = 0\}$. Taking the distorting effect of this projection into account, the following formula, which can be implemented recursively, is derived:

$$\mathrm{Vol}_d(P) = \frac{1}{d} \sum_{i=1}^{m} \frac{b_i}{a_{ij}} \mathrm{Vol}_{d-1}\left(\left\{x^{(i)} : A^{(i)} x^{(i)} \leq b^{(i)}\right\}\right).$$

In the same way as Cohen and Hickey's triangulation improves on boundary triangulation, the computational effort for Lasserre's algorithm can be reduced by translating each intermediate polytope into one of its vertices. In fact, the algorithm then implicitly constructs Cohen and Hickey's triangulation, with the determinant computation spread over the recursion tree. Since finding a vertex is a rather costly linear programming step, one instead translates into a (possibly infeasible) basis solution, which preserves the character of a signed decomposition.

Noticing that the same face may be considered at several places in the recursion tree, the efficiency of the algorithm can be increased by storing the volume of a face when it is computed for the first time, and retrieving the volume when a face reappears. Some care must be taken, however, when a face is projected onto different coordinate subspaces; then a determinant computation is required. For details, see [3].

Unlike in Cohen and Hickey's triangulation algorithm, there is no cheap way of testing whether an intermediate face is empty or of a too low dimension.

Hybrid orthonormalization technique. In [3] an algorithm is described which combines the advantages of Cohen and Hickey's and Lasserre's methods. On one hand, it exploits the information of both the $\mathcal{H}$- and the $\mathcal{V}$-representation, on the other hand, it stores and retrieves intermediate results.

Consider again the dissection into pyramids $\{\mathrm{conv}(v(P), e_i) : v(P) \notin e_i\}$ of Cohen and Hickey's triangulation algorithm, where as before the e_i's are given as the sets of their vertices. The volume formula for pyramids yields

$$\mathrm{Vol}_d(P) = \sum_{v(P) \notin e_i} \frac{1}{d} \,\mathrm{dist}(v(P), \mathrm{aff}(e_i))\, \mathrm{Vol}_{d-1}(e_i),$$

where $\mathrm{dist}(v(P), \mathrm{aff}(e_i))$ denotes the distance of the pyramid apex to the affine subspace corresponding to the pyramid base. Suppose that $\mathrm{Vol}_{d-1}(e_i)$ and on orthonormal basis of the linear subspace associated with e_i are known. Then the required distance as well as an orthonormal basis of the linear subspace corresponding to the pyramid with basis e_i can be computed easily.

J.B. Lasserre

Since the volume and an orthonormal basis of a one-dimensional polytope are trivial to determine, a recursive approach analogous to Cohen and Hickey's algorithm can be adopted.

So far, there is no advantage over the triangulation scheme. (In fact, the overhead of basis computation makes this algorithm slower in practice.) However, the strategy of storing and reusing intermediate results, which has been successfully employed to accelerate Lasserre's algorithm, is applicable now. Volumes and orthonormal bases of already visited faces can be stored and retrieved as soon as the face is reconsidered. As the bases require an enormous amount of storage space, it is reasonable to store only the face volumes and to recompute orthonormal bases from scratch when necessary.

The algorithm requires both the $\mathcal{H}$- and the $\mathcal{V}$-representation. Unlike all other methods presented above, it can only be implemented using floating point arithmetic and not rational arithmetic because orthonormalization involves square roots. Care must be taken to choose a numerically stable orthonormalization technique, e.g. using Householder transformations.

Choosing an algorithm. Experience with volume computation in higher dimensions is very limited; to date, only one study has been published [3]. Hence, while the following recommendations reflect the state of the art, they may soon be affected by algorithmic progress or new experimental results.

Low dimension. In accordance with the theoretical complexity results, volume computation is a simple task in practice if one restricts oneself to low dimensions (up to about 5). An algorithm should be chosen which works with the representation in which the polytopes are given.

Near-simple and near-simplicial polytopes. It can be observed that in practice triangulation methods behave particularly well on near-simplicial $\mathcal{V}$-polytopes, and that signed decomposition methods behave particularly well on near-simple $\mathcal{H}$-polytopes. This is in accordance with the general duality result above, but at first sight surprising when compared to the complexity result stating that these problems are polynomial when the polytopes are given by their converse representations. However, the polynomial complexity is obtained by solving a large number of linear programs, which is apparently not competitive for problems of a tractable size.

Lawrence's method, which generates a signed decomposition with especially few simplices for near-simple $\mathcal{H}$-polytopes, suffers from numerical instabilities as outlined above, so that Lasserre's algorithm is preferable in general.

Double representation. If both representations are known, then the hybrid orthonormalization technique proves to be the most efficient algorithm in practice. Although it is closer in spirit to a triangulation method, it is usually faster even than signed decomposition algorithms on near-simple polytopes since it efficiently exploits the additional structural information from the $\mathcal{V}$-representation.

Representation conversion. The cases left over as difficult are those of polytopes given by the "wrong" representation. An experimental finding is that $\mathcal{V}$-polytopes with a large ratio n/m of vertices to facets and $\mathcal{H}$-polytopes with a small ratio n/m pose problems. Unfortunately, most algorithms for converting between the representations face an "easy" and a "hard" direction, and exactly the hard direction is needed here. It has been observed, however, that on suitable classes of polytopes both directions have essentially the same complexity; the result is obtained using the easy transformation as an oracle for the hard one [2]. This technique will probably allow to efficiently apply the hybrid orthonormalization technique to those polytopes whose volumes are particularly hard to compute today.

References

[1] Bisztriczky, T., McMullen, P., Schneider, R., and Weiss, A. Ivić (eds.): *POLYTOPES: Abstract, Convex and Computational* (Dordrecht, 1994), Kluwer Academic Publishers.

[2] Bremner, D., Fukuda, K., and Marzetta, A.: 'Primal-Dual Methods for Vertex and Facet Enumeration', *Discrete & Computational Geometry* **20** (1998), 333–357.

[3] Büeler, Benno, Enge, Andreas, and Fukuda, Komei: 'Exact Volume Computation for Polytopes: A Practical Study': *[8]*, 1999.

[4] Cohen, Jacques, and Hickey, Timothy: 'Two Algorithms for Determining Volumes of Convex

Polyhedra', *Journal of the ACM* **26**, no. 3 (Mar. 1979), 401–414.

[5] Dyer, M. E., and Frieze, A. M.: 'On the Complexity of Computing the Volume of a Polyhedron', *SIAM Journal on Computing* **17**, no. 5 (Oct. 1988), 967–974.

[6] Filliman, P.: 'The volume of duals and sections of polytopes', *Mathematika* **39** (1992), 67–80.

[7] Gritzmann, Peter, and Klee, Victor: 'On the complexity of some basic problems in Computational Convexity: II. Volume and mixed volumes': *[1]*, 1994, pp. 373–466.

[8] Kalai, Gil, and Ziegler, Günter M. (eds.): *Polytopes — Combinatorics and Computation* (1999), DMV-Seminars, Birkhäuser Verlag.

[9] Kannan, R., Lovász, L., and Simonovits, M.: 'Random walks and an $O^*(n^5)$ volume algorithm for convex bodies', *Random Struct. Algorithms* **11**, no. 1 (1997), 1–50.

[10] Lasserre, J. B.: 'An Analytical Expression and an Algorithm for the Volume of a Convex Polyhedron in $\mathbb{R}^n$', *Journal of Optimization Theory and Applications* **39**, no. 3 (Mar. 1983), 363–377.

[11] Lawrence, Jim: 'Polytope Volume Computation', *Mathematics of Computation* **57**, no. 195 (July 1991), 259–271.

Andreas Enge
Lehrstuhl für Diskrete Mathematik,
Optimierung und Operations Research
Institut für Mathematik
Universität Augsburg, Deutschland
E-mail address: enge@math.uni-augsburg.de

AMS 1991 Subject Classification: 52B11, 52B45, 52B55.

Key words and phrases: volume, convex polytopes, algorithms.