Volume Computation Using a Direct Monte Carlo Method*

Sheng Liu[1,2], Jian Zhang[1], and Binhai Zhu[3]

[1] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
{lius,zj}@ios.ac.cn
[2] Graduate School, Chinese Academy of Sciences, Beijing 100049, China
[3] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA
bhz@cs.montana.edu

Abstract. Volume computation is a traditional, extremely hard but highly demanding task. It has been widely studied and many interesting theoretical results are obtained in recent years. But very little attention is paid to put theory into use in practice. On the other hand, applications emerging in computer science and other fields require practically effective methods to compute/estimate volume. This paper presents a practical Monte Carlo sampling algorithm on volume computation/estimation and a corresponding prototype tool is implemented. Preliminary experimental results on lower dimensional instances show a good approximation of volume computation for both convex and non-convex cases. While there is no theoretical performance guarantee, the method itself even works for the case when there is only a membership oracle, which tells whether a point is inside the geometric body or not, and no description of the actual geometric body is given.

1 Introduction

Volume computation is a highly demanding task in software engineering, computer graphics, economics, computational complexity analysis, linear systems modeling, VLSI design, statistics, etc. It has been studied intensively and lots of progress has been made especially in recent decades [7,2,14,13]. However, so far most of the research work is only concerned with the computational complexity aspect. For instance, some researchers tried to obtain a theoretical lower bound at all costs and neglected the practical feasibility of their algorithms. Therefore, although there are some strong theoretical results on this problem, little progress has been made in putting them into practical use. For this reason, this paper focuses on practically usable tools on computing/estimating the volume of a body.

* This work is partially supported by the National Natural Science Foundation (NSFC) under grant number 60673044 and 60633010, and by Montana EPSCOR Visiting Scholar's Program.

G. Lin (Ed.): COCOON 2007, LNCS 4598, pp. 198–209, 2007.

Generally speaking, we want to compute an arbitrary body's volume, i.e., in general a body does not even have to be a polyhedron. But for convenience, we will discuss convex polyhedron in most of the paper and in the end we will show some empirical results on some non-convex bodies' volume computation. Specially, we use the *halfspace representation* of a polyhedron. That is to say, a polyhedron is specified by $P = \{X|AX \leq B\}$ for some $m \times n$ matrix A and m-vector B (the pair (A, B) is called a *halfspace representation* of P).

As for two and three dimensional polyhedra, the volume of some basic shape (rectangle for instance) can be computed using some known mathematical formulas. But it is not straightforward to efficiently handle more general and complicated instances using exact analytic method. What is more, as the dimension n increases, the computational effort required arises drastically. Dyer and Frieze show that if P is a polyhedron then it is $\#P$-hard to evaluate the volume of P [4]. But later on, by introducing randomness into volume computation, Dyer, Frieze and Kannan gave a polynomial time randomized algorithm to estimate the volume to arbitrary accuracy in their pathbreaking paper [6]. That paper triggers the following works in this field, which reduce the complexity of the algorithm from n^{23} down to n^4 [10,9,1,5,11,8,12,13]. The method used is to reduce volume computation to sampling from a convex body, using the Multi-phase Monte Carlo method. It first constructs a sequence of incremental convex body $K_0 \subset K_1 \subset \cdots \subset K_m = V$ (the volume of K_0 is easy to compute). Then the volume of K_1 can be estimated by generating sufficiently many independent uniformly distributed random points in K_1 and counting how many of them fall in K_0, using an equation like equation (1). The other K_i's can be computed similarly. At last V (K_m) is obtained. When generating random points, the Markov Chain method is adopted.

However, in contrast with the brisk development on the complexity aspect of the randomized algorithms, little attention is paid to bring the theoretical results into practical use. As mentioned in [3], although these randomized algorithms are very interesting, there does not seem to be any implementation of them. That is why the authors of [3] ignored the randomized algorithm in their practical study on volume computation. In this paper, in contrast, we mainly focus on such practical randomized approximate algorithms. Our algorithm is also based on a Monte Carlo sampling method. But, compared with those Markov chain Monte-Carlo-based algorithms, our method is much simpler and much easier to implement. What is more, preliminary experimental results are very promising.

2 The Framework of the Sampling Algorithm

Assume that the volume of the convex polyhedron to be computed is V, our algorithm tries to estimate the value of V by the random sampling method.

We first generate N uniformly distributed random points in the convex polyhedron P, then at step i we build a probing sphere S with radius r_i in the

polyhedron[1]. After that we count the number of points that fall into S, denoted by N_p. Since the volume of the sphere (denoted by W) can be obtained immediately[2], we can easily obtain the volume of the polyhedron V_i from the following formula:

$$\frac{W}{V_i} = \frac{N_p}{N} \tag{1}$$

In particular, for the sake of higher accuracy, we carry out the probing procedure multiple times to obtain the average value (in the algorithm we use an adjustable parameter **Num_Of_Iteration** to denote this number).

Formally, the algorithm can be described as follows:

Algorithm 1. VOL(N)

```
1: generate N points randomly in P;
2: sum = 0;
3: for i = 1 to Num_Of_Iteration do
4:    build a probing sphere S in P with radius r_i;
5:    count the number of points in S;
6:    compute the volume V_i of P using formula (1);
7:    sum = sum + V_i;
8: end for
9: V = sum/Num_Of_Iteration;
10: return V;
```

3 Implementation

In general, the algorithm framework in section 2 is very simple and is easy to understand. But when putting it into practice, we must handle some difficult technical points and some of them need theoretical analysis.

3.1 Generating Random Points

As shown in Algorithm 1, first of all, we have to generate a lot of uniformly distributed points in the convex polyhedron. Generating points in a given (fat) convex body is easy. But it is hard to generate points that are distributed uniformly. Most previous work adopts a Markov Chain method to obtain theoretically uniformly distributed points. The idea is to divide the space into n-dimensional cubes and perform a random walk on all the cubes that lie within the given convex polyhedron. For each walk, one of the cubes that are orthogonally adjacent to the current cube in the convex body is randomly selected. Thus the random walk is ergodic and the stationary distribution is uniform on cubes in the

[1] Formally, an n-dimensional probing sphere S centering at $(o_1, o_2, \cdots, o_n)$ is defined as $S = \{(x_1, x_2, \cdots, x_n) | \sum_{j=1}^{n} (x_j - o_j)^2 \leq r_i^2\}$.

[2] We know that $W = \pi^{n/2} r_i^n / \Gamma(1 + n/2)$, where Γ denotes the gamma function [15].

convex polyhedron. However, this method is too complicated to use in practice. Instead, we use the pseudo-random number generator to generate many points in the polyhedron and assume them to be uniformly distributed. But there are still some uncertainties in our method. For instance, how many random points are needed. Apparently, for the sake of accuracy, the more the better. But on the other hand, more points mean more time and lower speed. So we must take both into consideration and find a proper compromise. In our preliminary implementation, the number of points is defined as an adjustable parameter so that it can be tuned according to different cases.

3.2 On Selecting the Center of the Sphere

Once the sampling problem has been solved, we begin to probe in the polyhedron with a probing sphere. But before probing, an implied prerequisite condition must be fulfilled. That is to say, the probing sphere must perfectly lie in the convex polyhedron. Otherwise, it is easy to see that the result obtained from equation (1) will not be accurate. But how to make sure that the whole probing sphere stays within the convex polyhedron? Strictly speaking, the distance from the center of the sphere to each of the facets of the convex polyhedron should be at least as large as the radius of the sphere. To achieve this goal, we define a new polyhedron contained in the original polyhedron named the *shrunk* polyhedron. The *shrunk* polyhedron has the same number of facets and vertices as the original polyhedron. In fact they should have the same shape except that it is a smaller version of the original polyhedron. Each facet of the *shrunk* convex polyhedron is parallel to its counterpart in the original convex polyhedron and the distance between them should be at least r_i. If we have such a *shrunk* polyhedron, then the problem can be solved easily by restricting the center of the sphere to lie within the *shrunk* polyhedron. If we use the *halfspace representation* of P, the *shrunk* polyhedron can be obtained easily and accurately by simply replacing each linear constraint $\sum_{k=1}^{n} a_{jk}x_k \leq b_j$ with $\sum_{k=1}^{n} a_{jk}x_k \leq b_j - r_i * \sqrt{\sum_{k=1}^{n} a_{jk}^2}$.

However, Algorithm 1 does not state that the body must be represented by linear constraints. In fact, the body can be presented to the algorithm using a very general mechanism called a *membership oracle*, which only tells whether a point is inside the body[3]. That is to say, the algorithm is also applicable to nonlinear constraints and other complicated constraints. But for these representations, it is hard for us to obtain the *shrunk* body accurately. So we have to use some approximate methods or heuristics. For example, we may adopt a random select-and-test heuristic. First, we randomly select a point in the original body as the center of the probing sphere. Then, given a radius r_i, we randomly choose some points on the surface of the probing sphere and test whether all of these points are also contained in the original body. If some point fails the test, we will try another point as the center of the probing sphere and perform this select-and-test procedure again. Formally, it can be formulated in Algorithm 2.

[3] Remember that in general a body may not be a polyhedron.

Algorithm 2. ForCenter(r_i)

```
1: FOUND=0;
2: for j = 1 to Num_Try_Center do
3:    Selecting a point in the body randomly as the center of the sphere;
4:    for k = 1 to Num_Try_Surface do
5:       Generating a point x on the surface of the sphere with radius r_i;
6:       if X is not in the body then
7:          break;
8:       end if
9:    end for
10:   if k > Num_Try_Surface then
11:      FOUND=1;
12:      break;
13:   end if
14: end for
15: return FOUND;
```

The select-and-test heuristic is easy to carry out but there are still some details that we need to clarify. For example, how many points on the surface of the sphere should be tested in the testing process. In general, the more the better. But again in practice more points mean more resources and more running time. The number of points should also vary with the dimension of the probing sphere. We again use an adjustable parameter **Num_Try_Surface** to represent the number in our experiments and it turns out that the heuristic works very well.

3.3 On Radius Selection

As described above, when building the probing sphere, we should determine the radius of the sphere beforehand. It is easy to understand that the radius should not be too small so that there is no point falling into the probing sphere at all. That is to say, there should be at least some point in the sphere. Otherwise, that probing sphere is useless. Based on the random sampling method, we have the following probabilistic analysis.

Theorem 1. *Given an n-dimensional polyhedron with volume V and the total number of sampling points N, if the radius r of the probing sphere satisfies $r = \Theta(\sqrt[n]{\frac{V*\ln N}{N}})$ then the probability that there is at least one point in the sphere converges to* 1 *as N grows to infinity.*

Proof. Let W be the volume of the probing sphere. Let C_1 denote $\pi^{n/2}/\Gamma(1 + n/2)$. Then $W = C_1 * r^n$. Assume that $r = C_2 * \sqrt[n]{\frac{V*\ln N}{N}}$. Let E represent the event that the sphere is empty, then $\bar{E}$ represents the event that there is at least one point in the sphere. Then we have:

$$\begin{aligned} P[\bar{E}] &= 1 - P[E] \\ &\geq 1 - (\frac{V-W}{V})^N \\ &= 1 - (1 - \frac{C_1 * r^n}{V})^N \\ &= 1 - (1 - \frac{C_1 * {C_2}^n * \ln N}{N})^N \\ &= 1 - e^{-C_1 * {C_2}^n * \ln N} \\ &= 1 - \frac{1}{N^{C_1 * {C_2}^n}} \end{aligned} \tag{2}$$

Although C_1 varies with dimension n [15], it is clear that given an n-dimensional instance, when N is big enough, $P[\bar{E}]$ will converge to 1. Thus we complete the proof. □

On the other hand, in theory, the bigger r is, the better the approximation is.

Claim. Convergence is better when r is bigger.

Proof. Let $p = W/V$ denote the probability that a random point from the polyhedron also falls into the probing sphere. Then the distribution of the random variable N_p will conform to a binomial distribution $\mathbf{B}(N, p)$. Thus we have

$$\mathbf{Var}(N_p) = N * p * (1-p) \qquad \mathbf{Mean}(N_p) = N * p$$

It follows that

$$\mathbf{Var}(N_p/N) = p * (1-p)/N \qquad \mathbf{Var}(N_p)/\mathbf{Mean}(N_p) = 1 - p$$

The formula on $\mathbf{Var}(N_p/N)$ reveals that the convergence is better when p is near one[4] than near 1/2. If $\mathbf{Var}/\mathbf{Mean}$ is used as a measure of convergence, we will also find that the smaller $1-p$ is, the better the convergence is. While small $1-p$ means big W, so it is easy to see that the claim holds. □

The above analyses suggest that we had better find a radius that is as big as possible. Theoretically it is indeed the case but in practice a probing sphere with the largest possible radius may have its weakness in the uniformity of probing. Take a triangle for example, the largest probing sphere inside it may be the inscribed circle of it. If we use the largest probing sphere, we will restrict our probing to the sampling points in the inscribed circle only. However, because all the sampling points are simulated by pseudo-random numbers, we cannot guarantee whether they are absolutely uniformly distributed in any part of the polyhedron. Therefore, we make sure that the probing sphere can visit as many parts of the sampling points as possible, so as to make the probing more general. For this reason, a moderately large but not an extremely large radius may be more suitable.

[4] The case of p near 0 is trivial, so we do not consider it.

Theorem 1 also reveals that r depends on V, which is exactly something we need to compute. In theory we can estimate an upper bound of V by building another convex polyhedron, which contains the original convex polyhedron and has a volume that is easier to compute. For example, the smallest axis-parallel bounding box may be enough for that purpose. This method heavily depends on the ratio between the volume of the polyhedron and the volume of the bounding box. But the ratio can be very small. What is more, for instances with nonlinear constraints, it is not always convenient to obtain the smallest bounding box.

To handle this problem, we adopt a self-adaptive heuristic in our implementation. First, we randomly choose two of the sampling points in the polyhedron and let r be the distance between them. With the current radius r, if we fail to find a proper center of the sphere after **Num_Try_Center** tries using the heuristic method given in the previous subsection, we assign $r \leftarrow r/2$. Once we find a proper sphere center, we stop so as to make r as big as possible. Experiments show that this self-adaptive method not only works on polyhedra with normal shapes, but is also competent for polyhedra of long skinny shapes.

4 Experiments and Analysis

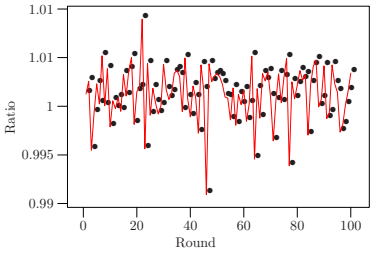
Based on the above observations, a prototype tool is implemented. We experiment on many simple instances to examine its performance. For the sake of comparison, we also test it on instances with known volume (named REAL volume) and examine the ratios of the results computed by our program to the REAL volume. Due to the space limit, we only introduce some simple ones.

4.1 Simple Examples

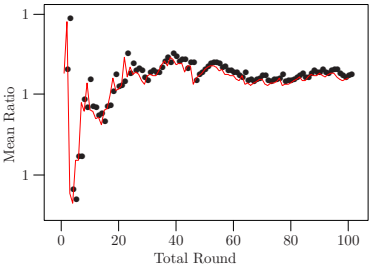
Example 1 $\begin{cases} -x + 2y \leq 200 \\ -y \leq 0 \\ x - y \leq 0 \\ -x - y \leq 50 \\ x + y \leq 200 \end{cases}$. It is in fact a pentagon with vertices $(0, 0)$, $(-50, 0)$, $(-100, 50)$, $(200/3, 400/3)$, and $(100, 100)$ and its REAL volume (the area of the pentagon) is $38750/3$.

Example 2 $\begin{cases} x + y + z \leq 255 \\ -x \leq 0 \\ -y \leq 0 \\ -z \leq 0 \end{cases}$. It is in fact a tetrahedron defined by the four vertices $(0, 0, 0)$, $(255, 0, 0)$, $(0, 255, 0)$ and $(0, 0, 255)$. So we can easily obtain the REAL volume $(255 * 255 * 255)/6$ by hand.

We test our tool on these instances and check the ratio of the program results to the REAL volumes. The experimental results are given in Fig. 1 and 2 respectively in detail.

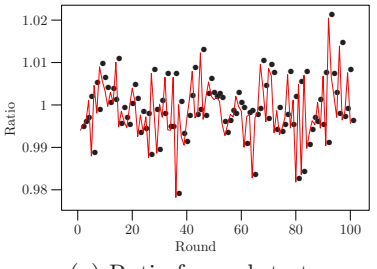


(a) Ratio for each test

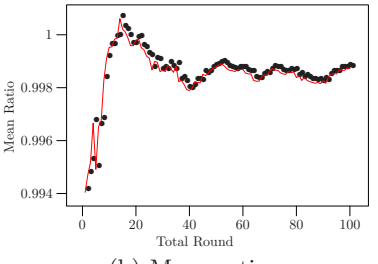


(b) Mean ratio

Fig. 1. Experimental results of *Example 1*



(a) Ratio for each test



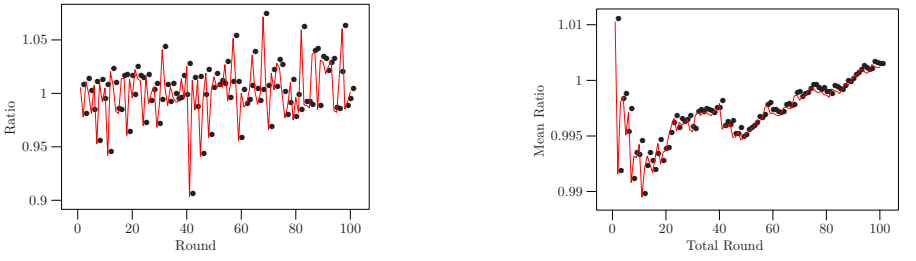
(b) Mean ratio

Fig. 2. Experimental results of *Example 2*

4.2 Variance Analysis

Experimental results from both Fig. 1 and Fig. 2 show that our method indeed has a good approximation. The mean values are very close to 1 and they converge well. However, there are still some small differences between Fig. 1(a) and Fig. 2(a). For example, all the ratio values in Fig. 1(a) fall within a small interval while those in Fig. 2(a) fall within a relatively large interval. As far as the variance is concerned, why does the data in Fig. 2(a) have a larger variance compared with those in Fig. 1(a)? To find out the possible reason, we experiment on *Example 2* again with fewer sampling points. Results are presented in Fig. 3. Comparing Fig. 3(a) with Fig. 2(a), we find that given a fixed volume, fewer sampling points result in relatively larger variance.

Theoretically speaking, our sampling algorithm needs sufficiently many independent uniformly distributed points, but in practice we can only generate a finite number of points. How large should this number be? It is a problem to be settled. If we use a fixed number for each dimension, then instances with smaller volume will have larger density compared with those with bigger volume. Given two polyhedra in the same dimension, the volume ratio, however, can be arbitrary large, which may result in sharp difference on density. On the other hand,



(a) Ratio for each test (b) Mean ratio

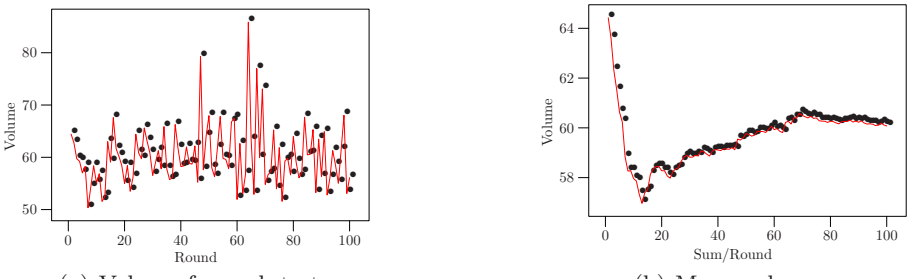
Fig. 3. Experimental results of *Example 2* (using fewer sampling points)

it is clear that this number should vary with the dimension. Maybe this number should be an exponential function on the dimension but we are not able to find a precise definition yet. In our current implementation, we use an adjustable parameter to denote the number of points before running the program. It is certainly better to find a self-adaptive heuristic, which can dynamically adjust the number during the running of the program. We leave this for our future work.

4.3 On the Mean Ratio

Although the variance varies with different sampling densities, all the mean ratios in Fig. 1(b), Fig. 2(b) and Fig. 3(b) show a very good approximation to 1. The curves reveal a nice trend of convergence to 1 as the total Round number grows from 1 to 100. This is easy to understand. Even if the ratio deviation of one particular case is 100%, it will only contribute 1% up to the total deviation of the mean value because it is divided equally among all the 100 sampling Rounds. So in general it makes sense to run more Rounds.

At last, we want to emphasize that the sampling algorithm is a general method. Although there are some negative results of it when the dimension



(a) Volume for each test (b) Mean volume

Fig. 4. Experimental results

n grows to infinity [14], our experiments show that the method is still feasible for real-life instances in high dimension. However, as for high dimensional and general non-convex instances, although the method can compute/approximate volume, we cannot always carry out comparisons as we do above, because we have no other means to obtain the REAL volume for general instances. Despite that, for some two-dimensional non-convex cases (whose REAL volume can be computed analytically) our method can obtain a mean ratio which always converges to one. For more complex non-convex cases, we believe that our algorithm also has good approximations in practice. For instance, given an instance $\begin{cases} y \geq x^2 - 2 \\ y \leq \frac{1}{2}x^2 \\ -10 \leq z \leq 5 \\ xyz \leq 1 \end{cases}$, we do not know the REAL volume, but our tool can tell us that it is about 60, as depicted in Fig. 4.[5]

5 Concluding Remarks

Volume computation is a very hard but widely studied problem in mathematics and computer science. Generally speaking, there are two different methods on this problem: the exact method and the randomized approximation method. The exact method itself can be classified into two classes: triangulation methods and signed decomposition methods. Benno Büeler, Andreas Enge and Komei Fukuda present some practical study on these methods [3]. But these methods are only applicable to convex bodies with linear constraints. The randomized approximation method is a more general one. It can cope with almost any constraints presented to it. Although the randomized approximation method is relatively new, a lot of progress has been made since its birth. Our work is also based on the randomized method.

However, most of these efforts on randomized algorithms are on the complexity aspect and little practical studies are given. In this paper some implementation issues of volume computation are studied. Some techniques in randomized volume computation algorithms are quantitatively evaluated. For example, our algorithm is also based on Monte Carlo sampling, but we do not use the Multiphase method as described above. Instead, we use only one phase Monte Carlo method but we run the probing process many times to obtain a more accurate average result. On generating uniformly distributed points, we do not use the Markov Chain method although it does well in simulating the uniformly distribution in theory. Instead, we generate random points directly within the polyhedron and view them as uniformly distributed ones in our algorithm. Techniques and problems on efficient and effective implementation to achieve good performance are also discussed. Preliminary empirical results show that the tool developed by utilizing these results works very well. Of course, there are still some unsolved problems related to some of the manually adjustable parameters. We leave them for future research.

[5] See Appendix A for more examples.

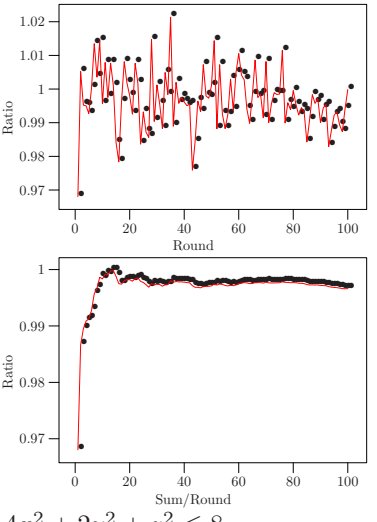
References

1. David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In: Proc. 23rd annual ACM symp. on Theory of Computing (STOC), pp. 156–163 (1991)
2. Bollobás, B.: Volume estimates and rapid mixing. Flavors of geometry. Math. Sci. Res. Inst. Publ. 31, 151–182 (1997)
3. Büeler, B., Enge, A., Fukuda, K.: Exact volume computation for polytopes: a practical study. Polytopes–combinatorics and computation (1998)
4. Dyer, M., Frieze, A.: On the complexity of computing the volume of a polyhedron. SIAM J. Comput. 17(5), 967–974 (1988)
5. Martin Dyer and Alan Frieze. Computing the volume of convex bodies: A case where randomness provably helps. In: Proc. 44th Symp. in Applied Mathematics (PSAM) (1991)
6. Dyer, M., Frieze, A., Kannan, R.: A random polynomial-time algorithm for approximating the volume of convex bodies. J. ACM 38(1), 1–17 (1991)
7. Gritzmann, P., Klee, V.: On the complexity of some basic problems in computational convexity: II. volume and mixed volumes. Polytopes: abstract, convex and computational (Scarborough, ON, 1993), NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., pp. 373–466 (1994)
8. Kannan, R., Lovász, L., Simonovits, M.: Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. Random Struct. Algorithms 11(1), 1–50 (1997)
9. ó Lovász, L.: How to compute the volume? Jber. d. Dt. Math.-Verein, Jubiläumstagung, B. G. Teubner, Stuttgart, pp. 138–151 (1990)
10. Lovász, L., Simonovits, M.: The mixing rate of markov chains, an isoperimetric inequality, and computing the volume. In: Proc. 31th IEEE Annual Symp. on Found. of Comp. Sci (FOCS), pp. 482–491 (1990)
11. Lovász, L., Simonovits, M.: Random walks in a convex body and an improved volume algorithm. Random Struct. Algorithms 4(4), 359–412 (1993)
12. Lovász, L., Vempala, S.: Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. In: ó Lovász, L. (ed.) Proc. 44th IEEE Annual Symp. on Found. of Comp. Sci (FOCS), pp. 650–659 (2003)
13. Rademacher, L., Vempala, S.: Dispersion of mass and the complexity of randomized geometric algorithms. In: Proc. 47th IEEE Annual Symp. on Found. of Comp. Sci (FOCS), pp. 729–738 (2006)
14. Simonovits, M.: How to compute the volume in high dimension? Mathematical Programming 97, 337–374 (2003)
15. Weisstein, E.: Ball. From MathWorld – A Wolfram Web Resource (2003), available at `http://mathworld.wolfram.com/Ball.html`

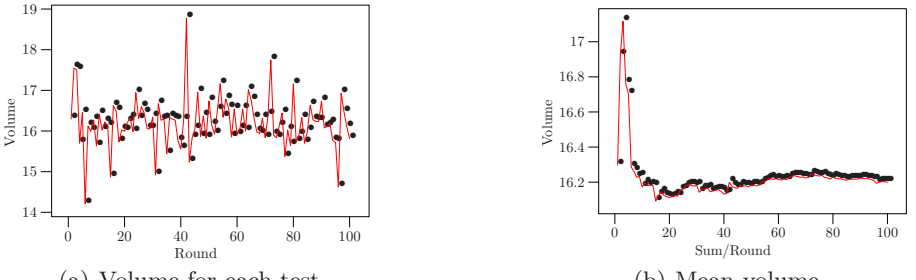
A More Examples

Example (a) Given a 4-dimensional polyhedron below, its REAL volume is about 2/3. Our tool can also give the results depicted on the right of the below instance.

$$\begin{cases} -x+y+z-u \le 1 \\ -x+y+z+u \le 1 \\ -x+y-z-u \le 1 \\ -x+y-z+u \le 1 \\ -x-y-z-u \le 1 \\ -x-y-z+u \le 1 \\ -x-y+z+u \le 1 \\ -x-y+z-u \le 1 \\ x-y-z+u \le 1 \\ x-y-z-u \le 1 \\ x-y+z+u \le 1 \\ x-y+z-u \le 1 \\ x+y-z+u \le 1 \\ x+y-z-u \le 1 \\ x+y+z+u \le 1 \\ x+y+z-u \le 1 \end{cases}$$



Example (b) Given a 3-dimensional body $\begin{cases} 4x^2+2y^2+z^2 \le 8 \\ x^2+4y^2+2z^2 \le 8 \\ 2x^2+y^2+4z^2 \le 8 \\ xy \le 1 \end{cases}$, it is in fact the intersection of three ellipsoids and another instance denoted by $\{xy \le 1\}$. We do not know its REAL volume, but our tool can tell us the approximate volume (about 16.2) as depicted below.



(a) Volume for each test (b) Mean volume

Fig. 5. Experimental results