

Geometric tools for the valuation of performance-dependent options

Thomas Gerstner and Markus Holtz
Institut für Numerische Simulation, Universität Bonn, Germany

Abstract

In this paper, we describe several methods for the valuation of performance-dependent options. Thereby, we use a multidimensional Black-Scholes model for the temporal development of the asset prices. The martingale approach then yields the fair price as a multidimensional integral whose dimension is the number of stochastic processes in the model. The integrand is typically discontinuous, though, which makes accurate solutions difficult to achieve by numerical approaches. However, using tools from computational geometry we are able to derive a pricing formula which only involves the evaluation of smooth multivariate normal distributions. This way, performance-dependent options can efficiently be priced even for high-dimensional problems as it is shown by numerical results.

Keywords: option pricing, multivariate integration, hyperplane arrangements

1 Introduction

Performance-dependent options are financial derivatives whose payoff depends on the performance of one asset in comparison to a set of benchmark assets. Here, we assume that the performance of an asset is determined by the relative increase of the asset price over the considered period of time. The performance of the asset is then compared to the performances of a set of benchmark assets. For each possible outcome of this comparison, a different payoff of the derivative can be realized.

We use a multidimensional Black-Scholes model, see, e.g., Karatzas [1] for the temporal development of all asset prices required for the performance ranking. The martingale approach then yields a fair option price as a multidimensional integral whose dimension is the number of stochastic processes used in the model. In the so-called full model, the number of processes equals the number of assets. In the reduced model, the number of processes can be smaller. Unfortunately, in either

case there is no direct closed-form solution for these integrals. Moreover, the integrands are typically discontinuous which makes accurate numerical solutions difficult to achieve.

The main contribution of this paper is the derivation of closed-form solutions to these integration problems. For the reduced model, two novel tools from computational geometry are used. These tools are a fast enumeration method for the cells of a hyperplane arrangement and an algorithm for its orthant decomposition. The resulting closed-form solutions only involve the evaluation of smooth multivariate normal distributions which can be efficiently computed using numerical integration schemes which we illustrate in various numerical results.

2 Performance-dependent options

We assume that there are n assets involved in total. The considered asset gets assigned label 1 and the $n - 1$ benchmark assets are labeled from 2 to n . The price of the i -th asset varying with time t is denoted by $S_i(t)$, $1 \leq i \leq n$. All stock prices at the end of the time period T are collected in the vector $\mathbf{S} = (S_1(T), \dots, S_n(T))$.

2.1 Payoff profile

First, we need to define the payoff of a performance-dependent option at time T . To this end, we denote the relative price increase of stock i over the time interval $[0, T]$ by $\Delta S_i := S_i(T)/S_i(0)$. We save the performance of the first asset in comparison to a given strike price K (often $K = S_1(0)$) and in comparison to the benchmark assets at time T in a ranking vector $\mathbf{Rank}(\mathbf{S}) \in \{+, -\}^n$ defined by

$$\mathbf{Rank}_1(\mathbf{S}) = \begin{cases} + & \text{if } S_1 \geq K, \\ - & \text{else} \end{cases} \quad \text{and} \quad \mathbf{Rank}_i(\mathbf{S}) = \begin{cases} + & \text{if } \Delta S_1 \geq \Delta S_i, \\ - & \text{else} \end{cases}$$

for $i = 2, \dots, n$. In order to define the the payoff of the performance-dependent option we require bonus factors $a_{\mathbf{R}}$ which determine the bonus for each possible ranking $\mathbf{R} \in \{+, -\}^n$, see Section 5 for example profiles. In all cases we set $a_{\mathbf{R}} = 0$ if $\mathbf{R}_1 = -$ which corresponds to the option characteristic that a non-zero payoff only occurs if the stock price is above the strike.

The payoff of the performance-dependent option at time T is then defined by

$$V(S_1, T) = a_{\mathbf{Rank}(\mathbf{S})} (S_1(T) - K). \quad (1)$$

In the following, we aim to determine the fair price $V(S_1, 0)$ of such an option at the current time $t = 0$.

2.2 Multivariate Black-Scholes model

We assume that the stock prices are driven by $d \leq n$ stochastic processes modeled by the system of stochastic partial differential equations

$$dS_i(t) = S_i(t) \left(\mu_i dt + \sum_{j=1}^d \sigma_{ij} dW_j(t) \right) \quad (2)$$

for $i = 1, \dots, n$, where μ_i denotes the drift of the i -th stock, σ the $n \times d$ volatility matrix of the stock price movements and $W_j(t)$, $1 \leq j \leq d$, the corresponding Wiener processes. The matrix $\sigma \sigma^T$ is assumed to be positive definite. If $d = n$, we call the corresponding model full, if $d < n$, the model is called reduced.

By Itô's formula we get the explicit solution of (2) by

$$S_i(T) = S_i(\mathbf{X}) = S_i(0) \exp \left(\mu_i T - \bar{\sigma}_i + \sqrt{T} \sum_{j=1}^d \sigma_{ij} X_j \right) \quad (3)$$

for $i = 1, \dots, n$ with $\bar{\sigma}_i := \frac{1}{2}(\sigma_{i1}^2 + \dots + \sigma_{id}^2) T$ and $\mathbf{X} = (X_1, \dots, X_d)$ being a $N(\mathbf{0}, \mathbf{I})$ -normally distributed random vector.

3 Pricing formula in the full model

We now derive the price of a performance-dependent option as a multivariate integral in the case that the number of stochastic processes d equals the number of assets n .

3.1 Martingale approach

Using the usual Black-Scholes assumptions, the option price $V(S_1, 0)$ is given by the discounted expectation

$$V(S_1, 0) = e^{-rT} E[V(S_1, T)] \quad (4)$$

of the payoff under the unique equivalent martingale measure. To this end, the drift μ_i in (3) is replaced by the riskless interest rate r for each stock i . Plugging in the density function $\varphi(\mathbf{x}) := \varphi_{\mathbf{0}, \mathbf{I}}(\mathbf{x})$ of the random vector \mathbf{X} (note that $\mathbf{S} = \mathbf{S}(\mathbf{X})$), we get that the fair price of a performance-dependent option with payoff (1) is given by the d -dimensional integral

$$V(S_1, 0) = e^{-rT} \int_{R^d} \sum_{\mathbf{R} \in \{+, -\}^n} a_{\mathbf{R}}(S_1(T) - K) \chi_{\mathbf{R}}(\mathbf{S}) \varphi(\mathbf{x}) d\mathbf{x} \quad (5)$$

where the characteristic function $\chi_{\mathbf{R}}(\mathbf{S})$ is defined to be equal to one if $\mathbf{Rank}(\mathbf{S}) = \mathbf{R}$ and zero else.

3.2 Pricing formula

Now, we aim to derive an analytical expression for the computation of (5) in terms of smooth functions. To proof our main theorem we need the following two lemmas. For the first Lemma, we denote by $\varphi_{\mu, \mathbf{C}}(\mathbf{x})$ the Gauss kernel with mean μ and covariance matrix \mathbf{C} and by $\Phi(\mathbf{C}, \mathbf{b})$ the multivariate normal distribution corresponding to $\varphi_{0, \mathbf{C}}$ with limits $\mathbf{b} = (b_1, \dots, b_d)$.

Lemma 3.1 *Let $\mathbf{b}, \mathbf{q} \in \mathbb{R}^d$ and $\mathbf{A} \in \mathbb{R}^{d \times d}$ with full rank, then*

$$\int_{\mathbf{Ax} \geq \mathbf{b}} e^{\mathbf{q}^T \mathbf{x}} \varphi(\mathbf{x}) d\mathbf{x} = e^{\frac{1}{2} \mathbf{q}^T \mathbf{A} \mathbf{q}} \Phi(\mathbf{A} \mathbf{A}^T, \mathbf{A} \mathbf{q} - \mathbf{b}).$$

Proof: A simple computation shows that $e^{\mathbf{q}^T \mathbf{x}} \varphi(\mathbf{x}) = e^{\frac{1}{2} \mathbf{q}^T \mathbf{A} \mathbf{q}} \varphi_{\mathbf{q}, \mathbf{I}}(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$. Using the substitution $\mathbf{x} = \mathbf{A}^{-1} \mathbf{y} + \mathbf{q}$ we obtain

$$\int_{\mathbf{Ax} \geq \mathbf{b}} e^{\mathbf{q}^T \mathbf{x}} \varphi(\mathbf{x}) d\mathbf{x} = e^{\frac{1}{2} \mathbf{q}^T \mathbf{A} \mathbf{q}} \int_{\mathbf{y} \geq \mathbf{b} - \mathbf{A} \mathbf{q}} \varphi_{0, \mathbf{A} \mathbf{A}^T}(\mathbf{y}) d\mathbf{y}$$

and thus the assertion. \square

For the second Lemma, we define a comparison relation for two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ with respect to the ranking \mathbf{R} by $\mathbf{x} \geq_{\mathbf{R}} \mathbf{y} : \Leftrightarrow R_i(x_i - y_i) \geq 0$ for $1 \leq i \leq n$.

Lemma 3.2 *We have $\mathbf{Rank}(\mathbf{S}) = \mathbf{R}$ exactly if $\mathbf{A} \mathbf{X} \geq_{\mathbf{R}} \mathbf{b}$ with*

$$\mathbf{A} := \sqrt{T} \begin{pmatrix} \sigma_{11} & \dots & \sigma_{1d} \\ \sigma_{11} - \sigma_{21} & \dots & \sigma_{1d} - \sigma_{2d} \\ \vdots & & \vdots \\ \sigma_{11} - \sigma_{n1} & \dots & \sigma_{1d} - \sigma_{nd} \end{pmatrix}, \quad \mathbf{b} := \begin{pmatrix} \ln \frac{K}{S_1(0)} - rT + \bar{\sigma}_1 \\ \bar{\sigma}_1 - \bar{\sigma}_2 \\ \vdots \\ \bar{\sigma}_1 - \bar{\sigma}_n \end{pmatrix}.$$

Proof: Using (3) we see that $\mathbf{Rank}_1 = +$ is equivalent to

$$S_1(T) \geq K \iff \sqrt{T} \sum_{j=1}^d \sigma_{1j} X_j \geq \ln \frac{K}{S_1(0)} - rT + \bar{\sigma}_1$$

which yields the first row of the system $\mathbf{A} \mathbf{X} \geq_{\mathbf{R}} \mathbf{b}$. Moreover, for $i = 2, \dots, n$, the outperformance criterion $\mathbf{Rank}_i = +$ can be written as

$$S_1(T)/S_1(0) \geq S_i(T)/S_i(0) \iff \sqrt{T} \sum_{j=1}^d (\sigma_{1j} - \sigma_{ij}) X_j \geq \bar{\sigma}_1 - \bar{\sigma}_i$$

which yields rows 2 to n of the system. \square

Now we can state the following pricing formula which, in a slightly more special setting, is originally due to Korn [2].

Theorem 3.3 *The price of a performance-dependent option with payoff (1) is for the model (2) in the case $d = n$ given by*

$$V(S_1, T) = \sum_{\mathbf{R} \in \{+, -\}^n} a_{\mathbf{R}} (S_1(0) \Phi(\mathbf{A}_{\mathbf{R}} \mathbf{A}_{\mathbf{R}}^T, -\mathbf{d}_{\mathbf{R}}) - e^{-rT} K \Phi(\mathbf{A}_{\mathbf{R}} \mathbf{A}_{\mathbf{R}}^T, -\mathbf{b}_{\mathbf{R}}))$$

where $(\mathbf{b}_{\mathbf{R}})_i := \mathbf{R}_i \mathbf{b}_i$, $(\mathbf{d}_{\mathbf{R}})_i := \mathbf{R}_i \mathbf{d}_i$ and $(\mathbf{A}_{\mathbf{R}})_{ij} := \mathbf{R}_i \mathbf{A}_{ij}$ with \mathbf{A} and \mathbf{b} defined as in Lemma 3.2. Furthermore, $\mathbf{d} := \mathbf{b} - \sqrt{T} \mathbf{A} \sigma_1$ with σ_1^T being the first row of the volatility matrix σ .

Proof: The characteristic function $\chi_{\mathbf{R}}(\mathbf{S})$ in the integral (5) can be eliminated using Lemma 3.2 and we get

$$V(S_1, 0) = e^{-rT} \sum_{\mathbf{R} \in \{+, -\}^n} a_{\mathbf{R}} \int_{\mathbf{A}\mathbf{x} \geq \mathbf{R}\mathbf{b}} (S_1(T) - K) \varphi(\mathbf{x}) d\mathbf{x}. \quad (6)$$

By (3), the integral term can be written as

$$S_1(0) e^{rT - \bar{\sigma}_1} \int_{\mathbf{A}\mathbf{x} \geq \mathbf{R}\mathbf{b}} e^{\sqrt{T} \sigma_1^T \mathbf{x}} \varphi(\mathbf{x}) d\mathbf{x} - K \int_{\mathbf{A}\mathbf{x} \geq \mathbf{R}\mathbf{b}} \varphi(\mathbf{x}) d\mathbf{x}.$$

Application of Lemma 3.1 with $\mathbf{q} = \sqrt{T} \sigma_1$ shows that the first integral equals

$$e^{\frac{1}{2} \mathbf{q}^T \mathbf{q}} \int_{\mathbf{y} \geq \mathbf{R}\mathbf{b} - \mathbf{A}\mathbf{q}} \varphi_{0, \mathbf{A}\mathbf{A}^T}(\mathbf{y}) d\mathbf{y} = e^{\bar{\sigma}_1} \int_{\mathbf{y} \geq \mathbf{d}_{\mathbf{R}}} \varphi_{0, \mathbf{A}_{\mathbf{R}} \mathbf{A}_{\mathbf{R}}^T}(\mathbf{y}) d\mathbf{y} = e^{\bar{\sigma}_1} \Phi(\mathbf{A}_{\mathbf{R}} \mathbf{A}_{\mathbf{R}}^T, -\mathbf{d}_{\mathbf{R}}).$$

By a further application of Lemma 3.1 with $\mathbf{q} = \mathbf{0}$ we obtain that the second integral equals $K \Phi(\mathbf{A}_{\mathbf{R}} \mathbf{A}_{\mathbf{R}}^T, -\mathbf{b}_{\mathbf{R}})$ and thus the assertion holds. \square

4 Pricing formula in the reduced model

The pricing formula of Theorem 3.3 allows a stable and efficient valuation of performance-dependent options in the case of moderate-sized benchmarks. If the number n of benchmark assets is large, the high number 2^n of terms and the high dimension of the required normal distributions prevents an efficient application of the pricing formula, however. In this Section, we will derive a similar pricing formula for the reduced model which incorporates less processes than companies ($d < n$). This way, substantially fewer rankings have to be considered and much lower-dimensional integrals have to be computed.

4.1 Geometrical view

Lemma 3.2 and thus representation (6) remains also valid in the reduced model. Note, however, that \mathbf{A} is now an $(n \times d)$ -matrix which prevents the direct application of Lemma 3.1. At this point, a geometrical point of view is advantageous to illustrate the effect of performance comparisons in the reduced model.

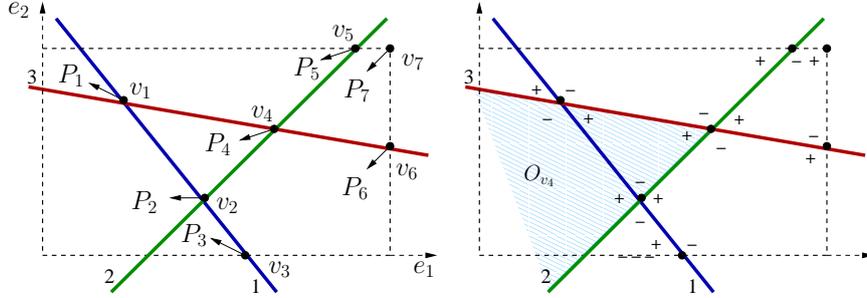


Figure 1: Illustration of the mapping between intersection points $\{v_1, \dots, v_7\}$ and polyhedral cells $P_j := P_{v_j}$ for a hyperplane arrangement $\mathcal{A}_{3,2}$ (left) and corresponding reflection signs $s_{v,w}$ as well as the orthant O_{v_4} (right).

The matrix \mathbf{A} and the vector \mathbf{b} define a set of n hyperplanes in the space \mathbb{R}^d . Its dissection into different cells is called a hyperplane arrangement and denoted by $\mathcal{A}_{n,d}$. Each cell in $\mathcal{A}_{n,d}$ is a (possibly open) polyhedron P which can uniquely be represented by a ranking vector $\mathbf{R} \in \{+, -\}^n$. Each element of the ranking vector indicates on which side of the corresponding hyperplane the polyhedral cell is located. Each polyhedron has the representation $P = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{A}\mathbf{x} \geq_{\mathbf{R}} \mathbf{b}\}$.

As the number of cells in the hyperplane arrangement $\mathcal{A}_{n,d}$ is much smaller than 2^n if $d < n$ (see Edelsbrunner [3]), we can significantly reduce the number of integrals which have to be computed by identifying all cells in the hyperplane arrangement. This way, (6) can be rewritten as

$$V(S_1, 0) = e^{-rT} \sum_{P \in \mathcal{A}} a_{\mathbf{R}} \int_P (S_1(T) - K) \varphi(\mathbf{x}) d\mathbf{x}. \quad (7)$$

4.2 Tools from computational geometry

Looking at (7), two problems remain: first, it is not easy to identify which ranking vectors appear in the hyperplane arrangement; second, the integration region is now a general polyhedron which requires involved integration rules. To resolve these difficulties, we need some more utilities from computational geometry.

First, we choose a set of linearly independent directions $\mathbf{e}_1, \dots, \mathbf{e}_d \in \mathbb{R}^d$ to impose an order on all points in \mathbb{R}^d . Thereby, we assume that no hyperplane is parallel to any of the directions. Moreover, we suppose that the hyperplane arrangement is non-degenerate which means that exactly d hyperplanes intersect in each vertex. Using the directions \mathbf{e}_i , an artificial bounding box which encompasses all vertices can be defined (see Figure 1, left). This bounding box is only needed for the localization of the polyhedral cells in the following Lemma and does not implicate any approximation.

Lemma 4.1 *Let the set \mathcal{V} consist of all interior vertices, of the largest intersection points of the hyperplanes with the bounding box and of the largest corner point of*

the bounding box. Furthermore, let $P_{\mathbf{v}} \in \mathcal{A}_{n,d}$ be the polyhedron which is adjacent to the vertex $\mathbf{v} \in \mathcal{V}$ and which contains no other vertex which is larger than \mathbf{v} with respect to the direction vectors. Then the mapping $\mathbf{v} \mapsto P_{\mathbf{v}}$ is one-to-one and onto.

Such a mapping is illustrated in Figure 1 (left). The proof of Lemma 4.1 can be found in our paper [4]. Using Lemma 4.1, an easy to implement optimal order algorithm can be constructed to enumerate all cells in a hyperplane arrangement.

Note that by Lemma 4.1 each vertex $\mathbf{v} \in \mathcal{V}$ corresponds to a unique cell $P_{\mathbf{v}} \in \mathcal{A}_{n,d}$ and thus to a ranking vector \mathbf{R} . We can, therefore, also assign bonus factors to vertices by setting $a_{\mathbf{v}} := a_{\mathbf{R}}$. Next, we assign to each vertex \mathbf{v} an associated orthant $O_{\mathbf{v}}$. An orthant is defined as an open region in \mathbb{R}^d which is bounded by $k \leq d$ hyperplanes. To find the orthant associated with the vertex \mathbf{v} , we look at k backward (with respect to the directions \mathbf{e}_i) points by moving \mathbf{v} backwards on each of the k intersecting hyperplanes. The unique orthant which contains \mathbf{v} and all backward points is denoted by $O_{\mathbf{v}}$. By definition, there exists a $(k \times d)$ -submatrix $\mathbf{A}_{\mathbf{v}}$ of \mathbf{A} and a k -subvector $\mathbf{b}_{\mathbf{v}}$ of \mathbf{b} such that the orthant $O_{\mathbf{v}}$ can be characterised as the set

$$O_{\mathbf{v}} = \left\{ \mathbf{x} \in \mathbb{R}^d : \mathbf{A}_{\mathbf{v}} \mathbf{x} \geq_{\mathbf{R}} \mathbf{b}_{\mathbf{v}} \right\}, \quad (8)$$

where \mathbf{R} is the ranking vector which corresponds to \mathbf{v} . Furthermore, given two vertices $\mathbf{v}, \mathbf{w} \in \mathcal{V}$, we define the reflection sign $s_{\mathbf{v},\mathbf{w}} := (-1)^{r_{\mathbf{v},\mathbf{w}}}$ where $r_{\mathbf{v},\mathbf{w}}$ is the number of reflections on hyperplanes needed to map $O_{\mathbf{w}}$ onto $P_{\mathbf{v}}$ (see Figure 1, right). Finally, let $\mathcal{V}_{\mathbf{v}}$ denote the set of all vertices of the polyhedron $P_{\mathbf{v}}$.

Lemma 4.2 *It is possible to algebraically decompose any cell of a hyperplane arrangement into a signed sum of orthant cells by*

$$\chi(P_{\mathbf{v}}) = \sum_{\mathbf{w} \in \mathcal{V}_{\mathbf{v}}} s_{\mathbf{v},\mathbf{w}} \chi(O_{\mathbf{w}}),$$

where χ is the characteristic function of a set. Moreover, all cells of a hyperplane arrangement can be decomposed into a signed sum of orthants using exactly one orthant per cell.

The first part of Lemma 4.2 is originally due to Lawrence [5], the second part can be found in [4].

4.3 Pricing formula

Now, we are finally able to give a pricing formula for performance-dependent options also for the reduced model.

Theorem 4.3 *The price of a performance-dependent option with payoff (1) is for the model (2) in the case $d \leq n$ given by*

$$V(S_1, 0) = \sum_{\mathbf{v} \in \mathcal{V}} c_{\mathbf{v}} (S_1(0) \Phi(\mathbf{A}_{\mathbf{v}} \mathbf{A}_{\mathbf{v}}^T, -\mathbf{d}_{\mathbf{v}}) - e^{-rT} K \Phi(\mathbf{A}_{\mathbf{v}} \mathbf{A}_{\mathbf{v}}^T, -\mathbf{b}_{\mathbf{v}}))$$

with $\mathbf{A}_\mathbf{v}$, $\mathbf{b}_\mathbf{v}$ as in (8) and with $\mathbf{d}_\mathbf{v}$ being the corresponding subvector of \mathbf{d} . The weights $c_\mathbf{v}$ are given by

$$c_\mathbf{v} := \sum_{\mathbf{w} \in \mathcal{V}: \mathbf{v} \in P_\mathbf{w}} s_{\mathbf{v},\mathbf{w}} a_\mathbf{w}.$$

Proof: By Lemma 4.1 we see that the integral representation (7) is equivalent to a summation over all vertices $\mathbf{v} \in \mathcal{V}$, i.e.

$$V(S_1, 0) = e^{-rT} \sum_{\mathbf{v} \in \mathcal{V}} a_\mathbf{v} \int_{P_\mathbf{v}} (S_1(T) - K) \varphi(\mathbf{x}) d\mathbf{x}.$$

By Lemma 4.2 we can decompose the polyhedron $P_\mathbf{v}$ into a signed sum of orthants and obtain

$$V(S_1, 0) = e^{-rT} \sum_{\mathbf{v} \in \mathcal{V}} a_\mathbf{v} \sum_{\mathbf{w} \in \mathcal{V}_\mathbf{v}} s_{\mathbf{v},\mathbf{w}} \int_{O_\mathbf{w}} (S_1(T) - K) \varphi(\mathbf{x}) d\mathbf{x}.$$

By the second part of Lemma 4.2 we know that only $c_{n,d}$ different integrals appear in the above sum. Rearranging the terms leads to

$$V(S_1, 0) = e^{-rT} \sum_{\mathbf{v} \in \mathcal{V}} c_\mathbf{v} \int_{O_\mathbf{v}} (S_1(T) - K) \varphi(\mathbf{x}) d\mathbf{x}.$$

Since now the integration domains $O_\mathbf{v}$ are orthants, Lemma 3.1 can be applied exactly as in the proof of Theorem 3.3 which finally implies the Theorem. \square

5 Numerical Results

In this Section, we present numerical examples to illustrate the use of the pricing formula from Theorem 4.3. In particular, we compare the efficiency of our algorithm to the standard pricing approach (denoted by STD) of quasi-Monte Carlo simulation of the expected payoff (4) based on Sobol point sets, see, e.g., Glasserman [6]. We systematically compare the numerical methods

- Quasi-Monte Carlo integration based on Sobol point sets (QMC),
- Product integration based on the Clenshaw Curtis rule (P), and
- Sparse Grid integration based on the Clenshaw Curtis rule (SG)

for the evaluation of the multivariate cumulative normal distributions (see Genz [7]). The Sparse Grid approach is based on [8]. All computations were performed on an Intel(R) Xeon(TM) CPU 3.06GHz processor. We consider a reduced Black-Scholes market with $n = 30$ assets and $d = 5$ processes. Thereby, we investigate two different choices for the bonus factors $a_\mathbf{R}$ in the payoff function (1):

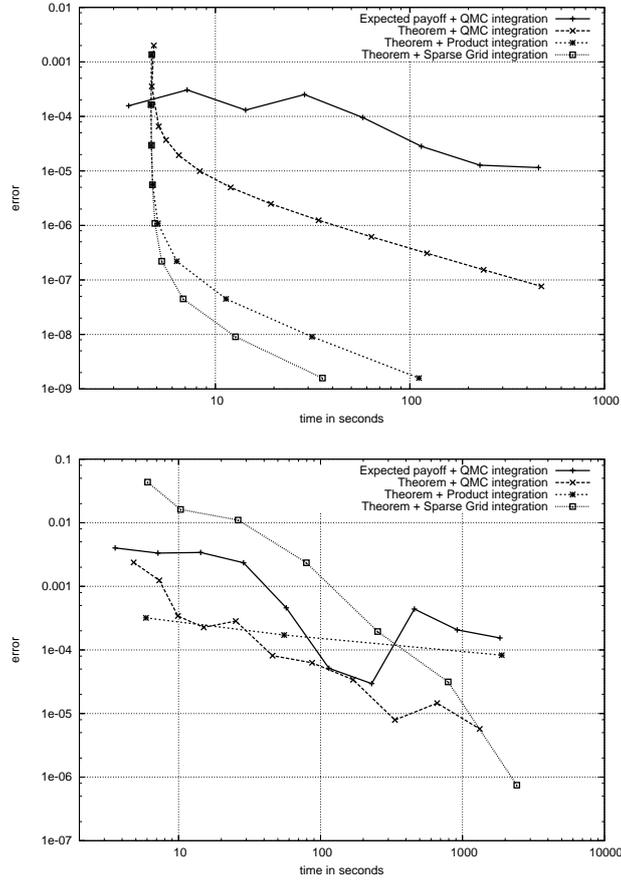


Figure 2: Errors and timings of the different numerical approaches to price the performance-dependent options of Examples 5.1 (top) and 5.2 (bottom).

Example 5.1 *Ranking-dependent option:*

$$a_{\mathbf{R}} = \begin{cases} m/(n-1) & \text{if } \mathbf{R}_1 = + \\ 0 & \text{else,} \end{cases}$$

where m denotes the number of outperformed benchmark assets. If the company ranks first there is a full payoff $(S_1(T) - K)^+$. If it ranks last the payoff is zero.

Example 5.2 *Outperformance option:*

$$a_{\mathbf{R}} = \begin{cases} 1 & \text{if } \mathbf{R} = (+, \dots, +) \\ 0 & \text{else.} \end{cases}$$

A payoff only occurs if $S_1(T) \geq K$ and if all benchmark assets are outperformed.

In both cases, we use the following model parameters: $K = 100$, $S_1(0) = 100$, $T = 1$, $r = 5\%$; σ is a 30×5 volatility matrix whose entries are uniformly distributed in $[-1/d, 1/d]$.

Depending on the specific choice of bonus factors, it turns out that often many weights $c_{\mathbf{v}}$ are zero in the formula of Theorem 4.3 which reduces the number of required normal distributions. Furthermore, all vertices \mathbf{v} located on the boundary of the bounding box correspond to orthants which are defined by $k < d$ intersecting hyperplanes. For these vertices, only a k -dimensional normal distribution has to be computed. In Example 5.1, we have 41 integrals with maximum dimension 2 while in Example 5.2, 31 integrals with maximum dimension 5 arise.

The convergence behaviour of the four different approaches (STD, QMC, P, SG) to price the options from the Examples 5.1 and 5.2 is shown in Figure 2. There, the time is displayed which is needed to obtain a given accuracy. One can see that the standard approach (STD) quickly achieves low accuracies. The convergence rate is slow and clearly lower than one, though. The integration scheme suffers under the irregularity of the integrand which is highly discontinuous and not of bounded variation. The QMC scheme clearly outperforms the STD approach in all examples. It exhibits a convergence rate of about one and leads to significantly smaller errors. As expected, the product integration approach (P) performs only really well in the Example 5.1 which is of low intrinsic dimension. The combination of Sparse Grid integration with our pricing formula (SG) leads to the best convergence rates. However, for higher dimensional problems as Example 5.2, this advantage is only visible if very accurate solutions are required. In the pre-asymptotic regime, the QMC scheme leads to smaller errors.

Acknowledgement: The authors wish to thank Ralf Korn, Kaiserslautern, for the introduction to this interesting problem and for his help with the derivation of the pricing formulas.

References

- [1] Karatzas, I., *Lectures on the Mathematics of Finance*, volume 8 of *CRM Monograph Series*. American Mathematical Society: Providence, R.I., 1997.
- [2] Korn, R., A valuation approach for tailored options. Working paper, 1996.
- [3] Edelsbrunner, H., *Algorithms in Combinatorial Geometry*. Springer, 1987.
- [4] Gerstner, T. & Holtz, M., The orthant decomposition of hyperplane arrangements, 2006. In preparation.
- [5] Lawrence, J., Polytope volume computation. *Math Comp*, **57(195)**, pp. 259–271, 1991.
- [6] Glasserman, P., *Monte Carlo Methods in Financial Engineering*. Springer, 2003.
- [7] Genz, A., Numerical computation of multivariate normal probabilities. *J Comput Graph Statist*, **1**, pp. 141–150, 1992.
- [8] Gerstner, T. & Griebel, M., Numerical integration using sparse grids. *Numerical Algorithms*, **18**, pp. 209–232, 1998.