Computational Geometry

Volume computation and random sampling

Vissarion Fisikopoulos

Department of Informatics & Telecommunications
National & Kapodistrian University of Athens

Spring 2025

Outline

Volume computation problem

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

Volume computation problem

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

1. What is convex?

Volume computation problem

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

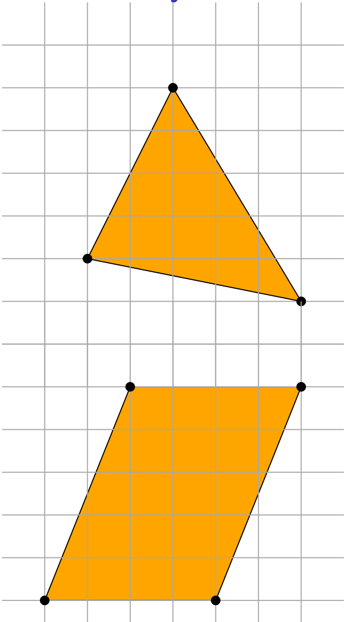
1. What is convex?
2. What is a polytope? How can we represent it?

Volume computation problem

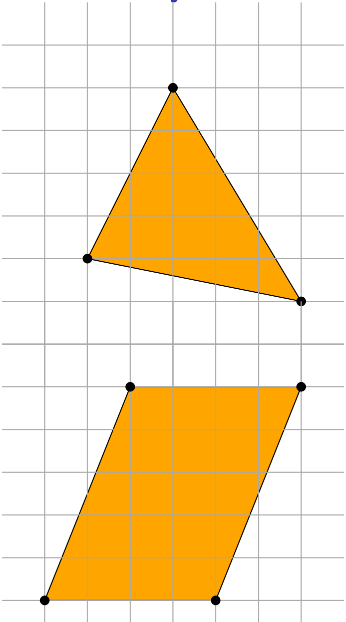
Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

1. What is convex?
2. What is a polytope? How can we represent it?
3. How large is d? e.g. $d = 2,\ 3,\ 50$

Easy cases: volume of elementary shapes



Easy cases: volume of elementary shapes



$$\begin{vmatrix} 1 & 2 & 1 \\ 3 & 6 & 1 \\ 6 & 1 & 1 \end{vmatrix} / 2! = 11$$

$$\begin{vmatrix} 2 & 5 \\ 4 & 0 \end{vmatrix} = 20$$

Volume of elementary shapes and a conjecture

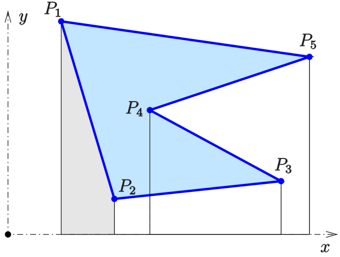
- What is the volume of the d-dimensional simplex, cube, crosspolytope?
- Mahler volume $= vol(P)vol(P^*)$ (P^* is the polar dual)
- Mahler conjecture: the minimum possible Mahler volume is attained by a hypercube

Easy cases: planar polygons

A planar simple polygon with a positively oriented (counter clock wise) sequence of points $P_1, \ldots, P_n$, $P_i = (x_i, y_i), i = 1, \ldots, n$.

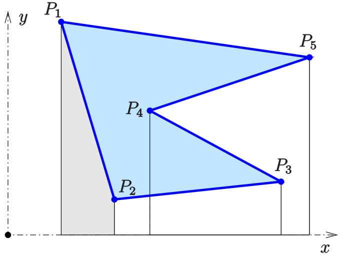
Easy cases: planar polygons

A planar simple polygon with a positively oriented (counter clock wise) sequence of points $P_1, \ldots, P_n$, $P_i = (x_i, y_i), i = 1, \ldots, n$.



Easy cases: planar polygons

A planar simple polygon with a positively oriented (counter clock wise) sequence of points $P_1, \ldots, P_n$, $P_i = (x_i, y_i), i = 1, \ldots, n$.



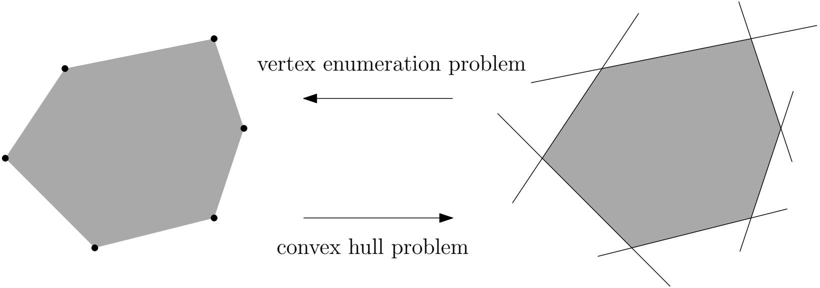
$$A = \frac{1}{2} \sum_{i=1}^{n} (y_i + y_{i+1})(x_i - x_{i+1})$$

$$= \frac{1}{2} \Big((y_1 + y_2)(x_1 - x_2) + \cdots + (y_n + y_1)(x_n - x_1) \Big)$$

Polytope Representations

A convex polytope $P \subseteq \mathbb{R}^d$ can be represented as the

1. convex hull of a pointset $\{p_1, \ldots, p_n\}$ (V-representation)
2. intersection of halfspaces $\{x \in \mathbb{R}^d \ : \ Ax \leq b\}$ (H-representation)



Faces of polytopes: vertices, edges, ..., facets

Volume computation problem (revisited)

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

Volume computation problem (revisited)

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

ChatGPT

Computing the volume of a convex polytope in $\mathbb{R}^d$ can be done using various methods depending on the representation of the polytope. Here's a general approach:

1. **Vertex Representation:** If the polytope is given by its vertices, you can use the formula for the volume of a convex polytope known as the "mixed volume." This formula generalizes the concept of volume to higher dimensions and is defined in terms of determinants.
2. **Half-space Representation:** If the polytope is given by the intersection of half-spaces (i.e., in the form $\{x \in \mathbb{R}^d : Ax \leq b\}$), then you can compute the volume using techniques like the Monte Carlo method or the Monte Carlo Markov Chain (MCMC) method. These methods involve sampling points within the polytope and estimating the volume based on the density of points.
3. **Convex Hull:** If you have a set of points representing the polytope, you can compute the convex hull of those points and then calculate the volume of the resulting convex hull.

Volume via triangulation

Algorithm: compute a triangulation of the input polytope, then sum up the volumes of simplices.

Volume via triangulation

Algorithm: compute a triangulation of the input polytope, then sum up the volumes of simplices.

Question: Which triangulation?

Volume via triangulation

Algorithm: compute a triangulation of the input polytope, then sum up the volumes of simplices.

Question: Which triangulation?

Problem: The size of the triangulation of n points could be exponential in the dimension d i.e. $O(n^{\lceil d/2 \rceil})$

Volume via triangulation

Algorithm: compute a triangulation of the input polytope, then sum up the volumes of simplices.

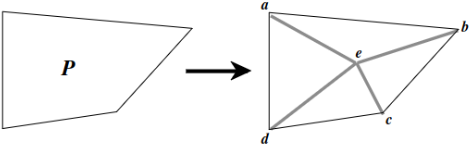
Question: Which triangulation?

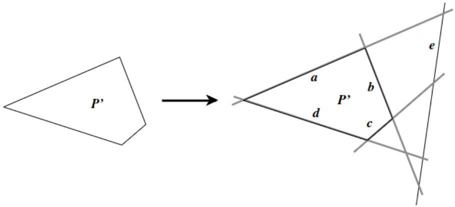
Problem: The size of the triangulation of n points could be exponential in the dimension d i.e. $O(n^{\lceil d/2 \rceil})$

Reference: Büeler, Enge, Fukuda - Exact Volume Computation for Polytopes: A Practical Study

Triangulation & sign decomposition methods

- Triangulation $T(P)$: $vol(P) = \sum_{s \in T(P)} vol(s)$
- Sign decomposition:
  $vol(P) = \sum_{v \in P} vol(cone(v) \cap e)\ sign(v)$
  $sign(v) = (-1)^{\#H}$





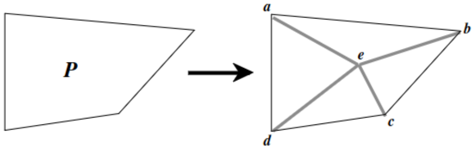
$$vol(p) = vol(ade) + vol(cbe) - vol(abe) - vol(cde)$$

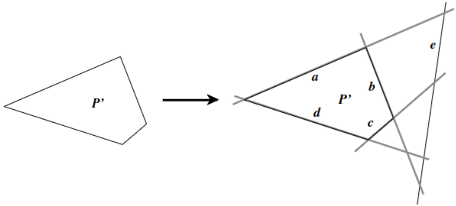
Implementations

- VINCI [Bueler et al'00], Latte [deLoera et al], Qhull [Barber et al], LRS [Avis], Normaliz [Bruns et al]

Implementations

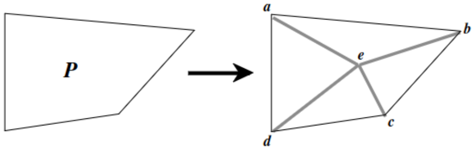
- VINCI [Bueler et al'00], Latte [deLoera et al], Qhull [Barber et al], LRS [Avis], Normaliz [Bruns et al]
- triangulation, sign decomposition methods

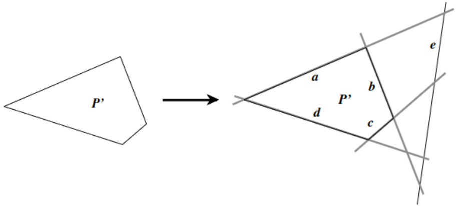




Implementations

- VINCI [Bueler et al'00], Latte [deLoera et al], Qhull [Barber et al], LRS [Avis], Normaliz [Bruns et al]
- triangulation, sign decomposition methods

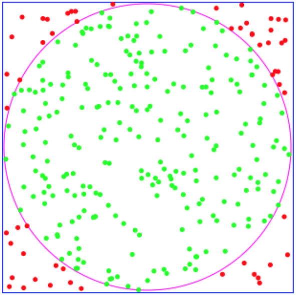




- cannot compute in high dimensions (e.g. > 15) in general

Volume via (naive) Monte Carlo

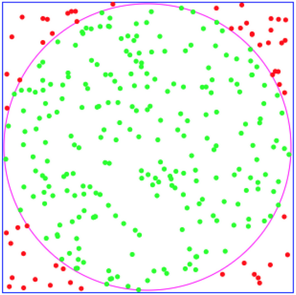
Rejections techniques (sample from bounding box)



Question: how to sample points from a cube?

Volume via (naive) Monte Carlo

Rejections techniques (sample from bounding box)



Question: how to sample points from a cube?

volume(unit cube) $= 1$
volume(unit ball) $\sim (c/d)^{d/2}$ –*drops exponentially with* d

Outline

Volume computation

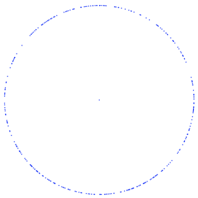
Random Sampling from Polytopes

Volume revisited (randomized)

Polytopes and Applications

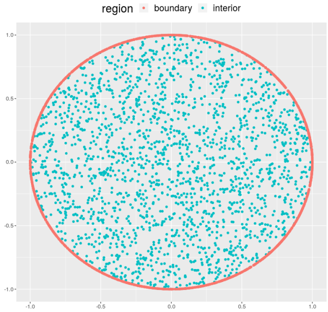
Uniform sampling from the simple shapes: hypersphere

- To sample uniformly from the **boundary** of a hypersphere of radius r:
  1. Sample d numbers $g_1, \dots, g_d$ from $\mathcal{N}(0,1)$.
  2. The point $v = r(g_1, \dots, g_d)/\sqrt{\sum g_i^2}$ is uniformly distributed on the surface of the d-dim hypersphere, of radius r and center the origin.
- To pick a random direction through point $p \in \mathbb{R}^d$, we sample from the surface of a hypersphere centered at p.



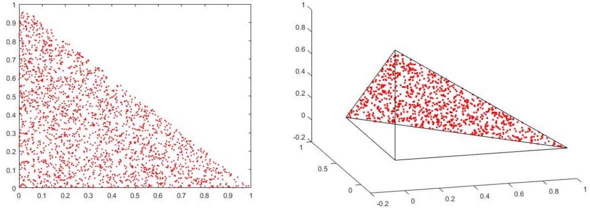
Uniform sampling from the simple shapes: hypersphere

- To sample uniformly from the interior of a hypersphere with radius r:
  1. Sample a point $v \sim \mathcal{U}(\partial B_d)$ and $u \sim \mathcal{U}(0,1)$.
  2. The point $p = ru^{1/d}v$ is uniformly distributed in the interior of the d-dim hypersphere, of radius r and center the origin.



Uniform Sampling from the simplex

1. [Smith, Tromble: 2004]:
   - Generate distinct: $0 = x_0 < x_1 < \cdots < x_{d+1} = M \in \mathbb{N}^*$. Return y: $y_i = \dfrac{x_i - x_{i-1}}{M}$, $i = 1, \ldots, d+1$. M: largest integer.
   - To guarantee distinct choice we use a variation of Bloom filter (check membership in a set).
   - Sampling one point takes $O(d \log d)$.
2. [Rubinstein, Melamed: 1998]:
   - Generate independent unit-exponential random variables, $X_1, \cdots, X_{d+1}$. Return $Y \in \mathbb{R}^{d+1}$: $Y_i = X_i / \sum_{i=1}^{d+1} X_i$.
   - Sampling one point takes $O(d)$.



General Polytopes: Geometric Random Walks

- A Geometric Random Walk starts at some interior point and at each step moves to a "neighboring" point, chosen according to some distribution depending only on the current point.

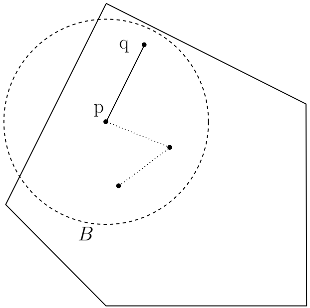


Figure: Steps of a ball walk.

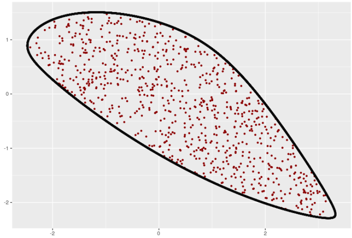


Figure: Uniform target distribution

Useful questions and terminology

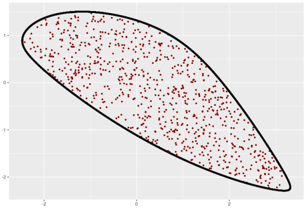
- Does the random walk converges asymptotically to the target distribution? (Correctness)
- How fast does it converge?
  (Equivalently) How many steps do we have to perform until we get a point that is ϵ-close to a point draw from the target distribution? (mixing time)
- Does the initial point of the walk affects the efficiency? (warm start)
- What is the cost per step of the random walk?
- Do we assume anything about input polytope P? (isotropic position, well rounded)

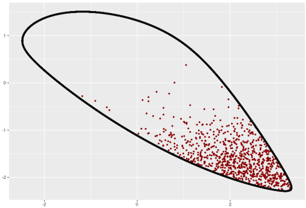
Target probability distributions

Definition

Let $\pi(\boldsymbol{x}) \propto e^{-f(\boldsymbol{x})}$, where $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function.
$\pi(\boldsymbol{x})$ is called *log-concave (LC) probability density*.

- Let $\pi(\boldsymbol{x})$ be restricted to **convex body** $K \subset \mathbb{R}^d$.
- Special cases: Uniform, Gaussian, Exponential/Boltzmann.

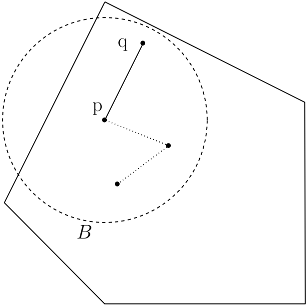




Ball walk

Ball Walk(K, p, δ, f): convex $K \subset \mathbb{R}^d$, $p \in P$, radius δ, $f : \mathbb{R}^d \to \mathbb{R}_+$

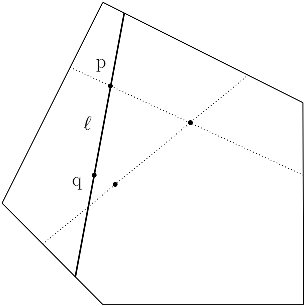
1. Pick a uniform random point x in $B(p, \delta)$.
2. **return** x with probability $\min\left\{1, \frac{f(x)}{f(p)}\right\}$;
   return p with the remaining probability.

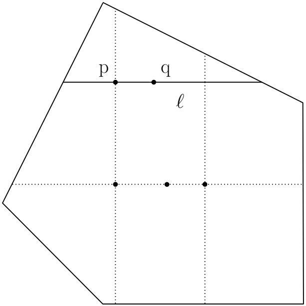


Hit-and-Run

Hit and Run(K, p, f): convex $K \subset \mathbb{R}^d$, point $p \in P$, $f : \mathbb{R}^d \to \mathbb{R}_+$

1. Pick uniformly a line ℓ through p.
2. **return** a random point on the chord $\ell \cap K$ chosen from the distribution $\pi_{\ell,f}$ restricted in $K \cap \ell$.



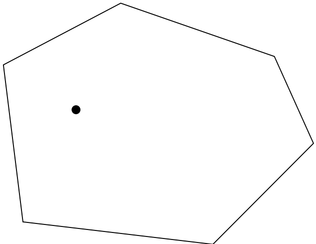


- **Q**: How do we compute $\ell \cap K$? Can we do it *exactly*?

Billiard walk - Uniform case

BW(K, p_i, τ, R) [Polyak'14]

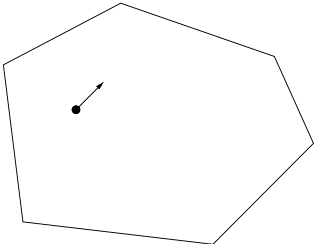
1. Generate the length of the trajectory $L = -\tau \ln \eta$, $\eta \sim U(0,1)$.
2. Pick a uniform direction v to define the trajectory. then the direction becomes $v \leftarrow v - 2\langle v, s\rangle$.
3. If the trajectory meets a boundary with internal normal $s,\ ||s|| = 1$,
4. **return** the end of the trajectory as p_{i+1}.
   If the number of reflections exceeds R, then **return** $p_{i+1} = p_i$.



Billiard walk - Uniform case

BW(K, p_i, τ, R) [Polyak'14]

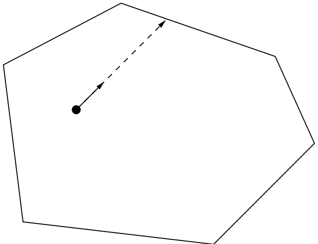
1. Generate the length of the trajectory $L = -\tau \ln \eta$, $\eta \sim U(0,1)$.
2. Pick a uniform direction v to define the trajectory. then the direction becomes $v \leftarrow v - 2\langle v, s\rangle$.
3. If the trajectory meets a boundary with internal normal $s,\ ||s|| = 1$,
4. **return** the end of the trajectory as p_{i+1}.
   If the number of reflections exceeds R, then **return** $p_{i+1} = p_i$.



Billiard walk - Uniform case

BW(K, p_i, τ, R) [Polyak'14]

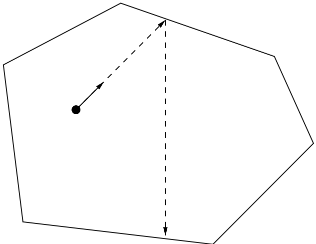
1. Generate the length of the trajectory $L = -\tau \ln \eta$, $\eta \sim U(0,1)$.
2. Pick a uniform direction v to define the trajectory. then the direction becomes $v \leftarrow v - 2\langle v, s \rangle$.
3. If the trajectory meets a boundary with internal normal $s, \ ||s|| = 1$,
4. **return** the end of the trajectory as p_{i+1}.
   If the number of reflections exceeds R, then **return** $p_{i+1} = p_i$.



Billiard walk - Uniform case

BW(K, p_i, τ, R) [Polyak'14]

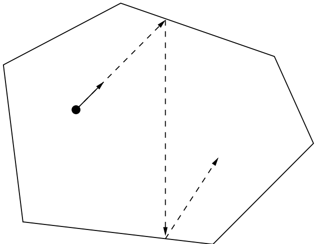
1. Generate the length of the trajectory $L = -\tau \ln \eta$, $\eta \sim U(0, 1)$.
2. Pick a uniform direction v to define the trajectory. then the direction becomes $v \leftarrow v - 2\langle v, s\rangle$.
3. If the trajectory meets a boundary with internal normal $s,\ ||s|| = 1$,
4. **return** the end of the trajectory as p_{i+1}.
   If the number of reflections exceeds R, then **return** $p_{i+1} = p_i$.



Billiard walk - Uniform case

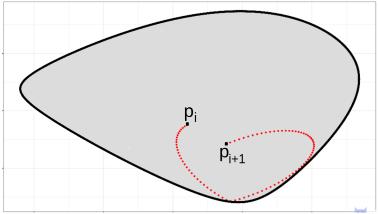
BW(K, p_i, τ, R) [Polyak'14]

1. Generate the length of the trajectory $L = -\tau \ln \eta$, $\eta \sim U(0,1)$.
2. Pick a uniform direction v to define the trajectory. then the direction becomes $v \leftarrow v - 2\langle v, s\rangle$.
3. If the trajectory meets a boundary with internal normal $s,\ ||s|| = 1$,
4. **return** the end of the trajectory as p_{i+1}.
   If the number of reflections exceeds R, then **return** $p_{i+1} = p_i$.

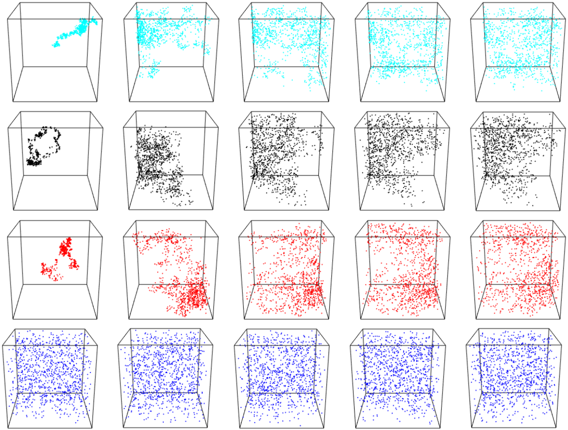


Hamiltonian Monte Carlo

- Similar to billiard walk but with non-linear trajectory
- Trajectory is defined by Hamiltonian dynamics simulated using a time-reversible and volume-preserving numerical integrator (typically the leapfrog integrator)
- Reflected: The trajectory stays inside K by using boundary reflections.
- Riemannian: Using the barrier of K the trajectory is always inside K.



Mixing time experiment (uniform case)



- Uniform sampling from the hypercube $[-1, 1]^{200}$ and projection to $\mathbb{R}^3$.
- Rows: Ball Walk, Coordinate Directions Hit and Run, Random Directions Hit and Run, Billiard Walk.
- Columns: walk length, $\{1, 50, 100, 150, 200\}$

Complexity bounds

Year & Authors	Random walk	Mixing time*	Distribution
[Smith: 1986]	Hit-and-Run	$\widetilde{O}(d^3)$	any LC
[Berbee, Smith: 1987]	Coordinate Hit-and-Run	$\widetilde{O}(d^{10})$	any LC
[Lovasz,Simonovits'90]	Ball walk	$\widetilde{O}(d^3)$	any LC
[Kannan,Narayanan'12]	Dikin walk	$\widetilde{O}(d^2)$	uniform (H-polytope)
[Polyak,Dabbene'14]	**Billiard walk**	??	uniform
[Afshar,Domke'15]	**Reflective HMC**	??	any LC (polytopes)
[Lee,Vempala'16]	Geodesic walk	$O(md^{3/4})$	uniform (H-polytope)
[Lee,Vempala'17]	Remannian HMC	$\widetilde{O}(md^{2/3})$	any LC (H-polytopes)
[Chen,Dwivedi,Wainwright,Yu'19]	John walk	$\widetilde{O}(d^{5/2})$	uniform (H-polytope)
[Chen,Dwivedi,Wainwright,Yu'19]	Vaidya walk	$O(m^{1/2}d^{3/2})$	uniform (H-polytope)

- Cost per sample: *cost per step* $\times$ *mixing time (#steps)*.
- The *cost per step* depends on the convex body.
- Hit-and-Run (HR): widely used & well studied.
- Coordinate Hit-and-Run (CDHR): seems more efficient than HR in practice.
- Most existing software uses either CDHR or HR (H-polytopes).

MCMC Convergence Diagnostics

How can we evaluate the quality of a sample obtained by a random walk?

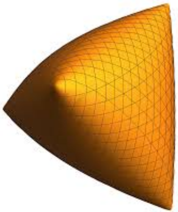
- [Convergence diagnostics for Markov chain Monte Carlo, Vivekananda Roy, '19].
- [Revisiting the Gelman-Rubin Diagnostic, Dootika Vats, Christina Knudson, '20].

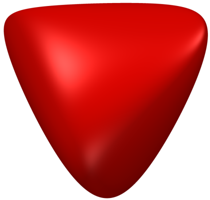
A MCMC convergence diagnostic can also be used as a termination criterion for sampling.

Examples: Effective Sample Size (ESS) and psrf (or Rhat)

Convex bodies

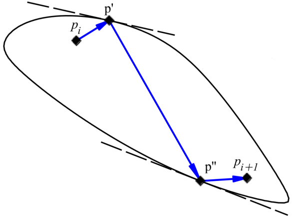
- H-polytopes: system of linear inequalities
- V-polytopes: convex hull of point sets
- Minkowski sums of polytopes
- Spectrahedra: feasible sets of linear matrix inequalities





Geometric and algebraic oracles

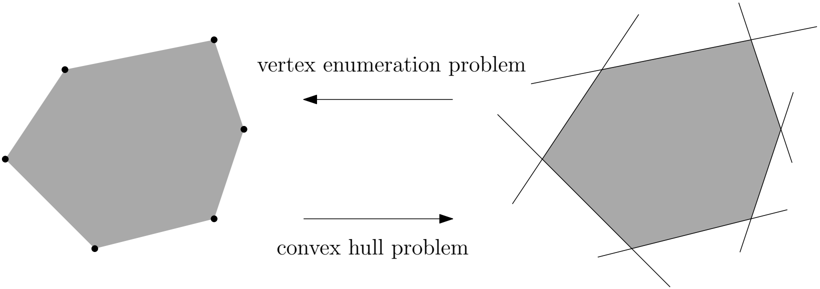
- Membership oracle (Ball walk)
- Boundary (intersection) oracle (HnR)
- Reflection oracle (Billiard, ReHMC)
- Optimization oracle (Minkowski sums, Secondary polytopes)



Explicit Polytope Representations

A convex polytope $P \subseteq \mathbb{R}^d$ can be represented as the

1. convex hull of a pointset $\{p_1, \ldots, p_m\}$ (V-representation)
2. intersection of halfspaces $\{h_1, \ldots, h_n\}$ (H-representation)



Faces of polytopes: vertices, edges, ..., facets

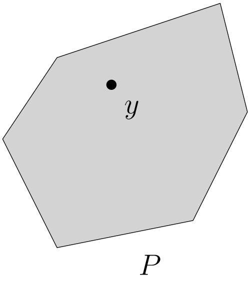
Implicit Polytope Representation (Oracles)

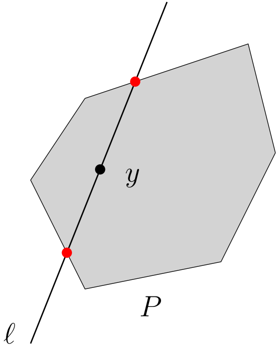
Membership oracle

Given point $y \in \mathbb{R}^d$, return yes if $y \in P$ otherwise return no.

Boundary oracle

Given point $y \in P$ and line ℓ goes through y return the points $\ell \cap \partial P$





Outline

Volume computation

Random Sampling from Polytopes

Volume revisited (randomized)

Polytopes and Applications

Problem complexity

Input: Polytope $P := \{x \in \mathbb{R}^d \mid Ax \leq b\}$ $A \in \mathbb{R}^{m\times d}$, $b \in \mathbb{R}^m$

Output: Volume of P

Problem complexity

Input: Polytope $P := \{x \in \mathbb{R}^d \mid Ax \leq b\}$ $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$

Output: Volume of P

Complexity

- #P-hard for vertex and for halfspace repres. [DyerFrieze'88]

Problem complexity

Input: Polytope $P := \{x \in \mathbb{R}^d \mid Ax \leq b\}$ $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$

Output: Volume of P

Complexity

- #P-hard for vertex and for halfspace repres. [DyerFrieze'88]
- open if both vertex (V-rep) & halfspace (H-rep) representation is available

Problem complexity

Input: Polytope $P := \{x \in \mathbb{R}^d \mid Ax \leq b\}$ $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$

Output: Volume of P

Complexity

- #P-hard for vertex and for halfspace repres. [DyerFrieze'88]
- open if both vertex (V-rep) & halfspace (H-rep) representation is available
- no deterministic poly-time algorithm can compute the volume with less than exponential relative error [Elekes'86]

Problem complexity

Input: Polytope $P := \{x \in \mathbb{R}^d \mid Ax \leq b\}$ $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$

Output: Volume of P

Complexity

- #P-hard for vertex and for halfspace repres. [DyerFrieze'88]
- open if both vertex (V-rep) & halfspace (H-rep) representation is available
- no deterministic poly-time algorithm can compute the volume with less than exponential relative error [Elekes'86]
- randomized poly-time approximation of volume of a convex body with high probability and arbitrarily small relative error [DyerFriezeKannan'91]
  $O^*(d^{23}) \to O^*(m^2 d^{\omega - 1/3})$ [LeeVempala'18],
  $O^*(md^{4.5} + md^4)$ [MangoubiVishnoi'19]

Randomized algorithms

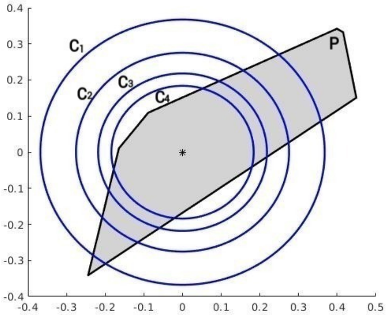
Volume algorithms parts

1. **M**ultiphase **M**onte **C**arlo (MMC)
   e.g. Sequence of balls, Annealing of functions
2. Sampling via geometric random walks
   e.g. grid-walk, ball-walk, hit-and-run, billiard walk

Notes:

- MMC (1) at each phase solves a sampling problem (2)
- geometric random walks are (most of the times) Marcov chains where each "event" is a d-dimensional point
- Algorithmic complexity is polynomial in d [Dyer, Frieze, Kannan'91]

Multiphase Monte Carlo

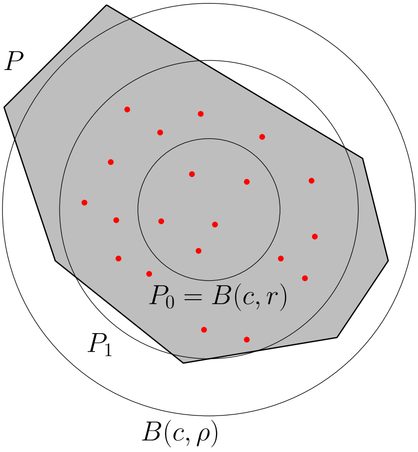


- Sequence of convex bodies $C_1 \supseteq \cdots \supseteq C_m$ intersecting P, then:

$$\text{vol}(P) = \text{vol}(P_m)\frac{\text{vol}(P_{m-1})}{\text{vol}(P_m)}\cdots\frac{\text{vol}(P_1)}{\text{vol}(P_2)}\frac{\text{vol}(P)}{\text{vol}(P_1)}$$

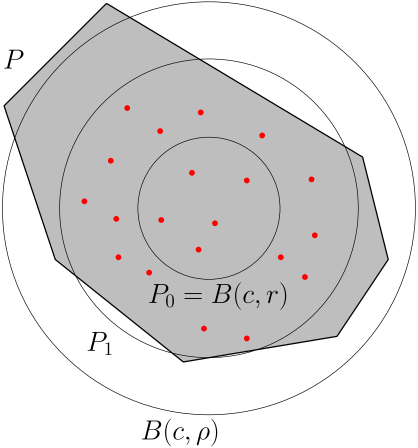
  where $P_i = C_i \cap P$ for $i = 1, \ldots, m$.
- Estimate ratios by sampling.

Multiphase Monte Carlo



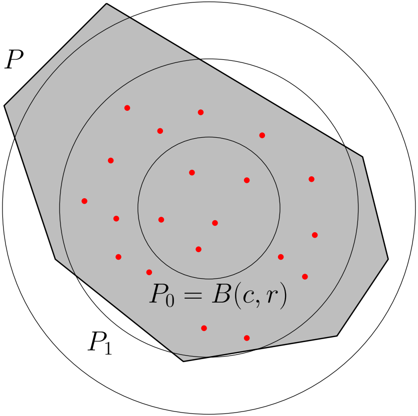
- Sequence of k cocentric balls, $B_0 = B(c, r) \subseteq P \subseteq B(c, \rho) = B_k$
- Set $P_i = P \cap B_i$
- Estimate $\frac{vol(P_1)}{vol(P_0)}, \frac{vol(P_2)}{vol(P_1)} \ldots$ via sampling
- $vol(P) = vol(P_0) \prod_{i=1}^{k} \frac{vol(P_i)}{vol(P_{i-1})}$
- How large is k?

Multiphase Monte Carlo



- $B(c, 2^{i/d})$, $i = \alpha, \alpha + 1, \ldots, \beta$,
  $\alpha = \lfloor d \log r \rfloor$, $\beta = \lceil d \log \rho \rceil$
- $P_i := P \cap B(c, 2^{i/d}),\ i = \alpha, \alpha + 1, \ldots, \beta$
  $P_\alpha = B(c, 2^{\alpha/d}) \subseteq B(c, r)$
- $k = d \log(\rho/r)$ where ρ/r is the "sandwitching ratio"

Multiphase Monte Carlo



- $B(c, 2^{i/d}),\ i = \alpha, \alpha+1, \ldots, \beta,$
  $\alpha = \lfloor d \log r \rfloor,\ \beta = \lceil d \log \rho \rceil$
- $P_i := P \cap B(c, 2^{i/d}),\ i = \alpha, \alpha+1, \ldots, \beta$
  $P_\alpha = B(c, 2^{\alpha/d}) \subseteq B(c, r)$
- $k = d \log(\rho/r)$ where ρ/r is the "sandwitching ratio"

Using sampling the polytope can be transformed into "near isotropic position" such that $\rho/r = O(d)$ [Lovász et al.'97]

Complexity [KannanLS'97]

Assuming $B(c, 1) \subseteq P \subseteq B(c, \rho)$, the volume algorithm returns an estimation of vol(P), which lies between $(1-\epsilon)$vol(P) and $(1+\epsilon)$vol(P) with probability $\geq 3/4$, making

$$O^*(d^5)$$

oracle calls, where ρ is the radius of a bounding ball for P.

Techniques:
Isotropic sandwitching: $O^*(\sqrt{d})$ and ball walk.

Runtime steps

- generates $d \log d$ balls
- generate $N = 400\epsilon^{-2} d \log d$ random points in each ball $\cap P$
- each point is computed after $O^*(d^3)$ random walk steps

Multiphase Monte Carlo: general case

Let a sequence of functions $\{f_0, \ldots, f_m\}$, $f_i : \mathbb{R}^d \to \mathbb{R}$. Then,

$$\text{vol}(P) = \int_P dx = \int_P f_m(x)dx \; \frac{\int_P f_{m-1}(x)dx}{\int_P f_m(x)dx} \cdots \frac{\int_P f_0(x)dx}{\int_P f_1(x)dx} \; \frac{\int_P dx}{\int_P f_0(x)dx}$$

Then select f_i s.t.,

- The number of phases, m, is as small as possible.
- Each integral ratio can be efficiently estimated by sampling from $\pi \propto f_i$ restricted to P (using geometric random walks).
- There is a closed formula for $\int_P f_m(x)dx$.

complexity = #phases × #points per phase × cost per point

State-of-the-art

Theory:

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Seq. of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Seq. of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

State-of-the-art

Theory:

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Seq. of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Seq. of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

Software:

1. [Emiris, F'14] Sequence of balls + coordinate hit-and-run
2. [Cousins, Vempala'16] Gaussian cooling + hit-and-run
3. [Chalikis, Emiris, F'20] Convex body annealing + billiard walk

State-of-the-art

Theory:

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Seq. of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Seq. of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

Software:

1. [Emiris, F'14] Sequence of balls + coordinate hit-and-run
2. [Cousins, Vempala'16] Gaussian cooling + hit-and-run
3. [Chalikis, Emiris, F'20] Convex body annealing + billiard walk

Notes:

- (2) is (theory + practice) faster than (1)
- (1),(2) efficient only for H-polytopes
- (3) efficient also for V-,Z-polytope, non-linear convex bodies

Outline

Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M

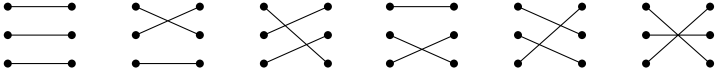
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$

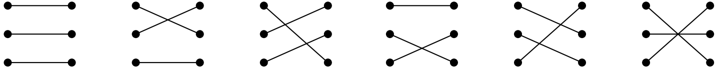
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$

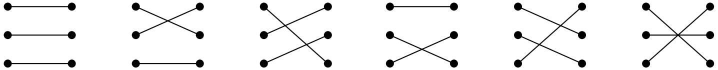
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 

Birkhoff polytopes

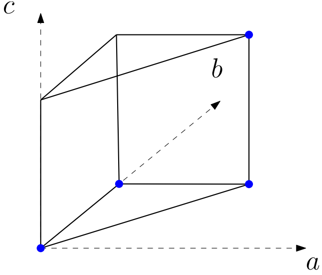
- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 
- # faces of B_3: 6, 15, 18, 9; $\mathrm{vol}(B_3) = 9/8$

Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 
- $\#$ faces of B_3: 6, 15, 18, 9; $\mathrm{vol}(B_3) = 9/8$
- there exist formulas for the volume [deLoera et al '07] but values only known for $n \leq 10$ after 1yr of parallel computing [Beck et al '03]

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



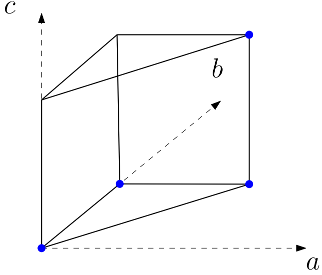
a, b, c

partial order: $a < b$

3 linear extensions: abc, acb, cab

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



a, b, c

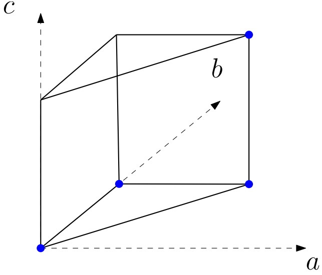
partial order: $a < b$

3 linear extensions: abc, acb, cab

- # linear extensions = volume of order polytope $\cdot\ n!$ [Stanley'86]

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



a, b, c

partial order: $a < b$

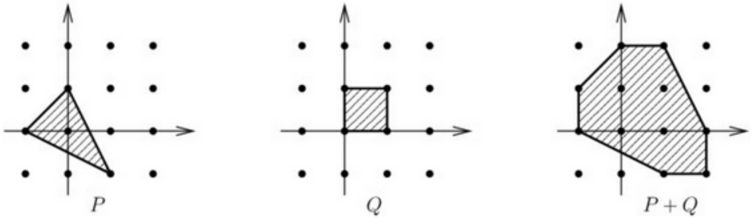
3 linear extensions: abc, acb, cab

- # linear extensions = volume of order polytope $\cdot\, n!$ [Stanley'86]
- Counting linear extensions is #P-hard [Brightwell'91]

Minkowski sum

The Minkowski sum of two convex sets P and Q is:

$$P + Q = \{p + q \mid p \in P, q \in Q\}$$



Volume of zonotopes (sums of segments) is used to test methods for order reduction which is important in several areas: autonomous driving, human-robot collaboration and smart grids

Mixed volume

Let $P_1, P_2, \ldots, P_d$ be polytopes in $\mathbb{R}^d$ then the mixed volume is

$$M(P_1, \ldots, P_d) = \sum_{I \subseteq \{1,2,\ldots,d\}} (-1)^{(d-|I|)} \cdot \mathrm{Vol}(\sum_{i \in I} P_i)$$

where the sum is the Minkowski sum.

Mixed volume

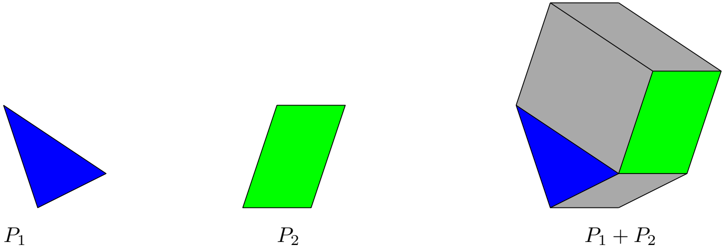
Let $P_1, P_2, \ldots, P_d$ be polytopes in $\mathbb{R}^d$ then the mixed volume is

$$M(P_1, \ldots, P_d) = \sum_{I \subseteq \{1,2,\ldots,d\}} (-1)^{(d-|I|)} \cdot \mathrm{Vol}(\sum_{i \in I} P_i)$$

where the sum is the Minkowski sum.

Example

For $d = 2$: $M(P_1, P_2) = \mathrm{Vol}(P_1 + P_2) - \mathrm{Vol}(P_1) - \mathrm{Vol}(P_2)$



Applications

Computing integrals for AI

- In Weighted Model Integration (WMI), given is a SMT formula and a weight function, then we want to compute the weight of the SMT formula.

- e.g. SMT formula:

$$(A \ \& \ (X > 20) \mid (X > 30)) \ \& \ (X < 40)$$

  Boolean formula + comparison operations. Let X has a weight function of $w(X) = X^2$ and $w(A) = 0.3$.

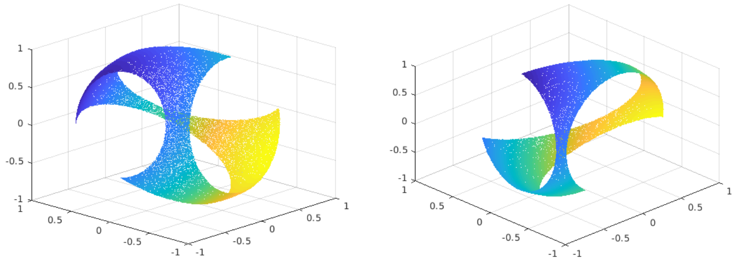
- WMI answers the question of the weight of this formula i.e. integration of a weight function over convex sets.

- [P.Z.D. Martires et al.2019]

Applications in finance

Portfolio analysis

- The set of portfolios (investments in a collection of stocks) is a simplex.
- Constraints on investments yield a general polytope.
- Portfolios with same volatility (the degree of variation of a trading price series over time) lie on an ellipsoid.

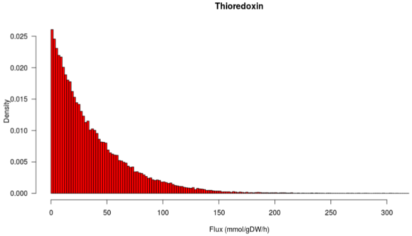


Randomized geometric tools for anomaly detection in stock markets [Bachelard,Chalkis,F,Tsigaridas'23]

Applications in structural biology

[Chalkis,F, Tsigaridas, Zafeiropoulos]

- Metabolic networks model the reactions of metabolites in an organim or system.
- Each reaction has a flow or rate called flux.
- The set of states of the network where fluxes are in balance (rate of production = rate of consumption) is a convex polytope.
- Sampling from polytope yield probability densities for reaction fluxes (example: thioredoxin)



Current and future state

https://github.com/GeomScale

Problem	current	future	description
volume computation	✓		8 algo. / thousands of dimensions / fastest practical estimation
sampling distributions uniform / gaussian/ Exp	✓		4 algo. / thousands of dimensions
log-concave densities	✓		HMC / Langevin Diffusion
sparsity		✓	lazy rounding / reflection walks (HMC & billiard)
convex optimization Semidefinite Programming	✓		special cases better than `SDPA` / working to improve
Linear Programming		✓	goal: best open source
multivariate integration simple MC integration	✓		hundreds of dimensions
importance sampling		✓	goal: best open source approximation
Preprocessing	✓		6 rounding algo. / 4 MCMC diagnostics

More open problems and future directions

- MCMC integration/volume with guarantees in practice (needed in ML/weighted model integration)
- Exploit sparsity (Vempala et al. - crHMC, Chen et al. - PolytopeWalk)
- Sampling on the boundary (applications in finance/biology)
- Ranomized SDP/LP solver
- Applications to counting problems (e.g. # LE)
- More efficient volume (reHMC)

GeomScale org

C++ library: sampling, integration/volume from convex bodies

Python interface with extra utilities for metabolic network analysis (FBA, copulas, visualization)

R interface with extra utilities for finance (portfolio analysis)

NumFOCUS Affiliated Project.

Support from an open source community.