# Review of Volume Computation

Rares Cristian

8 March 2019

# Contents

# 1 Introduction

Computing the volume of a convex body $K \subset \mathbb{R}^n$ has been a long-studied question, important from both a theoretical and practical perspective. We present a survey of current techniques to solve this problem, as well as connect it to the study of pseudorandom generators.

## 1.1 Volume

We define the volume of $K$ as the Lebesgue measure over $K$. This is a standard way of assigning measure to subsets of $\mathbb{R}^n$ and follows the standard definitions of area and volume in 2 and 3 dimensions that we are used to. We denote the volume of $K$ as $\text{vol}(K)$.

## 1.2 Model of Computation

A key aspect of the problem lies in the representation of the set. Perhaps in the most general sense, a description of the body is not directly given, but rather access is provided by an oracle. A *membership oracle* may be given an input $x \in \mathbb{R}^n$ and returns whether or not $x$ is contained in $K$. Alternatively, a *separation* oracle determines if $x$ lies in $K$, and if not provides a hyperplane separating $x$ and $K$. A well-guaranteed oracle additionally provides an initial point, $x_0$ guaranteed to be in $K$, as well as bounds on the size of $K$: a guarantee that a ball of radius $r$ is contained completely within $K$, and a guarantee that a ball of radius $R$ fully contains $K$. That is, $x_0 + rB^n \subset K \subset x_0 + RB^n$.

On the other hand, we may be given an explicit description of the set. The most common example is that of a polytope. Even here, there are various ways to represent it, for example, as the convex hull of a set of points, or as the intersection of halfspaces. The latter is often more expressive, since with only a polynomial number of halfspaces, we potentially require an exponential number of vertices to describe the same polytope. Take the simple example of a cube in $n$ dimensions. It is the intersection $2n$ halfspaces, but has $2^n$ corners.

# 2 Algorithms

Given only a well-guaranteed separation oracle, no deterministic polynomial-time algorithm can approximate the volume to within a factor exponential in the dimension [1]. For any sequence of $n^a$ points there exist two different convex sets with volume ratio

$$\left(\frac{cn}{a \log n}\right)^n$$

whose oracles produce the same result to each query (for some universal constant $c$). Therefore, given no further information, there is no way to differentiate between the two bodies. Surprisingly, there do exist efficient randomized algorithms to compute volume given only an oracle. Volume computation is one of a few problems where randomization provably helps.

## 2.1 Overview

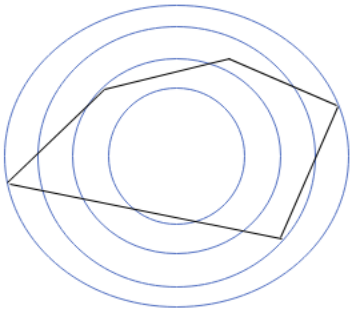
The main technique, which we will mainly follow, is to construct a series of bodies $K = K_0 \supset K_1 \supset \cdots \supset K_m$ in such a way that the volume of $K_m$ can be easily calculated, and the ratio of volumes of $K_i$ to $K_{i+1}$ can be well approximated. This last step may be done by sampling uniformly at random from the interior of $K_i$ and taking the ratio of points lying in $K_{i+1}$. The current state-of-the-art methods perform this sampling via random walks inside the body. The volume can be directly found as

$$\text{vol}(K) = \text{vol}(K_m) \prod_{i=0}^{m-1} \frac{\text{vol}(K_i)}{\text{vol}(K_{i+1})}$$

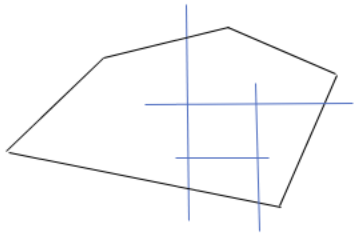
At this point, one wishes for two things: to reduce the number of bodies $K_i$ needed, and to reduce the number of samples required at each step. Note that the errors in approximating volume ratios are multiplicative; having fewer intermediary bodies allows us to have larger errors in approximating volume ratios, which in turn allows us to sample fewer points. At the same time, we want the ratios to be fairly large, so $K_{i+1}$ needs to have a constant fraction of the volume of $K_i$. This is so we only require a polynomial number of samples to accurately approximate the ratio. Note that this restriction places a lower bound on the number of intermediary bodies needed.

## 2.2 Constructing $K_i$

We present two ways of constructing the sequence $\{K\}_i$ of bodies. First, consider creating a sequence of balls whose volume roughly doubles at each iteration. Let $r_i$ be the radius of the $i^{th}$ ball, and $r_0 = r$. Let $r_i = r_{i-1}(1 + \frac{1}{n})B^n$. The volume of the $i^{th}$ ball is $(1 + \frac{1}{n})^n$ times greater than the previous ball, or roughly $e$ times larger. Define $K_i = K \cap (x_0 + r_i B^n)$.



The second method utilizes the center of gravity of $K$. A sequence of bodies $K_i$ may be found as follows. Let $z_i$ be the center of mass of $K_i$. Consider the hyperplane through $z_i$ perpendicular to $e_{i \bmod n}$, one of the standard axes (note we are simply choosing these in successive order, round-robin). This hyperplane divides the polytope into two halves, and we let $K_{i-1}$ be the half which contains $z_0$, the center of mass of $K$. Since we choose these hyperplanes perpendicular to the axes, this process will eventually terminate in some $K_m$ being a box, whose volume is simply the product of its side lengths. It remains to bound $m$.



We first note that this process is affine-invariant. Consider applying this algorithm on a body $K$ and applying it on $AK$, where $A$ is an affine transformation in $\mathbb{R}^n$. In $K$, this will produce a sequence $c_1, \ldots, c_m$ of centroids, and in $AK$ a sequence $b_1, \ldots, b_m$ of centroids. However, $Ac_i = b_i$. Thus, there is a bijection between the stesps taken in the $K$ and $AK$. Most importantly, termination in one would imply termination in the other and so the number of iterations required for $K$ and $AK$ are the same. Of particular interest is the affine transformation $A$ which brings $K$ into isotropic position.

A set $S$ is in isotropic position if for a random point $x \in S$, both $\mathbb{E}[x] = 0$ and $\mathbb{E}[x^T x] = I$. For a body $K$ in isotropic position,

$$\sqrt{\frac{n+2}{n}} B^n \subseteq K \subseteq \sqrt{n(n+2)} B^n$$

Essentially, a ball of unit radius is contained in $K$, and a ball of radius $O(n)$ contains $K$. We now focus on bounding $m$ for a body in isotropic position. As mentioned, this will directly imply the general case.

Choosing the centroid at each step brought us a few advantages. Any halfspace $H$ which contains the center of gravity of $K$, also contains $1/e$ of the volume of $K$ [**?**]. Therefore,

$$\frac{\text{vol}(K_{i+1})}{\text{vol}(K_i)} \leq 1 - \frac{1}{e}$$

We require to know the polytope contains a cube of side length $r$, and a cube of side length $R$ completely contains the polytope. For an isotropic body, a cube of width $O(n)$ clearly will contain it, and a cube of width $1/\sqrt{n}$ will be contained in the unit ball, and thus in $K$.

We may terminate this sequence of bodies once $K_m$ has a volume at most that of the small cube of side length $r$. We are guaranteed that this $K_m$ will be fully contained in the body, since we assured that at each step, $K_i$ roughly contains the cube of width $r$. Thus, we require on the order of $\log n^n / (1/\sqrt{n})^n = O(n \log n)$ intermediary bodies to reach a cuboid.

However, computing the centroid exactly is #P-hard [2]. On the other hand, the average of $O(\log^2 m)$ sample points provides a good approximation of the centroid [3]. Current algorithms find these via Markov Chain Monte Carlo methods.

# 3 Sampling

A geometric random walk is a sequence of points $x_0, x_1, \ldots$ in $\mathbb{R}^n$ for which $x_{i+1}$ is chosen from a neighborhood of $x_i$ according to some distribution depending only on $x_i$. In particular, we aim for random walks within $K$ whose stationary distribution is the uniform distribution over $K$. Additionally, we wish to bound the rate of convergence. Here, we have a brief discussion of three walks, as well as isoperimetry, a key tool used in their analysis. For a complete survey on geometric random walks, we defer to [4].

## 3.1 Examples

- Grid Walk

  The first major breakthrough in a polynomial-time algorithm for approximating volume came from Dyer and Frieze [5] in which they first introduced the ball walk. In it, we pick a grid point $y$ uniformly at random from the neighbors of the current point $x$. If $y \in K$, go to $y$, otherwise, stay at $x$. Since, many improvements have been found, as well as other walks that have faster mixing times.

- Ball Walk

  Here, we pick a uniform random point $y$ from the ball of radius $\delta$ centered at the current point $x$. If $y \in K$, we go to $y$. Otherwise, we stay at $x$. Some care needs to be taken here. If $x$ is near a corner, it possible that an exponentially small portion of the ball centered at $x$ will lie within the ball. This would imply that to take a single step, we may require an exponential number of steps. One possible rectification is to begin from a *warm start*. That is, we have an initial point drawn from a distribution already close to the uniform distribution (closeness here is measured by total variation distance). In this setting, these pathological initial points mentioned would essentially never be chosen to start with, and the chances of reaching them during the walk would be equally equally small.

The main result needed to be proven is to bound the conductance. This is generally done in two steps:

1. Show that, given two points that are close, then their 1-step distributions (that is, the set of points they could be in after a single step) will have significant overlap.

2. Show that large subsets have large boundaries. This essentially amounts to bounding the cheeger constant of the stationary distribution, which we discuss in the following section.

Clearly, if we only have access to a convex body through an oracle, there is no way to escape randomness. However, an outstanding problem is in finding a deterministic algorithm specifically for polytopes. The only known hardness results are for exact computation of the centroid, and so it is possible that there exists a deterministic polynomial-time algorithm for finding a constant approximation of the volume.

### 3.2 Isoperimetry

The classical isoperimtric problem is to find a set with minimal surface are for a given volume. The solution to this problem, the sphere, has been known since the ancient Greeks. However, here we consider a different version of this problem: We want to find a surface which divides a convex body in two with smallest surface area relative to the resulting volume of the two halves. That is, we want to find the largest $\psi$ (the cheeger constant) such that

$$\mathrm{vol}_{n-1}(\partial S) \geq \psi \min\{\mathrm{vol}(S), \mathrm{vol}(K \setminus S)\}$$

for any $S \subset K$. A lower bound on the conductance is

$$\psi \geq \frac{\ln 2}{M_1(K)}$$

where $M_1(K)$ is the average distance of a point to the center of gravity of the body $K$. If $K$ is isotropic, it follows that $\psi \leq \frac{\ln 2}{\sqrt{n}}$

But what is the motivation for studying this? For example, the mixing time for the ball walk is $O^*(\frac{n^2}{\psi^2})$. For isotropic bodies, we have the upper bound $\psi \geq \frac{c}{\sqrt{n}}$. Thus, the algorithm itself is currently known to require $O(n^3)$ steps to converge. However, it is conjectured that $\psi$ is lower bounded by a constant independent of the dimension. If this were true, it would imply that the algorithm actually requires $O(n^2)$ steps.

## 4 Pseudorandomness

It is explicitly required that we have access to a random source of bits, since as mentioned earlier, no deterministic algorithm can be efficient. This poses a problem as any implementation is inherently deterministic. Even using pseudorandom numbers cannot replace a source of truly random bits. We will now focus our attention specifically on polyoptes expressed as the intersection of halfpsaces rather than general sets described by an oracle.

### 4.1 Generators

A pseudorandom number generator is a function or algorithm which receives a short sequence of random bits, the seed, and produces a much longer sequence which appears to be random. Clearly, this output cannot be truly random, since the number of possible outputs far outnumbers the size of the seed. Nevertheless, we will use the notion of *computational indistinguishability* to describe randomness. That is, we say that a variable looks random if there is no algorithm which can efficiently distinguish it from a truly random variable. Formally,

> Random variables $X, Y$ taking values from $\{0,1\}^n$ are said to be $(t, \epsilon)$ indistinguishable if for any nonuniform algorithm $T$ running in at most $t$ steps,
>
> $$|Pr[T(X) = 1] - Pr[T(Y) = 1]| \leq \epsilon$$

We say a distribution is pseudorandom if it indistinguish able from the unifrom distribution.

### 4.2 Fooling Halfspaces and Polytopes

Here, we define a halfspace $h : \{0,1\}^n \to \{0,1\}$ to be a function of the form $h(x) = \text{sign}(w_1x_1 + \cdots + w_nx_n - \theta)$ where $w_1, \ldots, w_n, \theta \in R$. A generator $G$ with random seed length $s$ is said to $\epsilon$-fool any halfspace $h$ if

$$|\mathbb{E}_{x\in\{-1,1\}^s}[h(G(x))] - \mathbb{E}_{x\in\{-1,1\}^n}[h(x)]| \leq \epsilon$$

Polytopes are intersections of halfspaces, so we can define

$$F(x) = h_1(x) \wedge \cdots \wedge h_m(x)$$

In this polytope setting, we wish to find a small set of points $S$ in $\{0,1\}^n$ such that for all polytopes $F$, if $F$ accepts a $p$-fraction of the points in $\{0,1\}^n$, then $F$ accepts a $(p \pm \epsilon)$-fraction of points in $S$.

In particular, if our seed length $s = \log n$, we may enumerate over all possible bit strings of length $\log n$, of which there are $n$, and thus have a fully deterministic algorithm to fool halfspaces. Here are some of the current state of the art results:

| Class of Functions | Seed Length |
|---|---|
| Any function of $m$ general halfspaces [6] | $O(m \log 1/\epsilon)) \cdot \log n$ |
| Intersections of $m$ regular halfspaces [7] | $\text{poly}(\log m, 1/\epsilon) \cdot \log n$ |
| Intersections of $m$ low-weight halfspaces [8] | $\text{poly}(\log m, 1/\epsilon) \cdot \text{polylog } n$ |
| Interscetions of $m$ general halfspaces [9] | $\text{poly}(\log m, 1/\epsilon) \cdot \log n$ |

Note that these results are not sufficient for our purposes, since the seed length is still exponential in the error.

## 5 Final Remarks

Little is known about the necessity of randomness for solving various problems. The class of problems **P** is the only for which we have determistic polynomial-time algorithms. **BPP**, on the other hand, is the class of bounded-error probabilistic polynomial time problems. A problem is in **BPP** if there is a polynomial-time algorithm which is allowed to make coin flips to make its decisions that solves the problem. It is widely believed that **BPP** = **P**. If this were true, it would imply that there is indeed a polynomial-time approximation scheme to compute the volume of polytopes.

The converse, however, is not necessarily true. Nonetheless, there is currently a fundamental gap in our understanding of these two classes of problems. Resolving the polytope volume-approximation problem would bring new insights into the long-standing **BPP** = **P** problem.

# References


[1] Imre Bárány and Zoltán Füredi. Computing the volume is difficult. *Discrete & Computational Geometry*, 2(4):319–326, 1987.

[2] Luis A. Rademacher. Approximating the centroid is hard. In *Proceedings of the Twenty-third Annual Symposium on Computational Geometry*, SCG '07, pages 302–305, New York, NY, USA, 2007. ACM.

[3] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[4] Santosh Vempala. Geometric random walks: a survey. *Combinatorial and computational geometry*, 52(573-612):2, 2005.

[5] Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, January 1991.

[6] P. Gopalan, R. O' Donnell, Y. Wu, and D. Zuckerman. Fooling functions of halfspaces under product distributions. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 223–234.

[7] Prahladh Harsha, Adam Klivans, and Raghu Meka. An invariance principle for polytopes. *J. ACM*, 59(6):29:1–29:25, January 2013.

[8] R. A. Servedio and L. Tan. Fooling intersections of low-weight halfspaces. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 824–835.

[9] R. O'Donnell, R. A. Servedio, and L.-Y. Tan. Fooling Polytopes. *arXiv e-prints*, August 2018.