

Practical Volume Computation of Structured Convex Bodies, and an Application to Modeling Portfolio Dependencies and Financial Crises*

Ludovic Calès¹


European Commission, Joint Research Centre, Ispra, Italy
ludovic.cales@ec.europa.eu

Apostolos Chalkis²

Department of Informatics & Telecommunications
National & Kapodistrian University of Athens, Greece
achalkis@di.uoa.gr

Ioannis Z. Emiris²

Department of Informatics & Telecommunications
National & Kapodistrian University of Athens, Greece, and
ATHENA Research & Innovation Center, Greece
emiris@di.uoa.gr

 <https://orcid.org/0000-0002-2339-5303>

Vissarion Fisikopoulos

Oracle, Greece
vissarion.fysikopoulos@oracle.com
 <https://orcid.org/0000-0002-0780-666X>

Abstract

We examine volume computation of general-dimensional polytopes and more general convex bodies, defined as the intersection of a simplex by a family of parallel hyperplanes, and another family of parallel hyperplanes or a family of concentric ellipsoids. Such convex bodies appear in modeling and predicting financial crises. The impact of crises on the economy (labor, income, etc.) makes its detection of prime interest for the public in general and for policy makers in particular. Certain features of dependencies in the markets clearly identify times of turmoil. We describe the relationship between asset characteristics by means of a copula; each characteristic is either a linear or quadratic form of the portfolio components, hence the copula can be constructed by computing volumes of convex bodies.

We design and implement practical algorithms in the exact and approximate setting, we experimentally juxtapose them and study the tradeoff of exactness and accuracy for speed. We analyze the following methods in order of increasing generality: rejection sampling relying on uniformly sampling the simplex, which is the fastest approach, but inaccurate for small volumes; exact formulae based on the computation of integrals of probability distribution functions, which are the method of choice for intersections with a single hyperplane; an optimized Lawrence sign decomposition method, since the polytopes at hand are shown to be simple with additional structure; Markov chain Monte Carlo algorithms using random walks based on the hit-and-run paradigm generalized to nonlinear convex bodies and relying on new methods for computing a ball enclosed in the given body, such as a second-order cone program; the latter is experimentally

* The views expressed are those of the authors and do not necessarily reflect official positions of the European Commission.

¹ Calès acknowledges the financial support of the European Commission through its Proof-of-Concept program.

² Chalkis and Emiris are partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 734242 (Project LAMBDA).



extended to non-convex bodies with very encouraging results. Our C++ software, based on CGAL and Eigen and available on [github](#), is shown to be very effective in up to 100 dimensions. Our results offer novel, effective means of computing portfolio dependencies and an indicator of financial crises, which is shown to correctly identify past crises.

2012 ACM Subject Classification Theory of computation → Computational geometry, Theory of computation → Random walks and Markov chains

Keywords and phrases Polytope volume, convex body, simplex, sampling, financial portfolio

Digital Object Identifier 10.4230/LIPIcs.SoCG.2018.19

Related Version A full version of this paper is available at <https://arxiv.org/abs/1803.05861>

1 Introduction

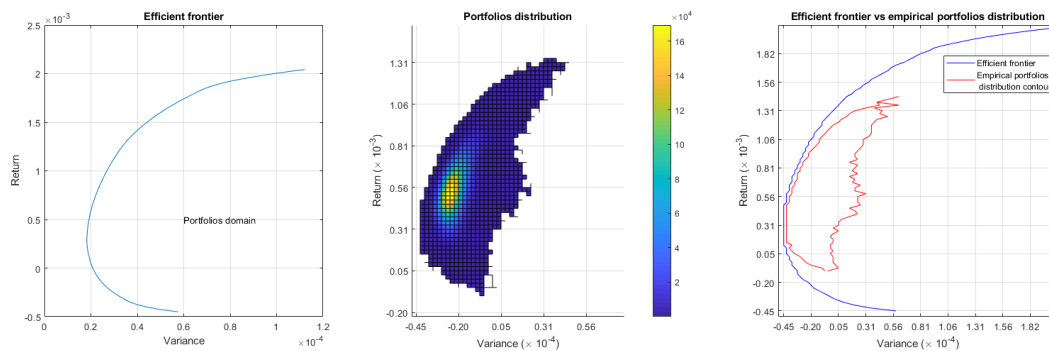
1.1 Financial context and motivation

Modern finance has been pioneered by Markowitz who set a framework to study choice in portfolio allocation under uncertainty [29], and for which he was awarded the Nobel Prize in economics (1990). Markowitz characterized portfolios by their return and their risk which is defined as the variance of the portfolios' returns. An investor would build a portfolio that maximizes its expected return for a chosen level of risk. In the same way, by choosing a level of expected return, an investor may construct a portfolio which minimizes risk. It has since become common for asset managers to optimize their portfolio within this framework. And it has led a large part of the empirical finance research to focus on the so-called efficient frontier which is defined as the set of portfolios presenting the lowest risk for a given expected return. Figure 1 (left panel) presents such an efficient frontier. The region on the left of the efficient frontier represent the portfolios domain.

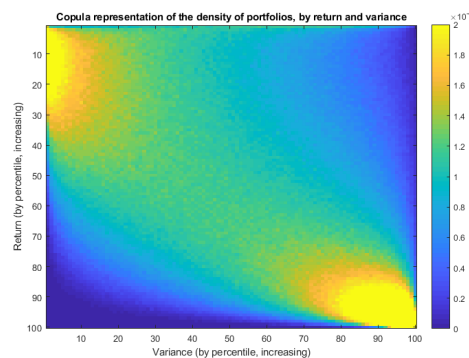
Interestingly, despite the fact that this framework considers the whole set of portfolios, no attention has been given to the distribution of portfolios. Figure 1 (middle panel) presents such distribution where 10.000.000 portfolios have been sampled as presented later in Section 2.1. When comparing the contour of the empirical portfolios distribution, i.e. the region over which at least one random portfolio lies, and the portfolio domain bounded by the efficient frontier in Figure 1 (right panel), we observe that the density of portfolios along the efficient frontier is dim and that most of the portfolios are located in a small region.

We also know from the financial literature that financial markets exhibit 3 types of behavior. In normal times, stocks are characterized by slightly positive returns and a moderate volatility, in up-market times (typically bubbles) by high returns and low volatility, and during financial crises by strongly negative returns and high volatility, see e.g. [5] for details. So, following Markowitz' framework, in normal and up-market times, the stocks and portfolios with the lowest volatility should present the lowest returns, whereas during crises those with the lowest volatility should present the highest returns. These features, also called "stylized facts" in the financial literature, motivate us to describe the time-varying dependency between portfolios' returns and volatility.

However this dependency is difficult to capture from the usual mean-variance representation, as in Figure 1 (middle panel), so we will rely on the copula representation of the portfolios distribution. A copula is a bivariate probability distribution for which the marginal probability distribution of each variable is uniform. As we following Markowitz' framework,



■ **Figure 1** (left) Efficient frontier; (middle) Empirical portfolio distribution by portfolios' return and variance; (right) Efficient frontier in blue and contour of the empirical portfolio distribution in red. The market considered is made of the 19 sectoral indices of the DJSTOXX 600 Europe. The data is from October 16, 2017 to January 10, 2018.



■ **Figure 2** Copula representation of the portfolios distribution, by return and variance. The market considered is made of the 19 sectoral indices of the DJSTOXX 600 Europe. The data is from October 16, 2017 to January 10, 2018.

the variables considered are the portfolios' return and variance. Figure 2 illustrates such a copula and shows a positive dependency between portfolios returns and variances. Each line and column sum to 1% of the portfolios.

The methods introduced here can be used to study other dependencies such as the momentum effect [23] which is implied by the dependencies of asset returns with their past returns.

The dependencies mentioned here are important because

- through the return/volatility dependency, the detection of crisis raises policy makers awareness and allows them to act accordingly with potentially large implications in citizens' life (employment, wages, pensions, etc).
- the momentum, if persistent, questions the efficiency of financial markets, a strong assumption which still cannot be proven wrong.

Interestingly, the copulas can be computed over a single period of time making the information available as early as the sample allows. The copula for the momentum dependency can be computed over very short periods (even intra-daily). The copula for the return/volatility dependency requires the estimation of the stock returns variance-covariance matrix which

has to be estimated over a sufficiently large period of time to be reliable thus delaying the detection of crisis.³

In the general case, the framework to describe the dependencies is as follows. First, as the set of portfolios, we consider the canonical d -dimensional simplex $\Delta^d \subset \mathbb{R}^{d+1}$ where each point represents a portfolio and $d + 1$ is the number of assets. The vertices represent portfolios composed entirely of a single asset. The portfolio weights, i.e. fraction of investment to a specific asset, are non-negative and sum to 1. This is the most common investment set in practice today, as portfolio managers are typically forbidden from short-selling or leveraging. Second, considering some asset characteristic ac quantified by $C \in \mathbb{R}^{d+1}$, we define a corresponding quantity $f_{ac}(\omega, C)$ for any portfolio $\omega \in \Delta^d$. For instance, considering the vector of asset returns $R \in \mathbb{R}^{d+1}$, ω has the return $f_{ret}(\omega, R) = R^T \omega$. Then, we define the cross-sectional score of a given portfolio ω^* as

$$\rho_{ac} = \frac{\text{vol}(\Delta^*)}{\text{vol}(\Delta^d)}, \text{ where } \Delta^* = \{\omega \in \Delta^d : f(\omega, C) \leq f(\omega^*, C)\},$$

which corresponds to the share of portfolios with a return lower or equal to $R^* = R^T \omega^*$. This score corresponds to the cumulative distribution function of $f_{ac}(\omega, C)$ where the portfolios are uniformly distributed over the simplex. In the following, we consider the cases where f_{ac} is a linear combination or a quadratic form of C . Finally, the relationship between two asset characteristics ac_1 and ac_2 is presented in the form of a *copula* whose marginals are ρ_{ac_1} , ρ_{ac_2} . In our applications, the asset characteristics considered are the assets' returns and variances, and their values correspond to a linear combination of the returns and a quadratic form of the returns, respectively.

A copula is computed by slicing a simplex, i.e. the set of portfolios, along the asset characteristics. Thus, these questions are formulated in terms of convex bodies defined by intersecting simplices on one hand by a family of parallel hyperplanes and, on the other hand, by another family of parallel hyperplanes in the linear case or a family of concentric ellipsoids in the quadratic case. The latter case yields non-convex bodies between two ellipsoids.

1.2 Previous work

The cross-sectional score of portfolio returns has been introduced in [32] and it is estimated by means of a quasi-Monte Carlo method. The applications have been limited in terms of dimensions: the 30 DAX components and the 24 MSCI Netherlands components in [32], the 35 components of the IBEX in [33]. This score has also been proposed in [4], for the set of long/short equally weighted zero-dollar portfolios and whose estimation relying on combinatorics and statistics is computationally limited to around 20 dimensions, and in [3] where the focus was not on a precise score.

Given that volume computation of polytopes is #P-hard for both V- and H-representations [17] and no poly-time algorithm can achieve better than exponential error [18], the problem is not expected to admit of an efficient deterministic algorithm in general dimension. Developing algorithms for volume computation has received a lot of attention in the exact setting [7]. In the approximate setting, following the breakthrough polynomial-time algorithm by random walks [16], several algorithmic improvements ensued. The current best theoretical bounds are in [26] and for polytope sampling in [27]. Interestingly, only two pieces of software offer

³ Methods exist to estimate the stock returns variance-covariance matrix over short periods, see e.g. the range-based estimation method [5]. However they usually requires high-frequency data and are not widely used. These methods are beyond the scope of this paper.

practical algorithms in high dimension: `VolEsti`, a public-domain C++ implementation that scales to a few hundred dimensions [19], based on the Hit-and-Run paradigm [28], and the Matlab implementation of [10], which seems competitive to `VolEsti` in very high dimensions. Sampling from non-convex bodies appears in experimental works, with very few methods offering theoretical guarantees, e.g. in star shaped bodies [9] or, more recently, in [1].

1.3 Our contribution

We design and implement the following different approaches for volume computation: Efficient sampling from the simplex and using rejection to approximate the target volume, which is fast but inaccurate for small volumes. Exact formulae of integrals of appropriate probability distribution functions, which are implemented for the case of a single hyperplane. Optimizing the use of Lawrence’s sign decomposition method, since the polytopes at hand are shown to be simple with extra structure; a major issue here is numerical instability. Extending state-of-the-art random walks, based on the hit-and-run paradigm, to convex bodies defined as the intersection of linear halfspaces and ellipsoids. The latter is experimentally generalized to non-convex bodies defined by two ellipsoids with same quadratic form, and accurate approximations are obtained under certain mild conditions.

Our randomized algorithms for volume approximation extend `VolEsti`, where the main problem to address is to compute the maximum inscribed ball of the convex body P a.k.a. Chebychev ball. This reduces to a linear program when P is a polytope. For a convex body defined by intersecting a polytope with k balls, the question becomes a second-order cone program (SOCP) with k cones. When interchanging input balls with ellipsoids, the SOCP yields a sufficiently good approximation of the Chebychev ball.

Our implementations are in C++, lie in the public domain (`github`), are based on CGAL, rely on Eigen for linear algebra, on `Boost` for random number generators, and experiment with two SOCP solvers for initializing random walks. Our software tools are general and of independent interest. They are applied to allow us to extend the computation of a portfolio score to up to 100 dimensions, thus doubling the size of assets studied in financial research. We thus provide a new description of asset characteristics dependencies. Our methods allow us to propose and to effectively compute a new indicator of financial crises, which is shown to correctly identify all past crises with which we experimented. More importantly, it allows us to establish that periods of momentum nearly never overlap with the crisis events, which is a new result in finance.

The rest of the paper is organized as follows. The next section presents the convex bodies that arise from our financial modeling; we overview methods for representing and uniformly sampling from the simplex. Section 3 considers volumes defined as the intersection of a simplex and one hyperplane or more hyperplanes, the latter being organized in at most two families of parallel hyperplanes. Section 4 studies convex and non-convex bodies defined as the intersection of a simplex and an ellipsoid. Implementations are discussed in Section 5, along with experiments. We conclude with current work and open questions. Figures, proofs and tables of experiments that do not fit here are included in the full version of the paper in [8].

2 Convex bodies and Financial modeling

We analyze real data consisting of regular interval (e.g. daily) returns of assets such as the constituents of the Dow Jones Stoxx 600 Europe™ (DJ600). These are points in real space of dimension $d = 600$, respectively: $r_i = (r_{i,1}, \dots, r_{i,d}) \in \mathbb{R}^d$, $i \geq 1$.

We apply the methodology to a subset of assets drawn from the DJ 600 constituents⁴. Since not all stocks are tracked for the full period of time, we select the 100 assets with the longest history in the index⁵, and juxtapose:

- stock returns and stock returns covariance matrix over the same period to detect crises,
- stock returns and past stock returns to observe any momentum effect,

In financial applications, one considers compound returns over periods of k observations, where typically $k = 20$ or $k = 60$; the latter corresponds to roughly 3 months when observations are daily. Compound returns are obtained using k observations starting at the i -th one where the j -th coordinate corresponds to asset j and the component j of the new vector equals:

$$(1 + r_{i,j})(1 + r_{i+1,j}) \cdots (1 + r_{i+k-1,j}) - 1, \quad j = 1, \dots, d.$$

This defines the normal vector to a family of parallel hyperplanes, whose equations are fully defined by selecting appropriate constants. The second family of parallel hyperplanes is defined similarly by using an adjacent period of k observations.

The covariance matrix of the stock returns is computed using the shrinkage estimator of [25],⁶ as it provides a robust estimate even when the sample size is short with respect to the number of assets. A covariance matrix C defines a family of ellipsoids centered at the origin $0 \in \mathbb{R}^d$ whose equations $x^T C x = c$ are fully specified by selecting appropriate constants c .

To compute the copulas, we determine constants defining hyperplanes and ellipsoids so that the volume between two consecutive such objects is 1% of the simplex volume. The former are determined by bisection using the Varsi's exact formula. For ellipsoids $E(x) = c_i$, we look for the c_i 's by sampling the simplex, then evaluating $E(x)$ at each point. The values are sorted and the c_i selected so as to define intervals containing 1% of the values. Two consecutive ellipsoids intersecting the simplex and the family of parallel hyperplanes define a non-convex body for which we practically extend **VolEsti** algorithm.

The volume between two consecutive hyperplanes and two consecutive ellipsoids defines the density of portfolios whose returns and volatilities lie between the specified constants. We thus get a copula representing the distribution of the portfolios with respect to the portfolios returns and volatilities.

The main problem is to compute all the volumes that arise from the intersection of the two families with the unit simplex. We have to handle three types of full dimensional bodies and thus we develop or use existing methods for three different problems. The first is to compute the volume of the polytope defined by the intersection of the unit simplex with four hyperplanes which are pairwise parallel. The second arises when an ellipsoid intersects the unit simplex and a family of parallel hyperplanes. The third is to compute the volume of a non-convex body defined by the intersection of two concentric ellipsoids with a simplex and a family of parallel hyperplanes.

We develop and use four methods in total. The first (M1) is an exact formula for the volume defined by the intersection of simplex with a hyperplane. The second (M2 or s/r) is to sample the unit simplex and approximate all the volumes directly. The third method (M3) is the optimized Lawrence formula for simple polytopes and is used for the first problem.

⁴ The data used is from BloombergTM. It is daily and ranges from 01/01/1990 to 31/11/2017.

⁵ This implies a survivor bias, but we use it to assess the effectiveness of the methodology. One would wish to keep 600 constituents, replacing the exiting stocks with the entering ones along the sample.

⁶ Matlab code on <http://www.econ.uzh.ch/en/people/faculty/wolf/publications.html>.

The fourth method (M4) is the generalization of the VolEsti algorithm to non-linear and non-convex bodies.

2.1 Simplex representation and sampling

This subsection sets the notation, surveys methods for uniform sampling of the simplex, and discusses their efficient implementation.

The d -dimensional simplex $\Delta^d \subset \mathbb{R}^{d+1}$ may be represented by barycentric coordinates $\lambda = (\lambda_0, \dots, \lambda_d)$ s.t. $\sum_{i=0}^d \lambda_i = 1$, $\lambda_i \geq 0$. The points are $\sum_{i=0}^d \lambda_i v_i$, where $v_0, \dots, v_d \in \mathbb{R}^d$ are affinely independent. It is convenient to use a full-dimensional simplex, by switching to Cartesian coordinates $x = (x_1, \dots, x_d)$ using transformation $m_{bc} : \mathbb{R}^{d+1} \mapsto \mathbb{R}^d : \lambda \rightarrow x = M(\lambda_1, \dots, \lambda_n)^T + v_0$, where $M = [v_1 - v_0 \ \dots \ v_d - v_0]$, is a $d \times d$ invertible matrix. The inverse transform is:

$$m_{cb} : \mathbb{R}^d \mapsto \mathbb{R}^{d+1} : x \rightarrow \lambda = \begin{bmatrix} -1_d^T \\ I_d \end{bmatrix} M^{-1}(x - v_0) + \begin{bmatrix} 1 \\ 0_d \end{bmatrix}, \quad (1)$$

where $0_d, 1_d$ are d -dimensional column vectors of 1's and 0's, respectively, and I_d is the d -dimensional identity matrix.

A number of algorithms exist for sampling, where some have been rediscovered, while others contain errors; see the survey [37]. Let us start with a unit simplex in Cartesian coordinates. A $O(d \log d)$ algorithm is the following [12, 13, 34]: Generate d distinct integers uniformly in $\{1, \dots, K - 1\}$, where K is the largest representable integer. Sort them as follows: $x_0 = 0 < x_1 < \dots < x_{d+1} = K$. Now $(x_i - x_{i-1})/K$, $i = 1, \dots, d$, defines a uniform point. Assuming we possess a perfect hash-function, the choice of distinct integers takes $O(d)$. For $d > 60$ we implement a variant of Bloom filter to guarantee distinctness.

A linear-time algorithm is given in [35], which is generally the algorithm of choice, although it is slower for $d < 80$: (1) Generate $d + 1$ independent unit-exponential random variables y_i by uniformly sampling real value $x_i \in (0, 1)$ and setting $y_i = -\log x_i$, (2) Normalize the y_i 's by their sum $s = \sum_{i=0}^d y_i$, thus obtaining a uniformly distributed point $(y_0/s, \dots, y_d/s)$ on the d -dimensional canonical simplex lying in \mathbb{R}^{d+1} , (3) Project this point along the x_0 -axis to $(y_1/s, \dots, y_d/s)$, which is a uniform point in the full-dimensional unit simplex.

To sample an arbitrary simplex, we can map sampled points from the unit simplex by transformation (1), which preserves uniformity. Due to applying the transformation, the complexity is $O(d^2)$ to generate a uniform point. The same complexity, though slower in practice, is achieved in [22].

Sampling could be used in to approximate all the volumes that arise when two families of parallel hyperplanes intersect with a simplex. One can sample the simplex and count the percentage of points in the region of interest (we call this method M2 or s/r). The complexity is $O(kd)$ to generate k points. In the case of a family of ℓ parallel hyperplanes, all sample points are evaluated at the hyperplane linear polynomials in time $O(kd)$. Given the ℓ constant terms characterizing the hyperplanes, for each point we perform a binary search so as to decide in which layer it lies. Hence the total complexity is $O(k \log \ell)$, which is dominated since $\ell \leq 100$ typically. Given a family of ℓ ellipsoids with same quadratic form intersecting a simplex, the method requires $O(kd^2)$ to evaluate all sample points and $O(k \log \ell)$ to assign them to layers.

3 Intersection with hyperplanes

This section considers computing the volume of the intersection of a simplex and one or more linear halfspaces. The most general case is to be given two families of parallel hyperplanes and consider all created polytopes. We assume that the simplex is given in V-representation, i.e. as a set of vertices, and the hyperplanes by their equations.

We can always transform the simplex to be a unit full-dimensional simplex with the origin as one vertex by the transformation of Section 2.1. The same transform applies to the hyperplanes, and volume ratios as preserved.

Surprisingly, there exist an exact, iterative formula for the volume defined by intersecting a simplex with a hyperplane. A geometric proof is given in [38], by subdividing the polytope into pyramids and, recursively, to simplices. We implement a somewhat simpler formula [2], which also requires $O(d^2)$ operations. Let $H = \{(x_1, \dots, x_d) \mid \sum_{i=1}^d a_i x_i \leq z\}$ be the linear halfspace.

1. Compute $u_j = a_j - z$, $j = 1, \dots, d$. Label the nonnegative u_j as Y_1, \dots, Y_K and the negatives as X_1, \dots, X_J . Initialize $A_0 = 1, A_1 = A_2 = \dots = A_K = 0$.
2. For $h = 1, 2, \dots, J$ repeat: $A_k \leftarrow \frac{Y_k A_k - X_h A_{k-1}}{Y_k - X_h}$, for $k = 1, 2, \dots, K$.

If $\Delta^d \subset \mathbb{R}^d$ is the unit simplex then, for $h = J$, $A_K = \text{vol}(\Delta^d \cap H) / \text{vol}(\Delta^d)$.

We now consider simple polytopes defined by a constant number of families of parallel hyperplanes; In our application there are two such families. The defined polytopes are simple, i.e., all vertices are defined at the intersection of d hyperplanes, assuming that no hyperplane contains any of the simplex vertices and, moreover, two hyperplanes do not intersect on a simplex edge at the same point.

For a simple polytope P , the decomposition by Lawrence [24] picks $c \in \mathbb{R}^d$, $q \in \mathbb{R}$ such that $c^T x + q$ is not constant along any edge, i.e. $c, -c$ do not lie on the normal fan of any edge. For each vertex v , let $A(v)$ be the $d \times d$ matrix whose columns correspond to the equations of hyperplanes through v . Then $A(v)$ is invertible and vector $\gamma(v)$ such that $A(v)\gamma(v) = c$ is well defined up to a permutation. The assumption on c assures no entry vanishes, then

$$\text{vol}(P) = \frac{1}{d!} \sum_v \frac{(c^T v + q)^d}{|\det A(v)| \prod_{i=1}^d \gamma(v)_i}.$$

The computational complexity is $O(d^3 n)$, where n is the number of vertices. We set $q = 0$ for simplicity in the implementation. The main drawback of Lawrence's decomposition remains numerical instability when executed with floating point numbers, and high bit complexity, when executed over rational arithmetic. The latter is indispensable for $d > 30$ in our applications, because then numerical results become very unstable.

To compute the volume defined by the intersection of a simplex and two arbitrary hyperplanes, we exploit the fact that the simplex is unit in order to compute more effectively the determinants and the solutions of the linear system. The hardest case is when vertex v is defined by the two arbitrary hyperplanes H_a, H_b , the supporting hyperplane $H_0 : \sum_{i=1}^d x_i = 1$, and $d - 3$ hyperplanes of the form $H_i : x_i = 0$. Then we could compute $\gamma(v)_i$, $i = 1, \dots, d$ by solving the linear system in $O(d)$. The corresponding determinant is computed in $O(1)$. For the number of vertices we show the Lemma below.

► **Lemma 1.** *Polytopes in H-representation, defined by intersecting the simplex with two arbitrary hyperplanes in \mathbb{R}^d , have $O(d^2)$ vertices, which are computed in $O(1)$ each.*

The proof of lemma 1 is given in [8]. Lawrence's formula requires both H- and V-representation. In our setting, the H-representation is known, but the previous lemma allows us to obtain vertices as well.

► **Proposition 2.** *Let us consider polytopes defined by intersecting the simplex with two arbitrary hyperplanes. The total complexity of the Lawrence sign decomposition method, assuming that the H -representation is given, is $O(d^3)$.*

The entire discussion extends to polytopes defined by two families of parallel hyperplanes. The matrices $A(v)$ remain of the same form because each vertex is incident to at most one hyperplane from each family.

4 Intersection with ellipsoids

This section considers more general convex bodies, defined as a finite, bounded intersection of linear and nonlinear halfspaces. For this, we extend the polynomial-time approximation algorithm in `VolEsti` [19] so as to handle nonlinear constraints. Our primary motivation here is computing the volume of the intersection of a simplex with an ellipsoid in general dimension.

4.1 Random walks

The method in [19] follows the Hit-and-Run algorithm in [28], and is based on an approximation algorithm in $O^*(d^5)$. It scales in a few hundred dimensions by integrating certain algorithmic improvements to the original method. We have to generalize the method because the input is not a polytope but a general convex body, while `VolEsti` works for d -polytopes. It suffices to solve two subproblems: Compute the maximum inscribed ball of the convex body a.k.a. Chebychev ball, and compute the intersection points of a line that crosses the interior of the convex body P with the boundary of P .

The first problem is treated in the next subsection. For the second one, when the body is the intersection of linear and quadratic halfspaces, it suffices to solve systems of linear or quadratic equations. In our case where P has few input hyperplanes we can optimize that procedure by transforming a base of our polytope to an orthonormal base thus obtaining very simple linear systems. Formally, every ray ℓ in Coordinate Direction Hit-and-Run is of the form $p + \lambda e_k$ and parallel to $d - 1$ simplex facets. The roots of $\lambda^2 + 2\lambda p_k + |p|^2 - R^2$ define the intersection of a sphere with radius R , centered at the origin, and a coordinate direction ray ℓ . If C is the matrix of an ellipsoid centered at the origin its intersections with ℓ are roots of $C_{kk}x^2 + bx + c = 0$, where $b = 2C_{kk}p_k + 2\sum_{j=k+1}^d C_{kj}p_j + 2\sum_{i=0}^{k-1} C_{ik}p_i$ and $c = \sum_{i=0}^d C_{ii}p_i^2 + 2\sum_{j=i+1}^d C_{ij}p_i p_j$. Computing the roots, and keeping the largest negative and smallest positive λ is quite fast.

In our application, there are non-convex bodies defined by the intersection of two parallel hyperplanes, two concentric ellipsoids and a simplex. We thus modify `VolEsti` in order to compute the non convex volume. We make two major changes. First, in ray shooting, we have to check whether one quadratic equation has only complex solutions, which implies the ray does not intersect the ellipsoid. For λ , we take the largest negative and the smallest positive root in every step as well. Second, for the initial interior point, we sample from the unit simplex and when we find a point inside the intersection we stop and use it for initialization. We define an inscribed ball with this center and radius equal to some small $\epsilon > 0$. We stop the algorithm when we find the first inscribed ball as described in the next subsection. So we can set ϵ sufficiently small so it always defines an inscribed ball in practice, but the enclosing ball is enough to run the algorithm and do not stop until we find an inscribed ball.

The method works fine for $d < 35$ using the same walk length and number of points as for the convex case, and has time complexity and accuracy competitive to running `VolEsti` on the convex set defined by one ellipsoid. For $d > 35$, the method fails to approximate volume for most of the cases. This should be due to inaccurate rounding bodies and the inscribed ball we define. Given these first positive results, various improvements are planned.

4.2 Chebychev ball

This section offers methods for computing a ball inside the given convex region. Ideally, this is the largest inscribed ball, aka Chebychev ball, but a smaller ball may suffice. Computing the Chebychev ball reduces to a linear program when P is a polytope (p. 148 in [6]). For general convex regions, more general methods are proposed.

At the very least, one point must be obtained inside the convex region. When we do not have the Chebychev ball, an issue is that concentric balls with largest radii will again be entirely contained in the convex region, thus wasting time in the computation. In practice we use the one interior point as center of an enclosing ball, then reduce the radius until the first inscribed ball. To decide whether a given ball is inscribed, with high probability, we check whether all boundary points in Hit-and-Run belong to the sphere instead of any other constraint.

For a convex body that comes from intersecting a polytope with k balls the problem becomes a Second-Order Cone Program (SOCP) with k cones. However in our case we need to consider input ellipsoids. Assume that we transformed the ellipsoid to a ball $B' = \{x'_c + u' : \|u'\| \leq r'\}$, and applied the same transformation to the simplex to have $a_i x \leq b_i$ for $i \in [d + 1]$, $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$. The following SOCP computes the maximum ball $B = \{x_c + u : \|u\| \leq r\}$ in the intersection of the simplex and B' :

$$\max r, \quad \text{subject to : } a_i^T x_c + r \|a_i\| \leq b_i, \|x'_c - x_c\| \leq r' - r.$$

There are several ways to solve SOCP's such as to reformulate it to as a semidefinite program or perform a quadratic program relaxation. Moreover, since in our case we only have a single cone we could utilize special methods as in [20]. However, for our case it suffices to use the generic SOCP solver from [14] as it is very efficient; for a random simplex and ball, it takes 0.06 sec in $d = 100$ and < 20 sec in $d = 1000$, on Matlab using `ecos` and `yalmip` packages.

It is possible to apply the inverse transformation and get an inscribed ellipsoid, which is not necessarily largest possible. However we can use the maximum inscribed ball in that ellipsoid as an approximation of the Chebychev ball, by taking the center of that ellipsoid and the minimum eigenvalue of its matrix as the radius.

4.3 Portfolios' variances expressed by ellipsoids

In our financial application, portfolios are points in the unit d -dimensional simplex $\Delta^d \subset \mathbb{R}^{d+1}$ defined as the convex hull of $v_0, \dots, v_d \in \mathbb{R}^d$, where v_i lies on the i -th axis. The simplex lies in hyperplane $\sum_{i=0}^d \lambda_i = 1$. To model levels of portfolios' variances, a family of full-dimensional ellipsoids in \mathbb{R}^{d+1} , centered at the origin, is defined by the covariance matrix C of asset returns. We wish to compute the volume of intersections of this family with the simplex and, moreover, with a family of hyperplanes on the simplex. Rejection sampling would work in this context, however methods employing random walks require a full-dimensional convex body. Given a full $(d + 1)$ -dimensional ellipsoid $G : \lambda^T C \lambda - c \leq 0$ centered at the origin, where $C \in \mathbb{R}^{(d+1) \times (d+1)}$ is symmetric positive-definite, we compute the equation of

the ellipsoid defined $G \cap \Delta^d \subset \mathbb{R}^d$, by imposing the constraint $\sum_{i=0}^d \lambda_i = 1$ by transform m_{cb} in expression (1), thus obtaining:

$$(x - v_0)^T \left(M^{-T} [-1 I_d] C \begin{bmatrix} 1 \\ 0_d \end{bmatrix} M^{-1} \right) (x - v_0) + A(x - v_0) = c',$$

where the expression in parenthesis is the matrix defining the new d -dimensional ellipsoid in Cartesian coordinates, and $A \in \mathbb{R}^{d \times d}$, $c' \in \mathbb{R}$ are obtained by direct calculation. Similarly the simplex maps to Cartesian coordinates.

5 Implementation and experiments

5.1 Implementation

Our implementations are in C++, lie in the public domain⁷, and are using CGAL and Eigen. All experiments of the paper have been performed on a personal computer with Intel Pentium G4400 3.30GHz CPU and 16GB RAM. Times are averaged over 100 runs. All the details of the experiments and the corresponding Tables are given in the full version of the paper in [8].

We test the following convex bodies: a d -simplex intersected with: (1) two arbitrary halfspaces, (2) two parallel halfspaces, (3) an ellipsoid, (4) two parallel halfspaces and two cocentric ellipsoids (non convex body).

In general, M1 is preferred when available. Method M2 is the fastest and scales easily to 100 dimensions, so it is expected to be useful for larger dimensions. However, for small volumes its accuracy degrades; sampling more points makes it slower than M4. The latter is thus the method of choice for volumes $< 1\%$ of the simplex volume, but it is not clear whether it would be fast beyond $d = 100$. Method M3 is useful, even for small volumes, but it cannot scale to $d = 100$ due to numerical instability; if we opt for exact computing, it becomes too slow.

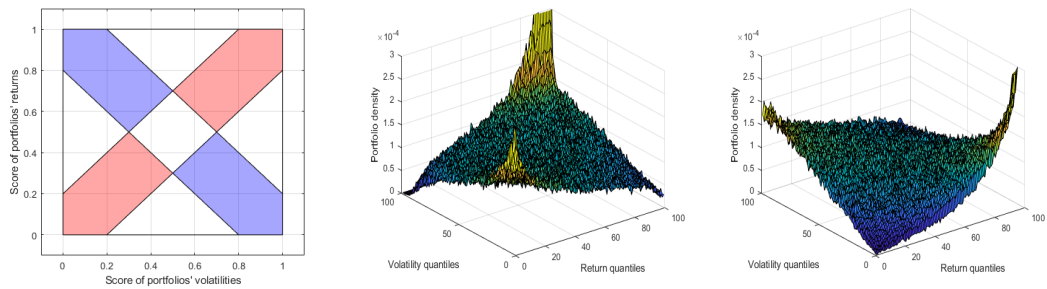
5.2 Financial modeling with real data

When working with real data and in order to build the indicator, we wish to compare the densities of portfolios along the two diagonals. In normal and up-market times, the portfolios with the lowest volatility present the lowest returns and the mass of portfolios should be on the up-diagonal. During crisis the portfolios with the lowest volatility present the highest returns and the mass of portfolios should be on the down-diagonal. Thus, defining up- and down-diagonal bands, we construct the indicator as the ratio of the mass on down-diagonal band over the one on the up-diagonal band, discarding the intersection of the two. Figure 3 illustrates the way the indicator is built.

In the following, the indicator is computed using copulas estimated using the sampling method, drawing 500,000 points. Computing the indicator over a rolling window of $k = 60$ days and with a band of $\pm 10\%$ with respect to the diagonal, we report in Table 1 all the periods over which the indicator is greater than 1 for more than 60 days. The periods should be more than 60 days to avoid the detection of isolated events whose persistence is only due to the auto-correlation implied by the rolling window.

We compare these results with the database for financial crises in European countries proposed in [15], and the longest periods found with the indicator coincide to crisis. The first

⁷ https://github.com/TolisChal/volume_approximation



(a) Diagonal bands considered to build the indicator. (b) Returns/variance relationship on the 1st September 1999. (c) Returns/variance relationship on the 1st September 2000.

■ **Figure 3** Illustration of the indicator. Panel (a): diagonal bands. Panels (b) and (c): copulas obtained during the dot-com bubble and at the beginning of the bubble burst, respectively. Blue=low density of portfolios, yellow=high density of portfolios.

crisis (from May 1990 to Dec. 1990) corresponds to the early 90's recession, the second one (from May 2000 to May 2001) to the dot-com bubble burst, the third one (from Oct. 2001 to Apr. 2002) to the stock market downturn of 2002, and the fourth one (from Dec. 2007 to Aug. 2008) to the sub-prime crisis. The remaining periods are shorter and correspond to periods of financial turmoil. They were not officially recognized as financial crisis.

Regarding the momentum effect, i.e. the effect of the compound returns of the last 60 days on the following 60-day compound returns. We observe that there were only 10 events of lasting momentum effect, mostly around the 1998-2004 period. We remark that they nearly never overlap with the crisis events, with the exception of the end of 2011. To the authors' knowledge, this last result is new in finance. Indeed, following the recommendations in [23], most of the literature focuses on the coincidence of momentum periods with economic crisis with inconclusive results, see e.g. [21] and [36]. By contrast, very few focus on its coincidence with financial crisis, probably because of the lack of financial crisis dating. Notably [11] find that momentum crash periods occur after the market has fallen, and when volatility is high and the market is recovering, i.e. at the end of a financial crisis. These two results are consistent.

6 Conclusion and future work

Since runtimes are very reasonable, we plan to extend our study to larger subsets of assets of DJ 600 and eventually the whole index in $d = 600$. Another extension is to consider polytopes defined by intersections of both families of parallel hyperplanes and the family of ellipsoids, thus creating 3-D diagrams of dependencies, which have never been studied in finance: one difficulty is to model the outcome since visualization becomes intricate.

An obvious enhancement is to parallelize our algorithms, which seems straightforward. Some other challenges are to obtain theoretical guarantees for the sampling-rejection methods and to extend the volume formula to the intersection with an ellipsoid. In [31], they propose a method to approximate the distribution f of quadratic forms in gamma random variables which is a similar problem to that in [30] (Section 3). It consists in fitting f with a generalized gamma distribution by matching its first 3 moments with those of f and to adjust the distribution with a polynomial in order to fit the higher moments. To get an approximation with a polynomial of degree k , the method requires the first $2k$ moments.

■ **Table 1** Phases of crisis (a) and momentum effect (b) detected with the indicator.

Start date	End date	Duration (days)	Start date	End date	Duration (days)
02-May-1990	20-Dec-1990	166	26-Dec-1990	16-Apr-1991	79
06-May-1992	14-Aug-1992	72	18-Oct-1993	11-Jan-1994	61
06-Oct-1994	27-Jan-1995	80	11-Aug-1998	24-Nov-1998	75
08-Apr-1996	24-Jul-1996	77	08-Nov-1999	04-Apr-2000	105
01-Jul-1997	13-Oct-1997	74	22-May-2001	04-Sep-2001	75
03-Mar-1999	01-Jun-1999	61	14-Jun-2002	09-Oct-2002	83
04-May-2000	09-May-2001	258	18-Oct-2002	27-Mar-2003	111
05-Oct-2001	05-Apr-2002	124	20-Aug-2004	21-Dec-2004	87
25-Feb-2004	28-May-2004	65	13-Oct-2006	19-Jan-2007	67
18-Nov-2005	11-Apr-2006	101	26-Jul-2011	21-Dec-2011	106
20-Dec-2007	04-Aug-2008	157	(b) All periods over which the momentum indicator is greater than one for more than 60 days.		
28-Dec-2010	12-Apr-2011	75			
18-Oct-2011	16-Jan-2012	63			
08-Oct-2013	04-Feb-2014	82			
04-Jun-2015	05-Oct-2015	87			
30-Nov-2015	03-Mar-2016	66			

(a) All periods over which the return/volatility indicator is greater than one for more than 60 days.

In the case of a quadratic form in d random variables, the moment of order m is obtained by a sum over all the partitions of m into d^2 terms. The number of partitions makes the computation of moments challenging even for $d \geq 5$.

References

- 1 Y. Abbasi-Yadkori, P.L. Bartlett, V. Gabillon, and A. Malek. Hit-and-run for sampling and planning in non-convex spaces. In *Proc. 20th Intern. Conf. Artificial Intelligence & Stat. (AISTATS)*, pages 888–895, 2017. URL: <http://proceedings.mlr.press/v54/abbasi-yadkori17a.html>.
- 2 M. Maswood Ali. Content of the frustum of a simplex. *Pacific J. Math.*, 48(2):313–322, 1973.
- 3 A. Banerjee and C-H. Hung. Informed momentum trading versus uninformed “naive” investors strategies. *J. Banking & Finance*, 35(11):3077–3089, 2011.
- 4 M. Billio, L. Calès, and D. Guéguan. A cross-sectional score for the relative performance of an allocation. *Intern. Review Appl. Financial Issues & Economics*, 3(4):700–710, 2011.
- 5 M. Billio, M. Getmansky, and L. Pelizzon. Dynamic risk exposures in hedge funds. *Comput. Stat. & Data Analysis*, 56(11):3517–3532, 2012. doi:10.1016/j.csda.2010.08.015.
- 6 S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, UK, 2004.
- 7 B. Büeler, A. Enge, and K. Fukuda. Exact volume computation for polytopes: A practical study. In G. Kalai and G.M. Ziegler, editors, *Polytopes: Combinatorics and Computation*, volume 29 of *Math. & Statistics*, pages 131–154. Birkhäuser, Basel, 2000.
- 8 L. Calès, A. Chalkis, I.Z. Emiris, and V. Fisikopoulos. Practical volume computation of structured convex bodies, and an application to modeling portfolio dependencies and

- financial crises. *CoRR*, arXiv:1803.05861, March 2018. URL: <https://arxiv.org/abs/1803.05861>.
- 9 K. Chandrasekaran, D. Dadush, and S. Vempala. Thin partitions: Isoperimetric inequalities and sampling algorithms for some nonconvex families. *CoRR*, abs/0904.0583, 2009. arXiv:0904.0583.
 - 10 B. Cousins and S. Vempala. A cubic algorithm for computing Gaussian volume. In *Proc. Symp. on Discrete Algorithms*, pages 1215–1228. SIAM/ACM, 2014.
 - 11 K. Daniel and T.J. Moskowitz. Momentum crashes. *J. Financial Economics*, 122(2):221–247, 2016.
 - 12 H.A. David. *Order statistics*. Wiley, New York, 2 edition, 1981.
 - 13 L. Devroye. *Non-uniform Random Variate Generation*. Springer, Berlin, 1986.
 - 14 A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *Proc. European Control Conference (ECC)*, pages 3071–3076, 2013.
 - 15 M. Lo Duca, A. Koban, M. Basten, E. Bengtsson, B. Klaus, P. Kusmierczyk, J.H. Lang, C. Detken, and T. Peltonen. A new database for financial crises in european countries. Technical Report 13, European Central Bank and European Systemic Risk Board, Frankfurt am Main, Germany, 2017.
 - 16 M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991. doi:10.1145/102782.102783.
 - 17 M.E. Dyer and A.M. Frieze. On the complexity of computing the volume of a polyhedron. *SIAM J. Comput.*, 17(5):967–974, 1988. doi:10.1137/0217060.
 - 18 G. Elekes. A geometric inequality and the complexity of computing volume. *Discrete & Computational Geometry*, 1:289–292, 1986.
 - 19 I.Z. Emiris and V. Fisikopoulos. Practical polytope volume approximation. *ACM Trans. Math. Soft.*, 2018. To appear. Prelim. version: Proc. Sympos. on Comput. Geometry, 2014.
 - 20 E. Erdougan and G. Iyengar. An active set method for single-cone second-order cone programs. *SIAM J. Optimization*, 17(2):459–484, 2006. doi:10.1137/040612592.
 - 21 J. Griffin, X. Ji, and J. Martin. Momentum investing and business cycle risk: Evidence from pole to pole. *J. Finance*, 58(6):2515–2547, 2003.
 - 22 C. Grimme. Picking a uniformly random point from an arbitrary simplex. Technical report, Information Systems and Statistics, Munster U., Germany, 2015.
 - 23 N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *J. Finance*, 48:65–91, 1993.
 - 24 J. Lawrence. Polytope volume computation. *Math. of Computation*, 57(195):259–271, 1991.
 - 25 O. Ledoit and M. Wolf. Honey, I shrunk the sample covariance matrix. *J. Portfolio Management*, 30(4):110–119, 2004. doi:10.3905/jpm.2004.110.
 - 26 Y.T. Lee and S.S. Vempala. Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. *CoRR*, abs/1710.06261, 2017. arXiv:1710.06261.
 - 27 Y.T. Lee and S.S. Vempala. Geodesic walks in polytopes. In *Proc. ACM Symp. on Theory of Computing*, pages 927–940. ACM, 2017. doi:10.1145/3055399.3055416.
 - 28 L. Lovász. Hit-and-run mixes fast. *Math. Programming*, 86:443–461, 1999. doi:10.1007/s101070050099.
 - 29 H. Markowitz. Portfolio selection. *J. Finance*, 7(1):77–91, 1952. doi:10.1111/j.1540-6261.1952.tb01525.x.
 - 30 A.M. Mathai. On linear combinations of independent exponential variables. *Communications in Statistics: Theory & Methods*, 2007.
 - 31 A.A. Mohsenipour and S.B. Provost. On approximating the distribution of quadratic forms in gamma random variables and exponential order statistics. *J. Statistical Theory & Appl.*, 12(2):173–184, 2013.

- 32 I. Pouchkarev. *Performance evaluation of constrained portfolios*. PhD thesis, Erasmus Research Institute of Management, The Netherlands, 2005.
- 33 I. Pouchkarev, J. Spronk, and J Trinidad. Dynamics of the spanish stock market through a broadband view of the IBEX 35 index. *Estudios Econom. Aplicada*, 22(1):7–21, 2004.
- 34 R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*. Wiley Interscience, New York, 2007.
- 35 R.Y. Rubinstein and B. Melamed. *Modern simulation and modeling*. Wiley, New York, 1998.
- 36 P. Scheurle and K. Spremann. Size, book-to-market, and momentum during the business cycle. *Review of Managerial Science*, 4(3):201–215, 2010.
- 37 N.A. Smith and R.W. Tromble. Sampling uniformly from the unit simplex. Technical report, Center for Language and Speech Processing, Johns Hopkins U., 2004.
- 38 G. Varsi. The multidimensional content of the frustum of the simplex. *Pacific J. Math.*, 46:303–314, 1973.