High-dimensional sampling and volume computation

Vissarion Fisikopoulos

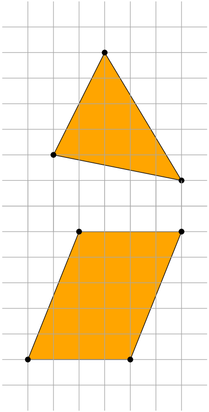
Dept. of Informatics & Telecommunications, University of Athens

KULeuven, 30/01/2020

Our problem

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

Some elementary polytopes (simplex, cube) have simple determinantal formulas.



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

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

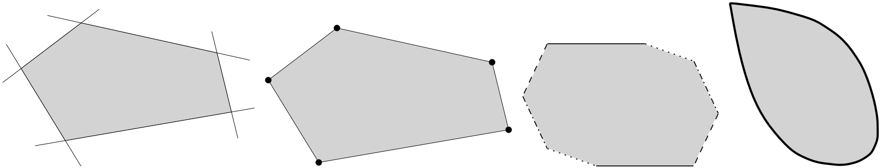
Convex bodies

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

V-polytope : P is the convex hull of a set of points in $\mathbb{R}^d$

Z-polytope : Minkowski sum of segments (projections of k-cubes)

LMI : $P = A_0 + x_1 A_1 + x_2 A_2 + \cdots + x_d A_d \succeq 0$, where A_i: symmetric matrices, $B \succeq 0$: B is positive semidefinite

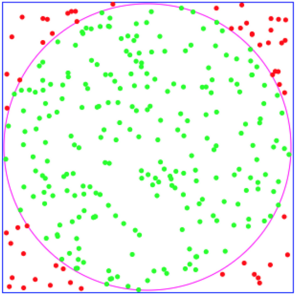


First thoughts for volume computation

- Triangulation (or sign decomposition) methods – exponential size in d

First thoughts for volume computation

- Triangulation (or sign decomposition) methods – exponential size in d
- Sampling/rejections techniques (sample from bounding box) fail in high dimensions



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

Volume computation is hard!

- #P-hard for V-, H-, Z-polytopes [DyerFrieze'88]
- no deterministic poly-time algorithm can compute the volume with less than exponential relative error (oracle model) [Elekes'86]
- open problem if V-polytope and H-polytope representations available

Randomized algorithms

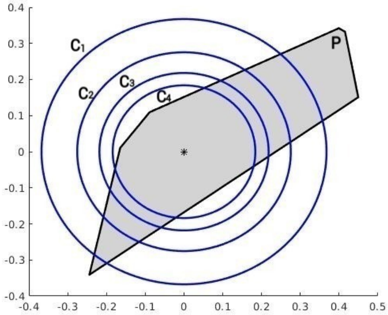
Volume algorithms parts

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 computes a ratio of integrals or volumes via sampling (2)
- geometric random walks are Markov chains where each "event" is a d-dimensional point
- Algorithmic complexity is polynomial in d [Dyer, Frieze, Kannan'91]

Multiphase Monte Carlo



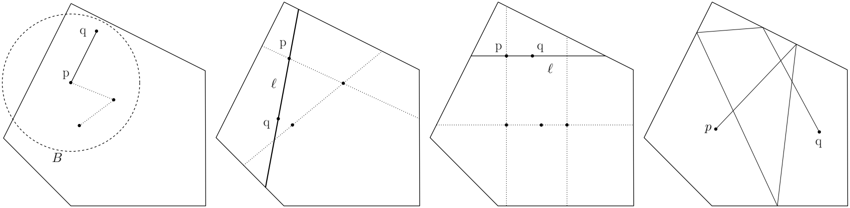
- Sequence of convex bodies $C_1 \supseteq \cdots \supseteq C_m$ intersecting P, then:

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

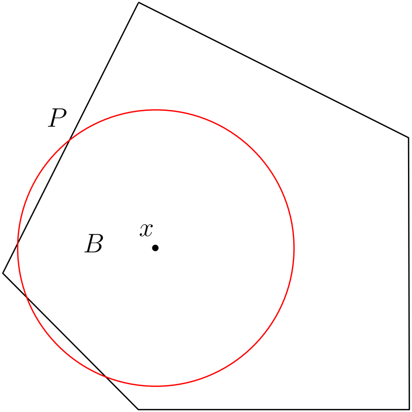
  where $P_i = C_i \cap P$ for $i = 1, \ldots, m$.
- Estimate ratios by sampling.

Four random walks

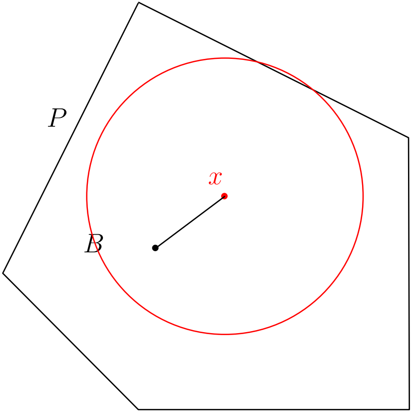
- Ball walk
- Random directions hit and run (rdhr)
- Cooridnate directions hit and run (cdhr)
- Billiard walk



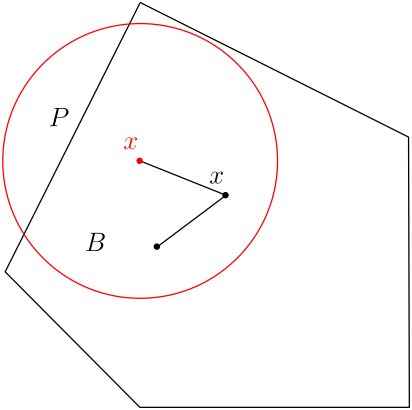
Ball walk



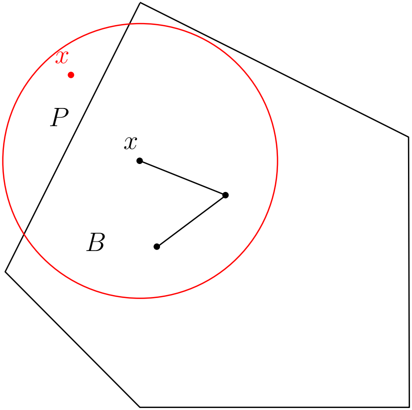
Ball walk



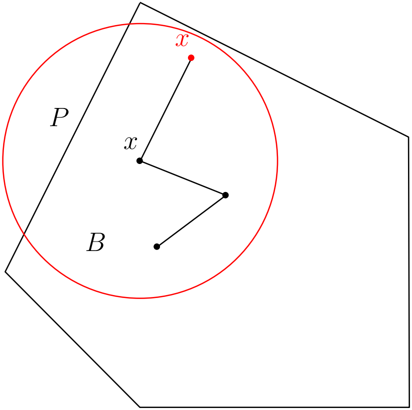
Ball walk



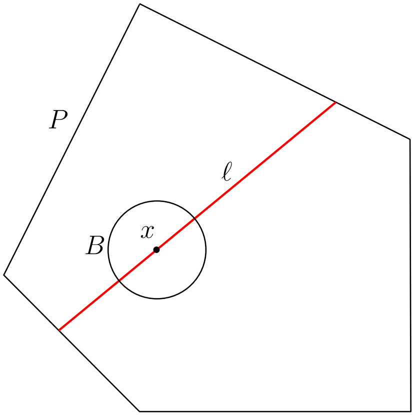
Ball walk



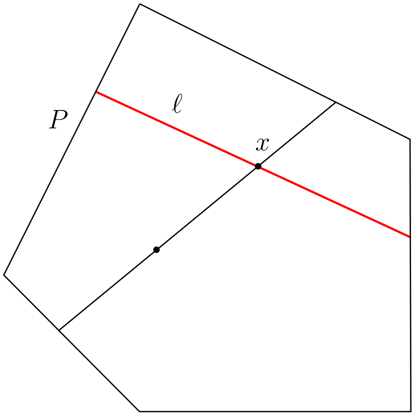
Ball walk



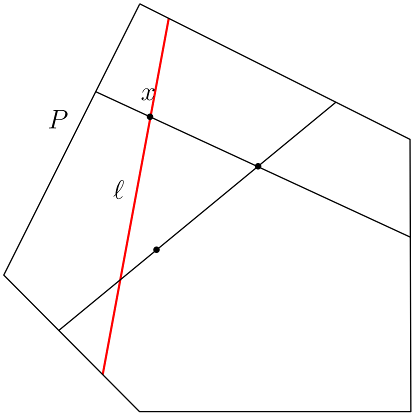
Random directions hit and run



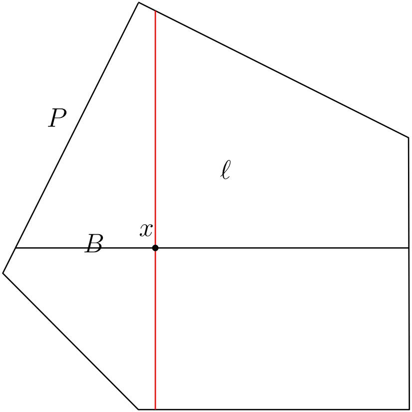
Random directions hit and run



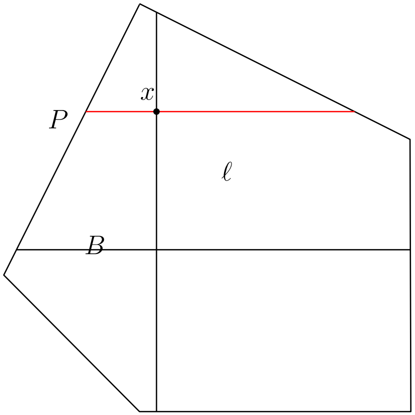
Random directions hit and run



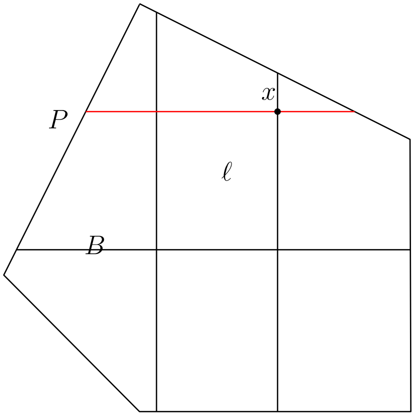
Coordinate directions hit and run



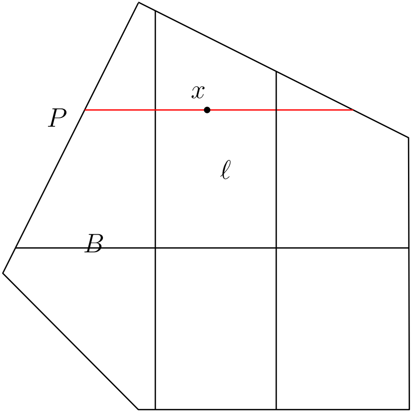
Coordinate directions hit and run



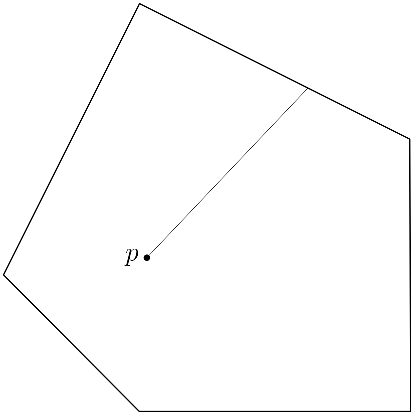
Coordinate directions hit and run



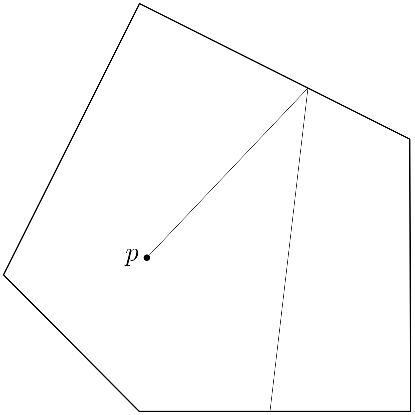
Coordinate directions hit and run



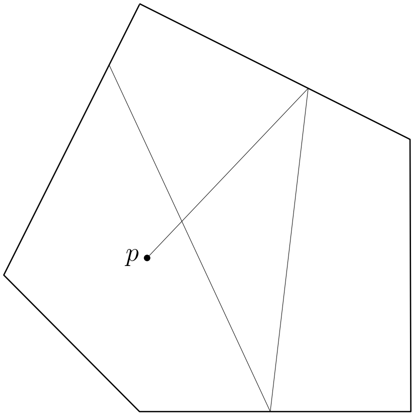
Billiard walk



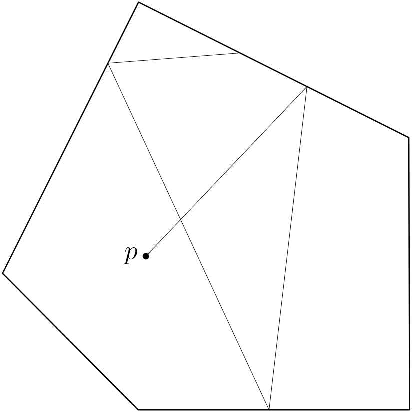
Billiard walk



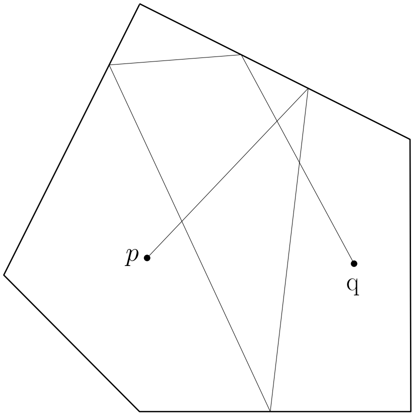
Billiard walk



Billiard walk



Billiard walk



Complexity of random walks

Year & Authors	Random walk	Mixing time	cost per step
[Berbee et al.'87]	Coordinate Hit-and-Run	??	$O(m)$
[Lovasz et al.'06]	Hit-and-Run	$O^*(d^3)$	$O(md)$
[Kannan et al.'09]	Dikin walk	$O(md)$	$O(md^2)$
[Polyak et al.'14]	Billiard walk	??	$O(mR + md)$
[Lee et al.'16]	Geodesic walk	$O(md^{3/4})$	$O(md^2)$
[Lee et al.'17]	Ball walk	$O^*(d^{2.5})$	$O(md)$
[Chen et al.'17]	Vaidya walk	$O(m^{1/2}d^{3/2})$	$O(md^2)$
[Lee et al.'17]	Riemmanian HMC	$O(md^{2/3})$	$O(md^2)$
[Chevallier et al.'18]	HMC with reflections	??	$O(md)$
[Mangoubi et al.'19]	sublinear Ball walk	$O(d^{4.5})$	$\sim O(m)$

- Mixing times are unrealistically high for practical purposes
- Billiard walk, CDHR and HMC with reflections seems the most efficient in practice but there is not guarantee on the mixing time

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]	$O^*(md^{\frac{2}{3}})$	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]	$O^*(md^{\frac{2}{3}})$	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]	$O^*(md^{\frac{2}{3}})$	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
- C++ implementation of (2) $\times$10 faster than original (MATLAB)

Convex body annealing

New features

- MMC using convex bodies (balls or H-polytopes that "fit well" and are "easy" to sample from)
- New annealing method. Bound each ratio $\mathrm{vol}(P_{i+1})/\mathrm{vol}(P_i)$ to a given interval with high probability thus reduces the sequence of bodies
- Billiard walk (constant number of steps in practice)

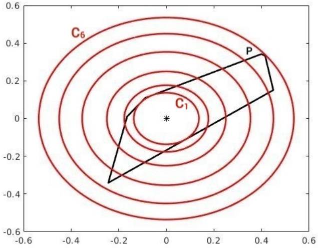
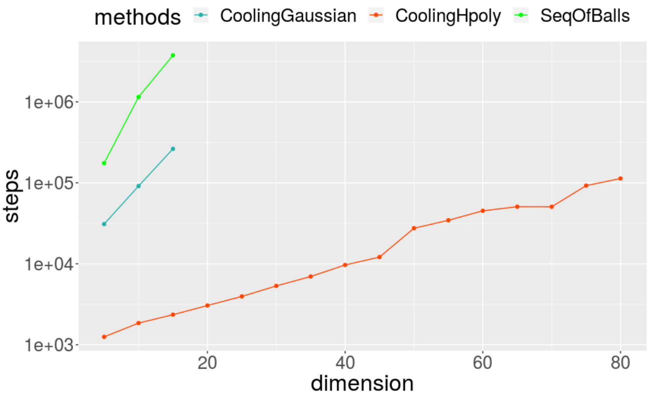


Figure. Sequence of balls vs. ball annealing

Comparison with state-of-the-art

Zonotopes

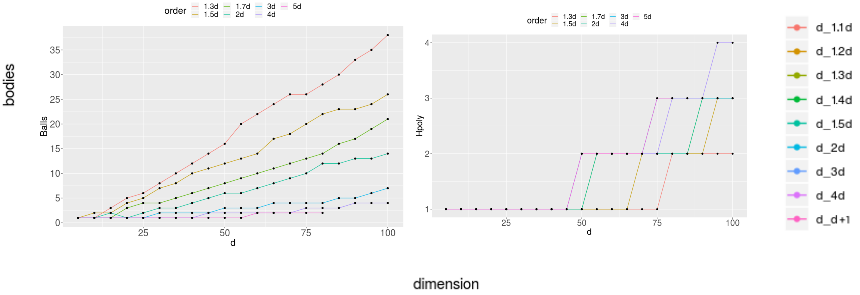


The number of steps for random zonotopes of order 2

- Our method is asymptotically better
- `CoolingGaussian` and `SeqOfBalls` takes $> 2hr$ for $d > 15$.
- for higher orders the difference is larger

Zonotopes

Number of phases



Two types of bodies: balls (left) symmetric H-polytopes (right)

- For low order (i.e. $\#generators/d$) zonotopes, ≤ 4, the number of bodies is smaller than the case of using balls
- For balls the number of phases reduces as the order increases for a fixed dimension

Zonotopes

Performance

Experimental results for zonotopes.						
z-d-k	*Body*	*order*	*Vol*	*m*	*steps*	*time*(sec)
z-5-500	Ball	100	4.63e+13	1	0.1250e+04	22
z-20-2000	Ball	100	2.79e+62	1	0.2000e+04	1428
z-50-65	Hpoly	1.3	1.42e+62	1	1.487e+04	173
z-50-75	Hpoly	1.5	2.96e+66	2	1.615e+04	253
z-100-150	Hpoly	1.5	2.32+149	3	15.43e+04	2992
z-60-180	Hpoly	3	8.71e+111	2	5.059e+04	417
z-100-200	Hpoly	3	5.27e+167	3	15.25e+04	2515

- *z-d-k*: random zonotope in dimension d with k generators; *Body*: the type of body used in MMC; m: number of bodies in MMC
- Used to evaluate zonotope approximation methods in engineering [Kopetzki'17]

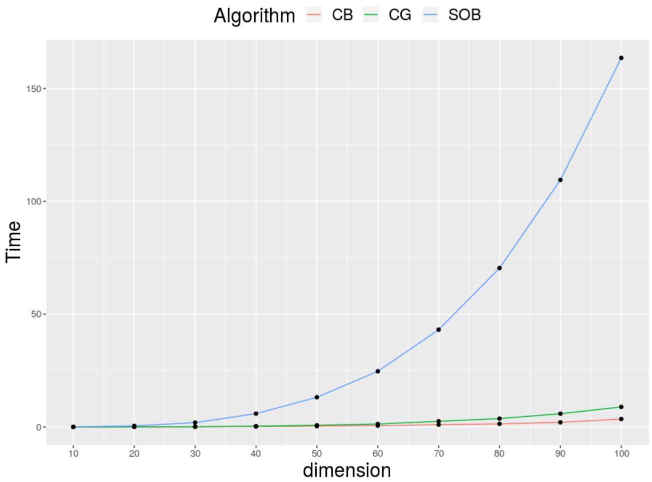
V-polytopes

P	Vol	m	steps	error	time(sec)	exact Vol	exact time
cross-60	1.27e-64	1	20.6e+03	0.08	60	−−	−−
cross-100	1.51e-128	2	94.2e+03	0.11	406	−−	−−
Δ-60	1.08e-82	10	77.4e+04	0.1	899	1.203-82	0.02
Δ-80	1.30e-119	13	187e+04	0.07	4140	1.39e-119	0.07
cube-10	1052.4	1	1851	0.03	54	−−	−−
cube-13	7538.2	1	2127	0.08	2937	−−	−−
rv-10-80	3.74e-03	1	0.185e+04	0.08	3	3.46e-03	7
rv-10-160	1.59e-02	1	0.140e+04	0.06	6	1.50e-03	59
rv-15-30	2.73e-10	1	0.235e+04	0.02	3	2.79e-10	2
rv-15-60	4.41e-08	1	0.235e+04	−−	6	−−	−−
rv-20-2000	2.89e-07	1	0.305e+04	−−	457	−−	−−
rv-80-160	5.84e-106	3	11.3e+04	−−	891	−−	−−
rv-100-200	1.08e-141	4	24.5e+04	−−	2312	−−	−−

time: the average time in seconds; ex. Vol: the exact volume; ex. time: the time in seconds for the exact volume computation i.e. `qhull` in `R` (package `geometry`); m is the number of phases. $--$ implies that the execution failed due to memory issues or exceeded 1 hr.

Performance

H-polytopes

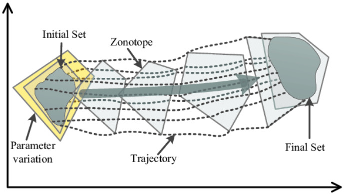


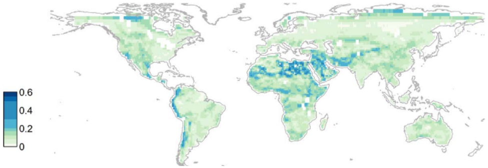
Sequence Of Balls (SOB), Cooling Gaussians (CG), Cooling Balls (CB)

Applications

Biogeography & engineering

- Volume of zonotopes is used to test methods for order reduction which is important in several areas: autonomous driving, human-robot collaboration and smart grids [Althoff et al.]
- Volumes of intersections of polytopes are used in bio-geography to compute biodiversity and related measures e.g. [Barnagaud, Kissling, Tsirogiannis, F, Villeger, Sekercioglu'17]





Applications

Combinatorics & Machine Learning

- Volume can be used for counting linear extensions of a partially ordered set. This arises in sorting [Peczarski 2004], sequence analysis [Mannila et al.2000], convex rank tests [Morton et al.2009], preference reasoning [Lukasiewicz et al.2014], partial order plans [Muise et al.2016], learning graphical models [Niinimäki et al. 2016] See also [Talvitie et al.AAAI'2018]
- *e.g.* elements a, b, c
  partial order $a < c$
  3 linear extensions: abc, acb, bac

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:

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

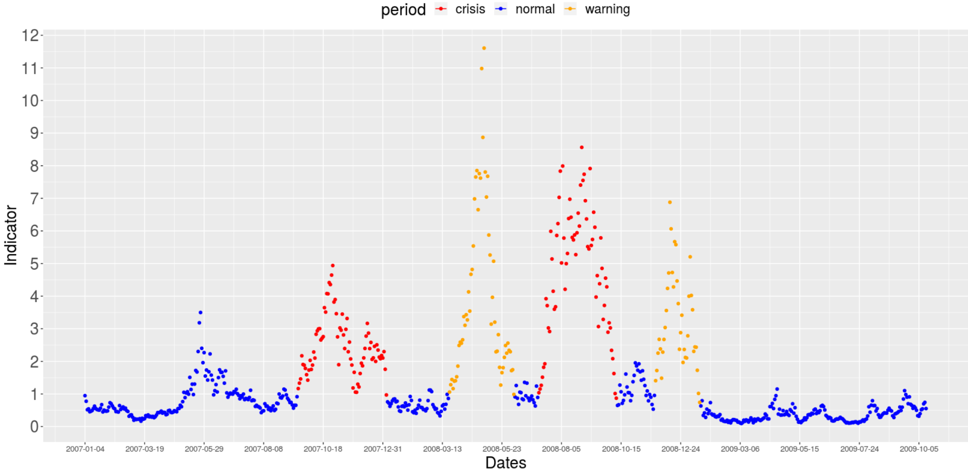
Boolean formula + comparison operations. Let X has a weight function of $w(X) = X^2$ and $w(A) = 0.3$.

- 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

When is the next financial crisis?



Cales, Chalkis, Emiris, Fisikopoulos - Practical volume computation of structured convex bodies, and an application to modeling portfolio dependencies and financial crises, SoCG 2018

VolEsti package: sampling and volume estimation

https://github.com/GeomScale/volume_approximation

- `C++`, `R`-interface, `python` bindings (limited)
- open source, LGPL3
- since 2014, CGAL (not any more), Eigen, LPSolve, Boost
- 3 volume algorithms, 4 sampling algorithms
- design: mix of obj oriented + templates
- `C++11` dependence (mostly `C++03`)
- main developers: Vissarion Fisikopoulos, Apostolos Chalkis

VolEsti package

- R package on CRAN
  https://cran.r-project.org/package=volesti
- Documentation
  https://github.com/GeomScale/volume_approximation/blob/develop/README.md

How to contribute (github account is needed):

- first star and fork the repo :D
- then follow the contribution tutorial

VolEsti on Google Summer of Code 2020

- Applying this year as an organization for GSoC 2020.
- See this wiki for more details.
- Tentative plan:
  - MCMC integration
  - sampling for structural biology
  - randomized algorithms for convex optimization
  - sampling in higher dimensions (reach the thousands)
- Communication channels: gitter, `geomscale-gsoc@googlegroups.com`

VolEsti tutorial

https://vissarion.github.io/tutorials/volesti_tutorial_pydata.html

CRAN mode (recommended!)

- in Rtudio install CRAN package "volesti"
- follow this https://github.com/GeomScale/volume_approximation/blob/tutorial/tutorials/volesti_tutorial.Rmd

develop mode

- `git clone git@github.com:GeomScale/volume_approximation.git`
- `git checkout tutorial`
- in Rtudio open `R-proj/volesti.Rproj` and then click `build source Package` and then `Install and Restart` in `Build` tab at the menu bar.
- follow this https://github.com/GeomScale/volume_approximation/blob/tutorial/tutorials/volesti_tutorial_1_1_0.Rmd

Thank you!