AMERICAN OPTIONS BASED ON MALLIAVIN CALCULUS AND NONPARAMETRIC VARIANCE REDUCTION METHODS

Lokman Abbas-Turki, Bernard Lapeyre

HAL Id: hal-00589081

https://hal.archives-ouvertes.fr/hal-00589081v2

Submitted on 28 Apr 2011

AMERICAN OPTIONS BASED ON MALLIAVIN CALCULUS AND NONPARAMETRIC VARIANCE REDUCTION METHODS

L. A. ABBAS-TURKI[*] AND B. LAPEYRE[†]

Abstract. This paper is devoted to pricing American options using Monte Carlo and the Malliavin calculus. Unlike the majority of articles related to this topic, in this work we will not use localization fonctions to reduce the variance. Our method is based on expressing the conditional expectation $E[f(S_t)/S_s]$ using the Malliavin calculus without localization. Then the variance of the estimator of $E[f(S_t)/S_s]$ is reduced using closed formulas, techniques based on a conditioning and a judicious choice of the number of simulated paths. Finally, we perform the stopping times version of the dynamic programming algorithm to decrease the bias. On the one hand, we will develop the Malliavin calculus tools for exponential multi-dimensional diffusions that have deterministic and no constant coefficients. On the other hand, we will detail various nonparametric technics to reduce the variance. Moreover, we will test the numerical efficiency of our method on a heterogeneous CPU/GPU multi-core machine.

Key words. American Options, Malliavin Calculus, Monte Carlo, GPU.

AMS subject classifications. 60G40, 60H07

Introduction and objectives. To manage CPU (Central Processing Unit) power dissipation, the processor makers have oriented their architectures to multi-cores. This switch in technology led us to study the pricing algorithms based on **M**onte **C**arlo (**MC**) for multi-core architectures using CPUs and GPUs (Graphics Processing Units) in [1] and [2]. In the latter articles we basically studied the impact of using GPUs instead of CPUs for pricing European options using MC and American options using the **L**ongstaff and **S**chwartz (**LS**) algorithm [3]. The results of this study proves that we can greatly decrease the execution time and the energy consumed during the simulation.

In this paper, we explore another method to price **A**merican **O**ptions (**AO**) and which is based on **MC** using the **M**alliavin calculus (**MCM**). Unlike the LS method that uses a regression phase which is difficult to parallelize according to [2], the MCM is a square[1] Monte Carlo method which is more adapted to multi-cores than the LS method. Moreover, using MCM without localization does not depend on parametric regression, we can increase the dimensionality of the problem without any constraints except for adding more trajectories if we aim at more accurate results.

American contracts can be exercised at any trading date until maturity and their prices are given, at each time t, by [4]

$$(0.1) \qquad P_t(x) = \sup_{\theta \in \mathcal{T}_{t,T}} E_{t,x} \left(e^{-r(\theta - t)} \Phi(S_\theta) \right),$$

where $\mathcal{T}_{t,T}$ is the set of stopping times in the time interval $[t, T]$, $E_{t,x}$ is the expectation associated to the risk neutral probability knowing that $S_t = x$ and r and $\phi(S_t)$ are respectively the risk neutral interest rate and the payoff of the contract.

---

[*]Universit Paris-Est, Laboratoire d'Analyse et de Mathmatiques Appliques, Champs-sur-Marne, 77454 Marne-la-Valle Cedex2, France.

[†]Ecole des Ponts ParisTech, CERMICS Applied Probability Research Group, Champs-sur-Marne, 77455 Marne-la-Valle cedex 2, France.

[1]What we mean by square Monte Carlo is not necessarily simulating a square number of trajectories, but a Monte Carlo simulation that requires a Monte Carlo estimation, for each path, of an intermediate value (here the continuation) and this can be done by using the same set of trajectories as the first Monte Carlo simulation.

With Markovian models (which is the case in this article), to evaluate numerically the price (0.1), we first need to approach stopping times in $\mathcal{T}_{t,T}$ with stopping times taking values in the finite set $t = t_0 < t_1 < ... < t_n = T$. When we do this approximation, pricing American options can be reduced to the implementation of the dynamic programming algorithm [4]. Longstaff and Schwartz consider the stopping times formulation of the dynamic programming algorithm which allows them to reduce the bias by using the actual realized cash flow. We refer the reader to [5] for a formal presentation of the LS algorithm and details on the convergence. In (0.2), we rewrite the dynamic programming principle in terms of the optimal stopping times τ_k, for each path, as follows

$$(0.2) \qquad \begin{array}{c} \tau_n = T, \\ \forall k \in \{n-1,...,0\}, \quad \tau_k = t_k 1_{A_k} + \tau_{k+1} 1_{A_k^c}, \end{array}$$

where the set $A_k = \{\Phi(S_{t_k}) > C(S_{t_k})\}$ and $C(S_{t_k})$ is the continuation value whose expression is given by

$$(0.3) \qquad C(S_{t_k}) = E\left(e^{-r(t_{k+1}-t_k)} P_{t_{k+1}}(S_{t_{k+1}}) \Big| S_{t_k}\right).$$

Thus, to evaluate the price (0.1), we need to estimate $C(S_{t_k})$. Algorithms devoted to American pricing and based on Monte Carlo, differ essentially in the way they estimate and use the continuation value (0.3). For example the authors of [6] perform a regression to estimate the continuation value, but unlike [3], they use $C(S_{t_k})$ instead of the actual realized cash flow $P_{t_{k+1}}(S_{t_{k+1}})$ to update the price in (0.2). Other methods use the Malliavin Calculus [7] or the quantization method [8] for $C(S_{t_k})$ estimation. In [2], we implement the LS method because it is gaining widespread use in the financial industry. As far as this work is concerned, we are going to implement MCM but unlike [7] we use the induction (0.2) for the implementation and we reduce the variance differently, without using localization.

Formally speaking, if $r = 0$, we can rewrite the continuation using the Dirac measure $\varepsilon_x(\cdot)$ at the point x

$$(0.4) \quad C(x) = \frac{E\left(P_{t_{k+1}}(S_{t_{k+1}})\varepsilon_x(S_{t_k})\right)}{E\left(\varepsilon_x(S_{t_k})\right)} = \frac{E\left(P_{t_{k+1}}(S_{t_{k+1}})1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}}\right)}{E\left(1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}}\right)}.$$

The second equality is obtained using the Malliavin calculus and we will specify, in section 2 expression (2.3), the value of $\pi_{t_k,t_{k+1}}$ by an integration by part argument for the **M**ulti-dimensional **E**xponential **D**iffusions with deterministic **C**oefficients (**MEDC**) model

$$dS_t = S_t\sigma(t)dW_t, \quad S_0 = y,$$

in the case of deterministic non-constant triangular matrix $\sigma(t)$ and when $\sigma_{ij}(t) = \sigma_{ij}\delta(i-j)$ with a fixed constant σ_{ij} (The latter case will be used as a benchmark). Instead of simulating directly the last term in (0.4), in section 3 we project $1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}}$ using a conditioning as follows

$$(0.5) C(x) = \frac{E\left(P_{t_{k+1}}(S_{t_{k+1}})E\left[1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}}\big|\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}\right]\right)}{E\left(E\left[1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}}\big|\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}\right]\right)}.$$

Then, in section 4, we estimate (0.5) by Monte Carlo simulation and we use the approximation

$$(0.6) \qquad C(x) \approx \frac{\frac{1}{N'}\sum_{l=1}^{N'} P_{t_{k+1}}^l(S_{t_{k+1}}) h(x, \{\int_0^{t_{k+1}} \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l)}{\frac{1}{N}\sum_{l=1}^{N} h(x, \{\int_0^{t_{k+1}} \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l)},$$

$h(x, \{y_{ij}\}_{j\leq i}) = E(1_{S_{t_k}\geq x}(S_{t_k})\pi_{t_k,t_{k+1}} | \{\int_0^{t_{k+1}} \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d} = \{y_{ij}\}_{1\leq j\leq i\leq d})$ and $N \neq N'$. Thus, in section 4, we provide another method to accelerate the convergence based on a choice of the appropriate relation between N and N' that reduces the variance of the quotient (0.6). Note that, even if one can reduce the variance by an "appropriate" control variable, we choose here not to implement this kind of method because it is not standard for American options.

In the last section, on the one hand, we provide the numerical result comparison of LS and MCM. On the other hand, we study the results of using the two variance reduction methods (0.5) and (0.6). Finally, we test the parallel capabilities of MCM on a desktop computer that has the following specifications: Intel Core i7 Processor 920 with 9GB of tri-channel memory at frequency 1333MHz. It also contains one NVIDIA GeForce GTX 480.

Let us begin with section 1 in which we establish the notations, the Malliavin calculus tools and the model used.

1. Notations, hypothesis and key tools. Let T be the maturity of the American contract, $(\Omega, \mathcal{F}, P)$ a probability space on which we define an d-dimensional standard Brownian motion $W = (W^1, ..., W^d)$ and $\mathbb{F} = \{\mathcal{F}_s\}_{s\leq T}$ the P-completion of the filtration generated by W until maturity. Moreover, we denote by $\{\mathcal{F}_s^{i,...,d}\}_{s\leq t}$ the P-completion of the filtration generated by $(W^i, ..., W^d)$ until the fixed time $t \in [0, T]$. The process S_t models the price of a vector of assets $S_t^1, ..., S_t^d$ which constitute the solution of the following stochastic differential equation (' is the transpose operator)

$$(1.1) \qquad \frac{dS_t^i}{S_t^i} = (\sigma_i(t))'dW_t, \quad S_0^i = z_i, \quad i = 1, .., d,$$

where $\sigma(t) = \{\sigma_{ij}(t)\}_{1\leq i,j\leq d}$ is a deterministic triangular matrix ($\{\sigma_{ij}(t)\}_{i<j} = 0$). We suppose that the matrix $\sigma(t)$ is invertible, bounded and uniformly elliptic which insures the existence of the inverse matrix $\rho(t) = \sigma^{-1}(t)$ and its boundedness.

We choose the dynamic (1.1) because it is largely used for equity models, HJM interest rate models and variance swap models. Moreover, the case of $\sigma_{ij}(t) = \sigma_{ij}\delta(i-j)$ (σ_{ij} is a constant) will be easily tested in the section 5. One should notice that in the case where the dynamic of S is given by

$$\frac{dS_t^i}{S_t^i} = (\sigma_i)'(t, S_t)dW_t, \quad S_0^i = z_i, \quad i = 1, .., d,$$

we can use, for instance, the following Euler scheme to reduce this model to the model (1.1)

$$d\log(S_t^i) = \sum_{k=0}^{n-1} 1_{t\in[t_k,t_{k+1}[} \left[(\sigma_i)'(t_k, S_{t_k})dW_t - \frac{1}{2}[(\sigma_i)'\sigma_i](t_k, S_{t_k})dt \right],$$

$$S_0^i = z_i, \quad i = 1, .., d, \quad t_k = \frac{kT}{n}.$$

4

Note that this scheme does not discretize the process S but the process $\log(S)$.

Throughout this article, we will use two operators: The Malliavin derivatives D and the Skorohod integral δ and we define them in the same way as in [9]. For a fixed $m \in \mathbb{N}$, we define the subdivision $\{t_m^k\}_{k \leq 2^m}$ of the finite interval $[0, T]$ by: $t_m^k = kT/2^m$. Then we introduce $\mathcal{S}(\mathbb{R}^{2^m})$ the Schwartz space of infinitely differentiable and rapidly decreasing functions on $\mathbb{R}^{2^m}$. Let $f \in \mathcal{S}(\mathbb{R}^{2^m})$, we define the set $\mathfrak{S}^m$ of simple functionals by the following representation

$$F \in \mathfrak{S}^m \Leftrightarrow F = f\left(W_{t_m^1} - W_{t_m^0}, W_{t_m^2} - W_{t_m^1}, ..., W_{t_m^{2^m}} - W_{t_m^{2^m-1}}\right).$$

One can prove that $\mathfrak{S} = \bigcup_{m \in \mathbb{N}} \mathfrak{S}^m$ is a linear and dense subspace in $L^2(\Omega)$ and that the Malliavin derivatives $D^i F$ of $F \in \mathfrak{S}$ defined by

$$D_t^i F = \sum_{k=0}^{2^m-1} \frac{\partial f}{\partial x^{i,k}}\left(W_{t_m^1} - W_{t_m^0}, ..., W_{t_m^{2^m}} - W_{t_m^{2^m-1}}\right) \mathbf{1}_{[t_m^k, t_m^{k+1}[}(t)$$

represents a process of $L^2(\Omega \times [0, T])$ with values in $L^2([0, T])$. We associate to $\mathfrak{S}$ the norm $|| \cdot ||_{1,2}$ defined by

$$||F||_{1,2}^2 = E|F|^2 + \sum_{i=1}^{d} E \int_0^T (D_t^i F)^2 dt.$$

Finally, the space $\mathbb{D}^{1,2}$ is the closure of $\mathfrak{S}$ with respect to this norm and we say that $F \in \mathbb{D}^{1,2}$ if there exists a sequence $F_m \in \mathfrak{S}$ that converges to F in $L^2(\Omega)$ and that $D_u F_m$ is a Cauchy sequence in $L^2(\Omega \times [0, T])$.

Now we use the duality property between δ and D to define the Skorohod integral δ. We say that the process $U \in Dom(\delta)$ if $\forall F \in \mathbb{D}^{1,2}$

$$E\left(\int_0^T U_t \cdot D_t F dt\right) \leq C(U)||F||_{1,2},$$

where $C(U)$ is a positive constant that depends on the process U. If $U \in Dom(\delta)$, we define the Skorohod integral $\delta(U) = \int U_t \delta W_t$ by

$$(1.2) \quad \forall F \in \mathbb{D}^{1,2}, \quad E\left(F \int_0^T U_t \cdot \delta W_t\right) = E\left(F \delta(U)\right) = E\left(\int_0^T U_t \cdot D_t F dt\right),$$

$(\cdot)$ is the inner scalar product on $\mathbb{R}^d$.

Below, we give some standard properties of the operators D and δ:
1. If the process U_t is adapted, $\delta(U) = \int U_t \delta W_t$ coincides with the It integral $\int U_t dW_t$.
2. *The Chain Rule:* Let $F = (F_1, F_2, ..., F_k) \in (\mathbb{D}^{1,2})^k$ and $\phi : \mathbb{R}^k \to \mathbb{R}$ a continuously differentiable function with bounded partial derivatives. Then $\phi(F_1, F_2, ..., F_k) \in \mathbb{D}^{1,2}$ and:

$$D_t \phi(F_1, F_2, ..., F_k) = \sum_{i=1}^{k} \frac{\partial \phi}{\partial x^i}(F_1, F_2, ..., F_k) D_t F_i.$$

3. *The Integration by Parts:* The IP formula will be extensively used in the next section on the time intervals $I = (0, s)$ and $I = (s, t)$ with $s < t \in]0, T]$: Let $F \in \mathbb{D}^{1,2}$, an adapted process $U \in Dom(\delta)$ and that $FU \in Dom(\delta)$. For each $1 \leq i \leq d$ we have the following equality

$$(1.3) \qquad \int_I F U_u \delta^i W_u = F \int_I U_u dW_u^i - \int_I U_u D_u^i F du.$$

To simplify the notations, we denote $H_i(S_s^i) = H(S_s^i - x_i)$ for the heaviside function of the difference between the ith stock and the ith coordinate of the positive vector x.

Throughout this article, we will suppose that $g \in \mathcal{E}_b$ is a measurable function with polynomial growth

$$\mathcal{E}_b(\mathbb{R}^d) = \left\{ f \in \mathcal{M}(\mathbb{R}^d) \ : \ \exists C > 0 \ \text{ and } \ m \in \mathbb{N}; \ \ f(y) \leq C(1 + |y|_d)^m) \right\},$$

where $\mathcal{M}(\mathbb{R}^d)$ is the set of measurable functions on $\mathbb{R}^d$. The elements of the set $\mathcal{E}_b(\mathbb{R}^d)$ satisfy the finiteness of the expectations computed in this article.

2. The expression of the continuation value . The first theorem of this section provides the expression of the continuation (0.3) when using Malliavin calculus for MEDC models. This theorem can be considered as an extension of the log-normal multi-dimensional model detailed in [7]. In Theorem 2.4, we provide the expression of $\Gamma_{s,t}^k$, introduced in Theorem 2.1, without using Malliavin derivatives and this expression can be computed using the relation (2.15). The last theorem is a special case of the first one because we take $\sigma_{ij}(t) = \sigma_{ij}\delta(i - j)$ (σ_{ij} is a constant) that will be used to test numerically our nonparametric variance reduction methods detailed in section 3 and section 4.

THEOREM 2.1. *For any $s \in]0, t[$, $g \in \mathcal{E}_b$ and $x > 0$*

$$(2.1) \qquad E\left(g(S_t)\Big|S_s = x\right) = \frac{T_{s,t}[g](x)}{T_{s,t}[1](x)},$$

with

$$(2.2) \qquad T_{s,t}[f](x) = E\left(f(S_t)\Gamma_{s,t}\prod_{k=1}^d \frac{H_k(S_s^k)}{S_s^k}\right),$$

where $\Gamma_{s,t} = \Gamma_{s,t}^1$ and $\Gamma_{s,t}^1$ can be computed by the following induction scheme $\Gamma_{s,t}^d = \pi_{s,t}^{d,d}$, for $k \in \{1, ..., d-1\}$: $\Gamma_{s,t}^k = \Gamma_{s,t}^{k+1}\pi_{s,t}^{k,d} - \sum_{j=k+1}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du$ with

$$\pi_{s,t}^{k,d} = 1 + \sum_{j=k}^d \int_0^t \varphi_{jk}(u)dW_u^j, \quad \varphi_{jk}(u) = \frac{1}{s}\rho_{jk}(u)1_{u \in]0,s[} - \frac{1}{t-s}\rho_{jk}(u)1_{u \in]s,t[},$$

where ρ is the inverse matrix $\rho(u) = \sigma^{-1}(u)$.

From this theorem the value of $\pi_{t_k,t_{k+1}}$ in (0.4) is given by

$$(2.3) \qquad \pi_{t_k,t_{k+1}} = \Gamma_{t_k,t_{k+1}}\prod_{i=1}^d \frac{1}{S_{t_k}^i}.$$

To prove Theorem 2.1, we need the following two lemmas which are proved in the appendix. Lemma 2.2 expresses the independence of the sum $\sum_{i=k}^d \rho_{ik}(u)D_u^i g(S_t)$ from the variable u.

LEMMA 2.2. *For any $u \in]0, t[$ and $f \in \mathcal{C}^1(\mathbb{R}^d)$ then*

$$(2.4) \qquad \sum_{i=k}^{d} \rho_{ik}(u) D_u^i f(S_t) = S_t^k \partial_{x_k} f(S_t), \quad \rho(u) = \sigma^{-1}(u).$$

The following lemma constitutes with equality (2.4) the two keys of the proof of Theorem 2.1.

LEMMA 2.3. *For any $I \subset]0, t[$, $h \in \mathcal{C}_b^\infty(\mathbb{R})$, $x \in \mathbb{R}_+^d$ and $F \in \mathbb{D}^{1,2}$, we have*

$$(2.5) \qquad \begin{aligned} E\left(\int_I \frac{F D_u^k h(S_s^k)}{\sigma_{kk}(u)} du\right) &= E\left(h(S_s^k) F \sum_{i=k}^{d} \int_I \rho_{ik}(u) dW_u^i\right) \\ &\quad - E\left(h(S_s^k) \sum_{i=k}^{d} \int_I \rho_{ik}(u) D_u^i F du\right), \end{aligned}$$

where ρ is the inverse matrix $\rho(u) = \sigma^{-1}(u)$.

Proof of Theorem 2.1. To prove Theorem 2.1, it is sufficient to prove the following recursive relation on the parameter k for each $h_i \in \mathcal{C}_b^\infty(\mathbb{R})$ and $f \in \mathcal{C}^1(\mathbb{R}^d) \cap \mathcal{E}_b(\mathbb{R}^d)$

$$(2.6) \qquad E\left(f(S_t) \prod_{i=1}^{d} h_i'(S_s^i)\right) = E\left(f(S_t) \Gamma_{s,t}^{k+1} \prod_{i=1}^{k} h_i'(S_s^i) \prod_{i=k+1}^{d} \frac{h_i(S_s^i)}{S_s^i}\right).$$

Indeed, if it is the case then by density of $\mathfrak{S}$ in $L^2(\Omega)$, one can approximate $f(S_t) \in L^2(\Omega)$ by $F_m \in \mathfrak{S}$ and pass to the limit on the left and on the right term of (2.6) using Cauchy-Schwarz inequality and the dominated convergence theorem. Let us now consider the singularity due to the heaviside, let $\phi \in \mathcal{C}_c^\infty(\mathbb{R})$ be a mollifier function with support equal to $[-1, 1]$ and such that $\int_{\mathbb{R}} \phi(y) dy = 1$, then for any $y \in \mathbb{R}$ we define

$$h_{mk}(y) = (H_k * \phi_m)(y) \in \mathcal{C}_b^\infty(\mathbb{R}), \quad \phi_m(y) = m^{-1}\phi(m^{-1}y).$$

If the equality (2.6) is correct for any k, then

$$(2.7) \qquad E\left(f(S_t) \prod_{k=1}^{d} h_{mk}'(S_s^k)\right) = E\left(f(S_t) \Gamma_{s,t} \prod_{k=1}^{d} \frac{h_{mk}(S_s^k)}{S_s^k}\right).$$

On the one hand, $h_{mk}(y)$ converges to $H_k(y)$ except at $y = 0$ and the absolute continuity of the law of S_s^k ensures that $h_{mk}(S_s^k)$ converges almost surely to $H_k(S_s^k)$. Using the dominated convergence theorem, we prove the convergence of $h_{mk}(S_s^k)$ to $H_k(S_s^k)$ in $L^p(\Omega)$ for $p \geq 1$. By Cauchy-Schwarz inequality, we prove the convergence

$$E\left(f(S_t) \Gamma_{s,t} \prod_{k=1}^{d} \frac{h_{mk}(S_s^k)}{S_s^k}\right) \longrightarrow E\left(f(S_t) \Gamma_{s,t} \prod_{k=1}^{d} \frac{H_k(S_s^k)}{S_s^k}\right).$$

On the other hand, $h_{mk}'(y_k) = \int_{\mathbb{R}} H_k(z_k) \phi_m'(y_k - z_k) dz_k = \phi_m(y_k - x_k)$. Moreover, we observe that, according to our assumption, the distribution of the vector $(S_s^1, ..., S_s^d, S_t^1, ..., S_t^d)$ admits a density with respect to the Lebesgue mesure on $\mathbb{R}^d \times \mathbb{R}^d$ we denote it by $p(y, z)$ with $y = (y_1, ..., y_d)$ and $z = (z_1, ..., z_d)$, thus

$$E\left(f(S_t) \prod_{k=1}^{d} h_{mk}'(S_s^k)\right) = \int_{\mathbb{R}^d} f(z) \left(\int_{\mathbb{R}^d} \prod_{k=1}^{d} \phi_m(y_k - x_k) p(y, z) dy_1...dy_d\right) dz_1...dz_d$$

Because $\int_{\mathbb{R}^d} \prod_{k=1}^d \phi_m(y_k - x_k) p(y,z) dy_1...dy_d$ converges to $p(x,z)$, we have

$$E\left(f(S_t) \prod_{k=1}^d h'_{mk}(S_s^k)\right) \longrightarrow E\left(f(S_t) \prod_{k=1}^d \varepsilon_{x_k}(S_s^k)\right),$$

which concludes the first part of this proof.

To prove the induction (2.6), we introduce the following notations:

$$\widehat{h}_k^d(x) = \prod_{i=k}^d \frac{h_i(x_i)}{x_i}, \quad \widehat{h'}_k(x) = \prod_{i=1}^k h'_i(x_i), \quad x = (x_1, ..., x_d).$$

The case $k = d$ is given by

$$\begin{aligned}
E\left(f(S_t)\widehat{h'}_d(S_s)\right) &= E\left(\frac{1}{s} \int_0^s f(S_t)\widehat{h'}_{d-1}(S_s)\frac{D_u^d h_d(S_s^d)}{D_u^d S_s^d} du\right) \\
&= E\left(\frac{1}{s} \int_0^s f(S_t)\widehat{h'}_{d-1}(S_s)\frac{D_u^d h_d(S_s^d)}{\sigma_{dd}(u) S_s^d} du\right),
\end{aligned}$$

where we replaced $h'_d(S_s^d)$ by $\frac{D_u^d h_d(S_s^d)}{D_u^d S_s^d}$ in the first equality and $D_u^d S_s^d$ by its value $\sigma_{dd}(u) S_s^d$ in the second equality. Using Lemma 2.3 with

$$F = \frac{f(S_t)}{S_s^d} \prod_{i=1}^{d-1} h'_i(S_s^i) = \frac{f(S_t)}{S_s^d} \widehat{h'}_{d-1}(S_s)$$

and the fact that $\widehat{h'}_{d-1}(S_s)$ does not depend on the dth coordinate of the Brownian motion yields

$$\begin{aligned}
&E\left(\frac{1}{s} \int_0^s f(S_t)\widehat{h'}_{d-1}(S_s)\frac{D_u^d h_d(S_s^d) du}{\sigma_{dd}(u) S_s^d}\right) \\
(2.8) \quad &= E\left(F h_d(S_s^d)\frac{1}{s} \int_0^s \frac{dW_u^d}{\sigma_{dd}(u)}\right) - E\left(h_d(S_s^d)\frac{1}{s} \int_0^s D_u^d \frac{\widehat{h'}_{d-1}(S_s) f(S_t)}{S_s^d}\frac{du}{\sigma_{dd}(u)}\right) \\
&= E\left(F h_d(S_s^d)\frac{1}{s} \int_0^s \frac{dW_u^d}{\sigma_{dd}(u)}\right) - E\left(\widehat{h'}_{d-1}(S_s) h_d(S_s^d)\frac{1}{s} \int_0^s D_u^d \frac{f(S_t)}{S_s^d}\frac{du}{\sigma_{dd}(u)}\right).
\end{aligned}$$

Besides using Lemma 2.2 for the Malliavin derivative of $f(S_t)$, we get for $v \in\,]s,t[$

$$\frac{1}{\sigma_{dd}(u)} D_u^d\left[\frac{f(S_t)}{S_s^d}\right] = \frac{1}{S_s^d \sigma_{dd}(v)} D_v^d f(S_t) - \frac{f(S_t)}{S_s^d}.$$

Thus, the value of the last term of (2.8) is given by

$$\begin{aligned}
E\left(\widehat{h'}_{d-1}(S_s) h_d(S_s^d)\frac{1}{s} \int_0^s D_u^d \frac{f(S_t)}{S_s^d}\frac{du}{\sigma_{dd}(u)}\right) &= -E\left(\widehat{h'}_{d-1}(S_s) h_d(S_s^d)\frac{f(S_t)}{S_s^d}\right) \\
&+ E\left(\widehat{h'}_{d-1}(S_s)\frac{h_d(S_s^d)}{S_s^d}\frac{1}{t-s} \int_s^t D_v^d f(S_t)\frac{dv}{\sigma_{dd}(v)}\right).
\end{aligned}$$

And by duality (1.2) we remove the Malliavin derivative of $f(S_t)$ in the last term of the previous equality

$$\begin{aligned}
E\left(\frac{\widehat{h'}_{d-1}(S_s) h_d(S_s^d)}{S_s^d}\frac{1}{t-s} \int_s^t \frac{D_v^d f(S_t) dv}{\sigma_{dd}(v)}\right) &= E\left(\frac{\widehat{h'}_{d-1}(S_s) h_d(S_s^d)}{S_s^d} E\left\{\frac{1}{t-s} \int_s^t \frac{D_v^d f(S_t) dv}{\sigma_{dd}(v)}\Big| \mathcal{F}_s\right\}\right) \\
&= E\left(\frac{\widehat{h'}_{d-1}(S_s) h_d(S_s^d)}{S_s^d} E\left\{f(S_t)\frac{1}{t-s} \int_s^t \frac{dW_v^d}{\sigma_{dd}(v)}\Big| \mathcal{F}_s\right\}\right).
\end{aligned}$$

Regrouping all terms together

$$E\left(f(S_t)\widehat{h}'_d(S_s)\right) = E\left(f(S_t)\Gamma^d_{s,t}\widehat{h}'_{d-1}(S_s)\widehat{h}^d_d(S_s)\right), \quad \Gamma^d_{s,t} = \pi^{d,d}_{s,t}.$$

Let us suppose that (2.6) is satisfied for k and prove it for $k-1$, thus

$$
\begin{aligned}
E\left(f(S_t)\widehat{h}'_{d-1}(S_s)\right) &= E\left(f(S_t)\Gamma^{k+1}_{s,t}\widehat{h}^d_{k+1}(S_s)\widehat{h}'_k(S_s)\right)\\
&= E\left(\frac{1}{s}\int_0^s f(S_t)\Gamma^{k+1}_{s,t}\widehat{h}^d_{k+1}(S_s)\widehat{h}'_{k-1}(S_s)\frac{D^k_u h_k(S^k_s)}{\sigma_{kk}(u)S^k_s}du\right)\\
&= E\left(\frac{1}{s}\int_0^s \frac{f(S_t)\Gamma^{k+1}_{s,t}\widehat{h}^d_{k+1}(S_s)\widehat{h}'_{k-1}(S_s)}{S^k_s}\frac{D^k_u h_k(S^k_s)}{\sigma_{kk}(u)}du\right),
\end{aligned}
$$

where we replaced $h'_k(S^k_s)$ by $\frac{D^k_u h_k(S^k_s)}{D^k_u S^k_s}$ in the second equality. Using Lemma 2.3 with

$$F = \frac{f(S_t)\Gamma^{k+1}_{s,t}\widehat{h}^d_{k+1}(S_s)\widehat{h}'_{k-1}(S_s)}{S^k_s}$$

and the fact that $\widehat{h}'_{k-1}(S_s)$ does not depend on the jth coordinate $(j \geq k)$ of the Brownian motion yields

$$E\left(\frac{1}{s}\int_0^s \frac{FD^k_u h_k(S^k_s)}{\sigma_{kk}(u)}du\right) = \sum_{j=k}^d E\left(Fh_k(S^k_s)\frac{1}{s}\int_0^s \rho_{jk}(u)dW^j_u\right)$$

$$(2.9)$$

$$- \sum_{j=k}^d E\left(h_k(S^k_s)\widehat{h}'_{k-1}(S_s)\frac{1}{s}\int_0^s D^j_u\left[\frac{f(S_t)\widehat{h}^d_{k+1}(S_s)\Gamma^{k+1}_{s,t}}{S^k_s}\right]\rho_{jk}(u)du\right).$$

Besides, if for $x = (x_1,...,x_d)$ we denote $\Pi(x) = \frac{\widehat{h}^d_{k+1}(x)}{x_k}$, the Malliavin derivative of the last term of (2.9) provides

$$
\begin{aligned}
D^j_u\left[\Gamma^{k+1}_{s,t}\Pi(S_s)f(S_t)\right] = D^j_u\Gamma^{k+1}_{s,t}\Pi(S_s)f(S_t) &+ \Gamma^{k+1}_{s,t}D^j_u\Pi(S_s)f(S_t)\\
&+ \Gamma^{k+1}_{s,t}\Pi(S_s)D^j_u f(S_t).
\end{aligned}
$$

Using Lemma 2.2 for the Malliavin derivative in the two last terms, we get

$$(2.10) \qquad \sum_{j=k}^d \rho_{jk}(u)D^j_u\Pi(S_s) = S^k_s\partial_{x_k}\Pi(S_s) = -\Pi(S_s),$$

$$(2.11) \qquad \sum_{j=k}^d \rho_{jk}(u)D^j_u f(S_t) = S^k_t\partial_{x_k}f(S_t).$$

From (2.10), we deduce that

$$\widehat{h}'_{k-1}(S_s)h_k(S^k_s)f(S_t)\Gamma^{k+1}_{s,t}\frac{1}{s}\int_0^s \sum_{j=k}^d \rho_{jk}(u)D^j_u\Pi(S_s)du = -\frac{\widehat{h}'_{k-1}(S_s)\widehat{h}^d_k(S_s)f(S_t)\Gamma^{k+1}_{s,t}}{S^k_s}.$$

Thus, introducing the random variable $\widetilde{\pi}^{k,d}_{s,t} = 1+\sum_{j=k}^d \int_0^s \rho_{jk}(u)dW^j_u$ and using (2.9)

$$
\begin{aligned}
E\left(\frac{1}{s}\int_0^s \frac{FD^k_u h_k(S^k_s)}{\sigma_{kk}(u)}du\right) &= E\left(\mathcal{F}h_k(S^k_s)\widetilde{\pi}^{k,d}_{s,t}\right)\\
(2.12) \qquad &- E\left(\frac{\widehat{h}^d_k(S_s)\widehat{h}'_{k-1}(S_s)f(S_t)}{S^k_s}\frac{1}{s}\int_0^s \sum_{j=k}^d \rho_{jk}(u)D^j_u\Gamma^{k+1}_{s,t}du\right)\\
&- E\left(\frac{\widehat{h}^d_k(S_s)\widehat{h}'_{k-1}(S_s)\Gamma^{k+1}_{s,t}}{S^k_s}\frac{1}{t-s}\int_s^t \sum_{j=k}^d \rho_{jk}(u)D^j_u f(S_t)du\right),
\end{aligned}
$$

where we used the fact (2.11) that $\sum_{j=k}^{d} \rho_{jk}(u) D_u^j f(S_t)$ does not depend on u. Let us develop the last term of (2.12)

$$
\begin{aligned}
E & \left(\frac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s)\Gamma_{s,t}^{k+1}}{S_s^k} \frac{1}{t-s} \int_s^t \sum_{j=k}^d \rho_{jk}(u) D_u^j f(S_t) du \right) \\
& = E\left(\frac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s)}{S_s^k} \sum_{j=k}^d E\left[\frac{1}{t-s} \int_s^t \Gamma_{s,t}^{k+1} \rho_{jk}(u) D_u^j f(S_t) du \Big| \mathcal{F}_s \right] \right) \\
& = E\left(\frac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s)}{S_s^k} \sum_{j=k}^d E\left[f(S_t) \frac{1}{t-s} \int_s^t \Gamma_{s,t}^{k+1} \rho_{jk}(u) \delta W_u^j \Big| \mathcal{F}_s \right] \right) \\
& = \sum_{j=k}^d E\left(F h_k(S_s^k) \frac{1}{t-s} \int_s^t \rho_{jk}(u) dW_u^j \right) \\
& \quad - \sum_{j=k}^d E\left(\frac{f(S_t)\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s)}{S_s^k} \frac{1}{t-s} \int_s^t \rho_{jk}(u) D_u^j \Gamma_{s,t}^{k+1} du \right).
\end{aligned}
$$

We applied (1.2) in the third equality to remove the Malliavin derivative of $f(S_t)$. We also used (1.3) in the last equality. To complete the proof, we should remark that

$$
\frac{1}{s} \int_0^s D_u^j \Gamma_{s,t}^{k+1} \rho_{jk}(u) du - \frac{1}{t-s} \int_s^t D_v^j \Gamma_{s,t}^{k+1} \rho_{jk}(v) dv = - \int_0^t D_y^j \Gamma_{s,t}^{k+1} D_y^j \pi_{s,t}^{k,d} dy
$$

and because $\Gamma_{s,t}^{k+1}$ is an $\mathcal{F}_t^{k+1,...,d}$-measurable random variable

$$
\Gamma_{s,t}^k = \Gamma_{s,t}^{k+1} \pi_{s,t}^{k,d} - \sum_{j=k}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du = \Gamma_{s,t}^{k+1} \pi_{s,t}^{k,d} - \sum_{j=k+1}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du.
$$

$\square$

Theorem 2.4 provides the expression of $\Gamma_{s,t}^k$ in (2.16) without using the Malliavin derivatives $\{D_u^j\}_{j>k}$ and which can be efficiently computed using (2.15). We will use in Theorem 2.4 the set of the second order permutations $\overline{\mathcal{S}}_{k,d}$ defined as the following

$$(2.13) \qquad \overline{\mathcal{S}}_{k,d} = \{p \in \mathcal{S}_{k,d}, \ p \circ p = Id\},$$

where $\mathcal{S}_{k,d}$ is the set of permutations on $\{k,...,d\}$ and Id is the identity application. By induction, one can easily prove that

$$(2.14) \quad \overline{\mathcal{S}}_{k,d} = \{\tau_k^k \circ p, \ p \in \overline{\mathcal{S}}_{k+1,d}\} \cup \{\tau_k^l \circ p, \ p \in \overline{\mathcal{S}}_{k+1,d}, \ p(l) = l, \ l \in \{k+1,...,d\}\},$$

with $\tau_i^j : i \mapsto j$ as the transition application on $\{k,...,d\}$. We also denote by Δ the determinant that involves only the permutations of $\overline{\mathcal{S}}_{k,d}$, that is to say, the Δ associated to the matrix $C = \{C_{i,j}\}_{k \leq i,j \leq d}$ is given by

$$
\Delta = \sum_{p \in \overline{\mathcal{S}}_{k,d}} \epsilon(p) \prod_{i=1}^d C_{i,p(i)}
$$

Using (2.14), we can easily prove that

$$(2.15) \qquad \Delta = C_{k,k} \Delta_{k,k} + \sum_{i=k+1}^d \epsilon(\tau_k^i) C_{i,k} C_{k,i} \Delta_{k,i}$$

where $\Delta_{k,i}$ is the Δ associated to the $C^{i,k}$ obtained from C by suppressing the line and the column i as well as the line and the column k. Based on the development according to the first line, relation (2.15) provides a recursive formula even more efficient than

the determinant formula. Of course, we can generalize the relation (2.15) to the one that involves the development according to a jth line or a jth column with $k \leq j \leq d$.

THEOREM 2.4. *Based on the assumptions and the results of Theorem 2.1, for $k \in \{1, ..., d\}$ the value of $\Gamma_{s,t}^k$ is given by*

$$(2.16) \quad \Gamma_{s,t}^k = \sum_{p \in \overline{\mathcal{S}}_{k,d}} \epsilon(p) A_{k,p(k)} A_{k+1,p(k+1)} ... A_{d,p(d)} = \sum_{p \in \overline{\mathcal{S}}_{k,d}} \epsilon(p) \prod_{i=k}^d A_{i,p(i)},$$

with $\epsilon(p)$ as the signature of the permutation $p \in \overline{\mathcal{S}}_{k,d}$, $\overline{\mathcal{S}}_{k,d}$ defined in (2.13) and

$$A = \begin{pmatrix} \pi_{s,t}^{1,d} & C_{1,2} & C_{1,3} & \cdots & C_{1,d} \\ 1 & \pi_{s,t}^{2,d} & C_{2,3} & \cdots & C_{2,d} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & \cdots & 1 & \pi_{s,t}^{d-1,d} & C_{d-1,d} \\ 1 & 1 & \cdots & 1 & \pi_{s,t}^{d,d} \end{pmatrix},$$

where $C_{k,l}$ is the covariance of $\pi_{s,t}^{k,d}$ and $\pi_{s,t}^{l,d}$.

Proof. We prove (2.16) by a decreasing induction. For $k = d$, the expression (2.16) is clearly satisfied. We suppose that (2.16) is satisfied for $k + 1$ and we prove it for k. According to Theorem 2.1, $\Gamma_{s,t}^k = \Gamma_{s,t}^{k+1} \pi_{s,t}^{k,d} - \sum_{j=k+1}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du$, but

$$\begin{aligned} D_u^j \Gamma_{s,t}^{k+1} &= \sum_{l=k+1}^d \sum_{p \in \overline{\mathcal{S}}_{k+1,d}} \epsilon(p) \prod_{i=k+1, i \neq l}^d A_{i,p(i)} D_u^j A_{l,p(l)} \\ &= \sum_{l=k+1}^d \sum_{p \in \overline{\mathcal{S}}_{k+1,d}, p(l)=l} \epsilon(p) \prod_{i=k+1, i \neq l}^d A_{i,p(i)} D_u^j A_{l,l}, \end{aligned}$$

the second equality is due to the fact that $A_{l,p(l)}$ is a constant except for $p(l) = l$. Subsequently

$$\begin{aligned} &- \sum_{j=k+1}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du \\ &= - \sum_{l=k+1}^d \sum_{p \in \overline{\mathcal{S}}_{k+1,d}, p(l)=l} \epsilon(p) \prod_{i=k+1, i \neq l}^d A_{i,p(i)} \sum_{j=k+1}^d \int_0^t D_u^j A_{l,l} D_u^j \pi_{s,t}^{k,d} \\ &= - \sum_{l=k+1}^d \sum_{p \in \overline{\mathcal{S}}_{k+1,d}, p(l)=l} \epsilon(p) \prod_{i=k+1, i \neq l}^d A_{i,p(i)} C_{k,l}. \end{aligned}$$

Finally

$$\begin{aligned} \Gamma_{s,t}^k &= \Gamma_{s,t}^{k+1} \pi_{s,t}^{k,d} - \sum_{j=k+1}^d \int_0^t D_u^j \Gamma_{s,t}^{k+1} D_u^j \pi_{s,t}^{k,d} du \\ &= \pi_{s,t}^{k,d} \sum_{p \in \overline{\mathcal{S}}_{k+1,d}} \epsilon(p) \prod_{i=k+1}^d A_{i,p(i)} - \sum_{l=k+1}^d C_{k,l} \sum_{p \in \overline{\mathcal{S}}_{k+1,d}, p(l)=l} \epsilon(p) \prod_{i=k+1, i \neq l}^d A_{i,p(i)} \\ &= \sum_{p \in \overline{\mathcal{S}}_{k,d}} \epsilon(p) \prod_{i=k}^d A_{i,p(i)}. \end{aligned}$$

The last equality is due to the development of $\sum_{p \in \overline{\mathcal{S}}_{k,d}} \epsilon(p) \prod_{i=k}^d A_{i,p(i)}$ according to the kth line of A which can be justified by (2.14). $\square$

As a corollary of Theorem 2.1 and Theorem 2.4, we obtain the following result.

COROLLARY 2.5. *For any $s \in]0,t[$, $g \in \mathcal{E}_b$ and $x > 0$, if $\sigma_{ij}(t) = \sigma_{ij}\delta(i-j)$ then*

$$E\left(g(S_t)\Big|S_s = x\right) = \frac{T_{s,t}[g](x)}{T_{s,t}[1](x)},$$

with

$$(2.17) \qquad T_{s,t}[f](x) = E\left(f(S_t)\prod_{k=1}^{d}\frac{H_k(S_s^k)W_{s,t}^k}{\sigma_k s(t-s)S_s^k}\right),$$

and

$$W_{s,t}^k = (t-s)(W_s^k + \sigma_k s) - s(W_t^k - W_s^k), \quad k = 1, ..., d.$$

3. Variance reduction method based on conditioning .

In this section, we show that one can reduce the variance by a projection on $L^2\left(\left\{\int_0^t \sigma_{ij}(u)dW_u^j\right\}_{i,j}\right)$ and by using a closed formula of $T_{s,t}[1](x)$. Like in section 2, we give in Theorem 3.1 the results of the special case $\sigma_{ij}(t) = \sigma_{ij}\delta(i-j)$ (σ_{ij} is a constant) that will be used to test our variance reduction method.

We begin with $T_{s,t}[1](x)$, we can compute the explicit value of this function of x. The $T_{s,t}[1](x)$ closed formula can be got, for instance, from a change of probability. Indeed, we define the probability $\mathbb{P} = N_{coeff}(\prod_{k=1}^{d} S_0^k/S_s^k)P$ which yields

$$T_{s,t}[1](x) = \frac{1}{N_{coeff}}\mathbb{E}\left(\left[\prod_{k=1}^{d} H_k(S_s^k)\right]\Gamma_{s,t}\right),$$

N_{coeff} is a deterministic normalization coefficient such that $M_s = N_{coeff}(\prod_{k=1}^{d} S_0^k/S_s^k)$ is an exponential martingale with $E(M_s) = 1$. Under $\mathbb{P}$, $\Gamma_{s,t}$ has the same law as a polynomial of Gaussian variables which is sufficient to conduct the computations.

Let us now denote

$$h(x, \{y_{ij}\}_{j\leq i}) = E\left(\Gamma_{s,t}\prod_{k=1}^{d}\frac{H_k(S_s^k)}{S_s^k}\Big| \left\{\int_0^t \sigma_{ij}(u)dW_u^j\right\}_{1\leq j\leq i\leq d} = \{y_{ij}\}_{1\leq j\leq i\leq d}\right)$$

In what follows, we are going to prove that the function $h(x, \{y_{ij}\}_{1\leq j\leq i\leq d})$ can be explicitly known if, for each j, the $(d-k)\times(d-k)$ matrix $\Sigma_{jt} = \left\{\Sigma_{jt}^{ik}\right\}_{j\leq i,k\leq d} = \left\{\int_0^t \sigma_{ij}(u)\sigma_{kj}(u)du\right\}_{j\leq i,k\leq d}$ is invertible. First, please remark that according to our notations $i - j + 1$ and $k - j + 1$ are the indices of the element Σ_{jt}^{ik} in the matrix Σ_{jt} (we will use similar convention also for A^j, B^j, Ψ_{jt} and Φ_{jt}). Also we notice that the condition of invertibility of Σ_{jt} is not an important constraint, because one can choose a time discretization $\{t_m\}$ such that the matrices $\{\Sigma_{jt_m}\}_{k\leq d}$ fulfill this condition[2].

---

[2]Nevertheless, this is a difficult task when the dimension is sufficiently big.

The computation of $h(x, \{y_{ij}\}_{1\leq j\leq i\leq d})$ is based on a regression of Gaussian variables according to the Gaussian variables $Y_{ij} = \int_0^t \sigma_{ij}(u)dW_u^j$. First, we perform a linear regression of $\int_0^t \varphi_{jk}(u)dW_u^j$ according to Y_{ij}

$$(3.1) \qquad \int_0^t \varphi_{jk}(u)dW_u^j = \sum_{i=j}^d a_{i,k}^j Y_{ij} + X_{jk},$$

with $\{X_{jk}\}_{1\leq k\leq j\leq d}$ as a Gaussian vector $\mathcal{N}(0, C_X)$ which is orthogonal to Y. Using It isometry twice and the orthogonality of Y and X, we obtain

$$E\left(\int_0^t \varphi_{jk}(u)dW_u^j Y_{lj}\right) = \int_0^t \varphi_{jk}(u)\sigma_{lj}(u)du = \sum_{j=k}^n \Sigma_{jt}^{li} a_{i,k}^j.$$

If we denote $A^j = \{a_{i,k}^j\}_{j\leq i,k\leq d}$ and $\Psi_{jt} = \left\{\int_0^t \varphi_{jk}(u)\sigma_{lj}(u)du\right\}_{k,l}$, we get

$$A^j = \Sigma_{jt}^{-1}\Psi_{jt}.$$

In the same way, we perform a linear regression of $\int_0^s \sigma_{kj}(u)dW_u^j$ according to Y_{ij}

$$(3.2) \qquad \int_0^s \sigma_{kj}(u)dW_u^j = \sum_{i=j}^d b_{i,k}^j Y_{ij} + Z_{kj},$$

with $\{Z_{kj}\}_{1\leq j\leq k\leq d}$ as a Gaussian vector $\mathcal{N}(0, C_Z)$ which is orthogonal to Y. Using It isometry twice and the orthogonality of Y and Z, we obtain

$$E\left(\int_0^s \sigma_{kj}(u)dW_u^j Y_{lj}\right) = \int_0^s \sigma_{kj}(u)\sigma_{lj}(u)du = \sum_{i=j}^d \Sigma_{jt}^{li} b_{i,k}^j.$$

If we denote $B^j = \{b_{i,k}^j\}_{j\leq i,k\leq d}$, we get

$$B^j = \Sigma_{jt}^{-1}\Sigma_{js}.$$

Now using (3.1), (3.2) and the value of A and B, the covariance matrices C_X, C_Z and $C_{XZ} = E(XZ)$ are given by ($\Phi_{jt}^{i,k} = \int_0^t \varphi_{ji}(u)\varphi_{jk}(u)du$)

$$[C_X]_{i,k}^j = E(X_{ji}X_{jk}) = \Phi_{jt}^{i,k} - (A_k^j)'\Psi_{jt}^i - (A_i^j)'\Psi_{jt}^k + (A_k^j)'\Sigma_{jt}A_i^j,$$

$$[C_Z]_{i,k}^j = E(Z_{ij}Z_{kj}) = \Sigma_{js}^{i,k} - (B_k^j)'\Sigma_{js}^i - (B_i^j)'\Sigma_{js}^k + (B_k^j)'\Sigma_{jt}B_i^j,$$

$$[C_{XZ}]_{i,k}^j = E(X_{ji}Z_{kj}) = \Psi_{js}^{i,k} - (A_k^j)'\Sigma_{js}^i - (B_i^j)'\Psi_{jt}^k + (A_k^j)'\Sigma_{jt}B_i^j.$$

Using (3.1) and (3.2), we express $\Gamma_{s,t}$ and S_s^k according to Y_{ij}, Z_{ij} and X_{ji} then we conduct standard Gaussian computations to obtain the expression of $h(x, y_{ij})$ [3]. In Theorem 3.1, we give an explicit expression of $T_{s,t}[1](x)$ and $h(x, y_{ij})$ in the case of multi-dimensional B&S models with independent coordinates.

We can see that now that we know the explicit value of $T_{s,t}[1](x)$ and $h(x, \{y_{ij}\}_{1\leq j\leq i\leq d})$, subsequently, we should choose between the simulation of:

---

[3] One can use Mathematica to compute it formally.

P1) N paths of $g(S_t)h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{i,j}\right)$ then set the continuation to the value

$$C(x) := \frac{\frac{1}{N}\sum_{l=1}^N g^l(S_t)h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l\right)}{T_{s,t}[1](x)}.$$

P2) N' paths of $g(S_t)h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{i,j}\right)$ and N paths of $h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{i,j}\right)$ then set the continuation to the value

$$C(x) := \frac{\frac{1}{N'}\sum_{l=1}^{N'} g^l(S_t)h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l\right)}{\frac{1}{N}\sum_{l=1}^N h\left(x, \{\int_0^t \sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l\right)}.$$

Based on a variance reduction argument, Theorem 4.3 will indicate the preferable method to use.

THEOREM 3.1. *For any $s \in]0,t[$, $g \in \mathcal{E}_b$ and $x > 0$, if $\sigma_{ij}(t) = \sigma_{ij}\delta(i-j)$ then the conditional expectation given in Theorem 2.5 can be reduced to*

$$E\left(g(S_t)\Big|S_s = x\right) = \frac{E\left(g(S_t)\prod_{k=1}^d \sqrt{t}\exp\left(\frac{-s\sigma_k}{t}\left(\frac{s\sigma_k}{t} + W_t^k\right) - \frac{(d_{2k}(W_t^k)+m_k)^2}{2}\right)\right)}{\prod_{k=1}^d \sqrt{(t-s)}e^{-\frac{d_{1k}^2}{2}}},$$

with

$$m_k = \sigma_k\sqrt{\frac{s(t-s)}{t}}, \quad d_{2k}(W_t^k) = \sqrt{\frac{t}{s(t-s)}}\left(\beta_k - \frac{sW_t^k}{t}\right), \quad d_{1k} = \frac{\beta_k + \sigma_k s}{\sqrt{s}},$$

where

$$\beta_k = \frac{1}{\sigma_k}\left(\ln\left[\frac{x_k}{S_0^k}\right] + \frac{\sigma_k^2}{2}\right).$$

Proof. We simplify the constant $\sigma_k s(t-s)$ in (2.17) from the denominator and the numerator of the conditional expectation $E\left(g(S_t)\Big|S_s = x\right)$, then we use the independence of the coordinates to obtain

$$E\left(g(S_t)\Big|S_s = x\right) = \frac{E\left(g(S_t)\prod_{k=1}^d H_i(S_s^k)W_{s,t}^k/S_s^k\right)}{\prod_{k=1}^d E\left(H_k(S_s^k)W_{s,t}^k/S_s^k\right)}.$$

Afterwards, we use the independence of the increments to obtain

$$\begin{aligned}
E\left(\frac{H_k(S_s^k)}{S_s^k}W_{s,t}^k\right) &= E\left(\frac{H_k(S_s^k)}{S_s^k}[(t-s)(W_s^k + \sigma_k s) - s(W_t^k - W_s^k)]\right) \\
&= (t-s)E\left(\frac{H_k(S_s^k)}{S_s^k}(W_s^k + \sigma_k s)\right) - sE\left(\frac{H_k(S_s^k)}{S_s^k}\right)E(W_t^k - W_s^k) \\
&= (t-s)E\left(\frac{H_k(S_s^k)}{S_s^k}(\sqrt{s}G + \sigma_k s)\right),
\end{aligned}$$

where the random variable G has a standard normal distribution. Moreover we have the following equality in distribution

$$S_s^k \doteq S_0^k \exp\left(-\frac{\sigma_k^2}{2}s + \sigma_k\sqrt{s}G\right).$$

Computing the expectation we obtain

$$(3.3) \qquad E\left(\frac{H_k(S_s^k)}{S_s^k}(\sqrt{s}G + \sigma_k s)\right) = \alpha_k(t-s)\sqrt{\frac{s}{2\pi}}e^{-\frac{d_{1k}^2}{2}},$$

with $\alpha_k = e^{\sigma_k^2 s}$.

Regarding the numerator, we condition according to $W_t^k = w^k$ and we use the independence of coordinates

$$E\left(g(S_t)\prod_{k=1}^d H_k(S_s^k)W_{s,t}^k/S_s^k\right) = E\left(g(S_t)\prod_{k=1}^d h_k(W_t^k)\right),$$

with

$$(3.4) \qquad h_k(w^k) = E\left(H_k(S_s^k)W_{s,t}^k/S_s^k\Big|W_t^k = w^k\right).$$

Knowing $W_0^k = 0$ and $W_t^k = w^k$, when we fix s the random variable $W_s^k \doteq \frac{sw^k}{t} + \sqrt{\frac{s(t-s)}{t}}G$ and G has a standard normal distribution. Also, we have the following equality in distribution for $W_{s,t}^k$: $W_{s,t}^k \doteq (t-s)\sigma_k s + \sqrt{ts(t-s)}G$ and $S_s^k \doteq S_0^k \exp\left(-\frac{\sigma_k^2}{2}s + \sigma_k\frac{sw^k}{t} + \sigma_k\sqrt{\frac{s(t-s)}{t}}G\right)$. Then we compute (3.4) which yields:

$$(3.5) \quad h_k(w^k) \doteq \alpha_k\sqrt{\frac{ts(t-s)}{2\pi}}\exp\left(\frac{-s\sigma_k}{t}\left(\frac{s\sigma_k}{t} + w^k\right) - \frac{(d_{2k}(w^k) + m_k)^2}{2}\right),$$

with $\alpha_k = e^{\sigma_k^2 s}$.

Using (3.3) and (3.5) we obtain the requested result. $\square$

4. Advanced variance reduction method. We present, in this section, a less intuitive idea of variance reduction that is based on an appropriate relation between N and N' in (0.6). This method can be applied independently from conditioning detailed in previous section.

LEMMA 4.1. *Let $(X_k)_{k\in\mathbb{N}^*}$ be a sequence of independent $\mathbb{R}^n$-valued random variables that have the same law. We suppose that X_k is square integrable and we denote $\mu = E(X_k)$, $C_{i,j} = Cov(X_i, X_j)$. Let $r > 0$, $V_\mu = \{x \in \mathbb{R}^n, ||x - \mu||_{\mathbb{R}^n} < r\}$ and $g : \mathbb{R}^n \to \mathbb{R}$ such that $g \in \mathcal{C}^1(V_\mu)$, then we have the following limits when $N \to \infty$*

$$g(\overline{X}_N) \longrightarrow g(\mu) \ a.s., \quad \sqrt{N}(g(\overline{X}_N) - g(\mu)) \longrightarrow \mathcal{N}(0, \Sigma) \ in \ law,$$

such that

$$(4.1) \qquad \overline{X}_N = \frac{1}{N}\sum_{i=1}^N X_i, \quad \Sigma = \left(\frac{\partial g}{\partial x_1}, ..., \frac{\partial g}{\partial x_n}\right)_{x=\mu} C\left(\frac{\partial g}{\partial x_1}, ..., \frac{\partial g}{\partial x_n}\right)_{x=\mu}^t.$$

Proof. The almost sure convergence of $g(\overline{X}_N)$ results from the law of the large numbers and from the continuity of g in μ. For the same reasons, the gradient vector $\frac{\partial g}{\partial x}(\overline{X}_N)$ converges a.s. to $\frac{\partial g}{\partial x}(\mu)$. Besides $\sqrt{N}(g(\overline{X}_N) - g(\mu)) = \frac{\partial g}{\partial x}(\overline{X}_N) \cdot \sqrt{N}(\overline{X}_N - \mu) + \sqrt{N}(\overline{X}_N - \mu) \cdot \epsilon(\overline{X}_N - \mu)$ and using the Slutsky Theorem, with $G \sim \mathcal{N}(0, C)$

- $(\frac{\partial g}{\partial x}(\overline{X}_N), \sqrt{N}(\overline{X}_N - \mu))$ converges in law to $(\frac{\partial g}{\partial x}(\mu), G)$.

- $(\epsilon(\overline{X}_N - \mu), \sqrt{N}(\overline{X}_N - \mu))$ converges in law to $(0, G)$.

Finally, because $(x, y) \mapsto xy$ and $(x, y) \mapsto x + y$ are continuous, then $\sqrt{N}(g(\overline{X}_N) - g(\mu))$ converges in law to $\frac{\partial g}{\partial x}(\mu)G$. $\square$

Let us denote Q as the quotient given by

$$(4.2) \qquad Q = \frac{\frac{1}{N'}\sum_{i=1}^{N'} X_i}{\frac{1}{N}\sum_{i=1}^{N} Y_i}$$

If $|E(Y_i)| \geq \varepsilon > 0$, according to Lemma 4.1 Q converges to $E(X_i)/E(Y_i)$. In the following two theorems we will prove that we can accelerate the speed of convergence when acting on the relation between N and N'. We analyze the two cases:

case 1: $N' = \lambda_1 N$ with $\lambda_1 \in [1/N, 1]$ and we normalize (4.2)

$$Q = \frac{\frac{1}{N'}\sum_{i=1}^{N'} X_i}{\frac{1}{N}\left(\frac{N'}{N'}\sum_{i=1}^{N'} Y_i + \frac{N-N'}{N-N'}\sum_{i=1}^{N-N'} Y_i\right)} = \frac{A_{N'}}{\lambda_1 B_{N'} + (1-\lambda_1)B_{N,N'}},$$

where

$$A_{N'} = \frac{1}{N'}\sum_{i=1}^{N'} X_i, \quad B_{N'} = \frac{1}{N'}\sum_{i=1}^{N'} Y_i, \quad B_{N,N'} = \frac{1}{N-N'}\sum_{i=1}^{N-N'} Y_i.$$

We set $g_1(x, y, z) = x/(\lambda_1 y + (1-\lambda_1)z)$ and (4.1) provides

$$(4.3) \qquad \Sigma_1(\lambda_1) = \frac{1}{B^2}\left((2\lambda_1^2 - 2\lambda_1 + 1)\frac{A^2}{B^2}\sigma_2^2 + \sigma_1^2 - \frac{2\lambda_1 A}{B}\sigma_1\sigma_2\rho\right),$$

with $A = E(X)$, $B = E(Y)$, $\sigma_1^2 = Var(X)$, $\sigma_2^2 = Var(Y)$ and $\rho = Cov(X, Y)/(\sigma_1\sigma_2)$.

case 2: $N = \lambda_2 N'$ with $\lambda_2 \in [1/N', 1]$ and we normalize (4.2)

$$Q = \frac{\frac{1}{N'}\left(\frac{N}{N}\sum_{i=1}^{N} X_i + \frac{N'-N}{N'-N}\sum_{i=1}^{N'-N} X_i\right)}{\frac{1}{N}\sum_{i=1}^{N} Y_i} = \frac{\lambda_2 A_N + (1-\lambda_2)A_{N',N}}{B_N},$$

where

$$A_N = \frac{1}{N}\sum_{i=1}^{N} X_i, \quad A_{N',N} = \frac{1}{N'-N}\sum_{i=1}^{N'-N} X_i, \quad B_N = \frac{1}{N}\sum_{i=1}^{N} Y_i.$$

We set $g_2(x, y, z) = (\lambda_2 x + (1-\lambda_2)y)/z$ and (4.1) provides

$$(4.4) \qquad \Sigma_2(\lambda_2) = \frac{1}{B^2}\left((2\lambda_2^2 - 2\lambda_2 + 1)\sigma_1^2 + \frac{A^2}{B^2}\sigma_2^2 - \frac{2\lambda_2 A}{B}\sigma_1\sigma_2\rho\right),$$

with $A = E(X)$, $B = E(Y)$, $\sigma_1^2 = Var(X)$, $\sigma_2^2 = Var(Y)$ and $\rho = Cov(X, Y)/(\sigma_1\sigma_2)$.

THEOREM 4.2. *Based on what we defined above:*

1. *If $A^2\sigma_2^2 \geq B^2\sigma_1^2$ the minimum variance $\Sigma_{min} = \Sigma_1(\lambda_1^{min})$, with*

$$\lambda_1^{min} = \frac{1}{2} + \frac{B\sigma_1\rho}{2A\sigma_2}, \quad \Sigma_1 \text{ given in } (4.3).$$

2. *If $A^2\sigma_2^2 \leq B^2\sigma_1^2$ the minimum variance $\Sigma_{min} = \Sigma_2(\lambda_2^{min})$, with*

$$\lambda_2^{min} = \frac{1}{2} + \frac{A\sigma_2\rho}{2B\sigma_1}, \quad \Sigma_2 \text{ given in } (4.4).$$

Proof. We almost proved this theorem, indeed, one can easily verify that λ_1^{min} is the minimum of $\Sigma_1(\lambda_1)$ and λ_2^{min} is the minimum of $\Sigma_2(\lambda_2)$. To conclude we verify that $\Sigma_1(\lambda) \leq \Sigma_2(\lambda)$ if and only if $A^2\sigma_2^2 \geq B^2\sigma_1^2$. ☐

What is really appealing, in this theorem, is the fact that even if $\rho = 0$, one should use $N = (1/2)N'$ or $N' = (1/2)N$ depending on whether $A^2\sigma_2^2 \geq B^2\sigma_1^2$ or not. Nevertheless, in order to apply the results of either this theorem or Theorem 4.3, we should have a "sufficiently good" approximation of σ_1, σ_2, A, B and ρ. With our model $B = T_{s,t}[1](x)$ is explicitly known and we can have σ_2 in the same fashion as $T_{s,t}[1](x)$. In section 5, procedure **P2** is implemented by using the closed expression of B and σ_2 and simulating σ_1, A, ρ to get an approximation of λ_1^{min} or λ_2^{min} that we use to re-simulate Q. In the case where B and σ_2 are not known, we can implement one of the two methods that are also efficient:

M1) Using all the simulated paths N_{max}, we approximate the values of σ_1, σ_2, A, B and ρ then we compute λ_1^{min} or λ_2^{min} that we use to re-simulate Q.

M2) A fixed point alike method: Using all the simulated paths N_{max}, we approximate the values of σ_1, σ_2, A, B and ρ then fix a threshold ϵ and test the condition $A^2\sigma_2^2 \geq B^2\sigma_1^2$:

If $A^2\sigma_2^2 \geq B^2\sigma_1^2$: Use the previous approximations except A that will be simulated using $\lambda_1 N_{max}$ paths, such that λ_1^{min} is reached when

$$\left|\lambda_1 - \frac{1}{2} - \frac{B\sigma_1\rho}{2A\sigma_2}\right| < \epsilon.$$

If $A^2\sigma_2^2 \leq B^2\sigma_1^2$: Use the previous approximations except B that will be simulated using $\lambda_2 N_{max}$ paths, such that λ_2^{min} is reached when

$$\left|\lambda_2 - \frac{1}{2} - \frac{A\sigma_2\rho}{2B\sigma_1}\right| < \epsilon.$$

In the following theorem, we will answer on whether we should implement the simulation procedure **P1** or **P2**.

THEOREM 4.3. *Based on what was defined above and on the values of λ_1^{min} and λ_2^{min} given in Theorem 4.2, if*

1. $A^2\sigma_2^2 \geq B^2\sigma_1^2$ and $1 \geq \rho > \frac{A\sigma_2}{B\sigma_1}\left(\frac{\sqrt{13}-3}{2}\right)$ then $\left(B^2\Sigma_1(\lambda_1^{min}) - \sigma_1^2\right) < 0$.

2. $A^2\sigma_2^2 \leq B^2\sigma_1^2$ and $1 \geq \rho > \frac{B\sigma_1}{A\sigma_2}\left(\sqrt{\frac{5}{4} + \frac{2A^2\sigma_2^2}{B^2\sigma_1^2}} - \frac{3}{2}\right)$ then $\left(B^2\Sigma_2(\lambda_2^{min}) - \sigma_1^2\right) < 0$.

Proof.

1. If $A^2\sigma_2^2 \geq B^2\sigma_1^2$: $\Sigma_1(\lambda_1^{min}) = \Sigma_1(\rho)$ then we look for the condition on ρ that allows that the trinomial $\left(B^2\Sigma_1(\rho) - \sigma_1^2\right)$ is negative.

2. We go through the same argument as in 1.

☐

Theorem 4.3 tells us that, even though we can compute explicitly the expression of $T_{s,t}[1](x)$, according to the correlation, one can accelerate the convergence when using the quotient of two Monte Carlo estimators.

5. Simulation and numerical results. In this section we test our simulations on a geometric average payoff that has the following payoff

$$\text{(5.1)} \qquad \Phi^d_{geo}(S_T) = \left(K - \prod_{i=1}^{d} (S_{iT})^{1/d} \right)_+ .$$

In addition, we will test the American put on minimum and the American call on maximum that have the following payoffs

$$\text{(5.2)} \quad \Phi_{min}(S_T) = \left(K - \min(S_T^1, S_T^2) \right)_+, \quad \Phi_{max}(S_T) = \left(\max(S_T^1, S_T^2) - K \right)_+ .$$

The parameters of the simulations are the following: The strike $K = 100$, the maturity $T = 1$, the risk neutral interest rate $r = \ln(1.1)$, the time discretization is defined using the time steps that is given as a parameter in each simulation, $S_0^i = 100$ and $\sigma_{ij}(t) = \sigma_{ij}(t)\delta(j - i)$ with $\sigma_{ii} = 0.2$. The model considered is a multidimensional log-normal model

$$\frac{dS_t^i}{S_t^i} = rdt + (\sigma_i)'dW_t, \quad S_{i0} = y_i, \quad i = 1, .., d.$$

All the prices and the standard deviations are computed using a sample of 16 simulations. Besides, the true values, to which we compare our simulation results, are set using:

- the one-dimensional equivalence and a tree method [10], available in Premia [11], for options with $\Phi^n_{geo}(S_T)$ as payoff,
- the Premia implementation of a finite difference algorithm [12] in two dimensions for $\Phi_{min}(S_T)$ and $\Phi_{max}(S_T)$.

In Figure 5.1, we compare the **P2** ($N \neq N'$) version of MCM with a standard LS algorithm. The LS is implemented using linear regression for multidimensional contracts and using up to three degree monomials for the one-dimensional contract. The reason behind the choice of linear regression in the multidimensional case is the fact that the regression phase of LS can really increase the execution time without a significant amelioration of the prices tested.

In Figure 5.1, even if all the prices are sufficiently good, we see that MCM provides better prices than those of LS. Also when we increase the time steps, MCM is more stable than LS. However, for $n = 10$ and time steps > 10, we remark that one should simulate 2^{14} trajectories to stablize MCM. This fact is expected due to the important variance of the ten dimension contract and that one should simulate more trajectories, on the one hand, to have an asymptotically good approximation of the relation between N and N' and, on the other, to have a sufficient number of trajectories for the approximation of the continuation. The executions of MCM and LS with 2^{10} trajectories are carried out in less than one second. Moreover, using 2^{14} trajectories the LS and MCM are executed within seconds ($< 5s$). As a conclusion from this figure, MCM provides better results than LS in approximately the same execution time. When we increase the simulated trajectories to 2^{14}, the MCM prices are stabilized for high dimensions and are always better than LS prices.
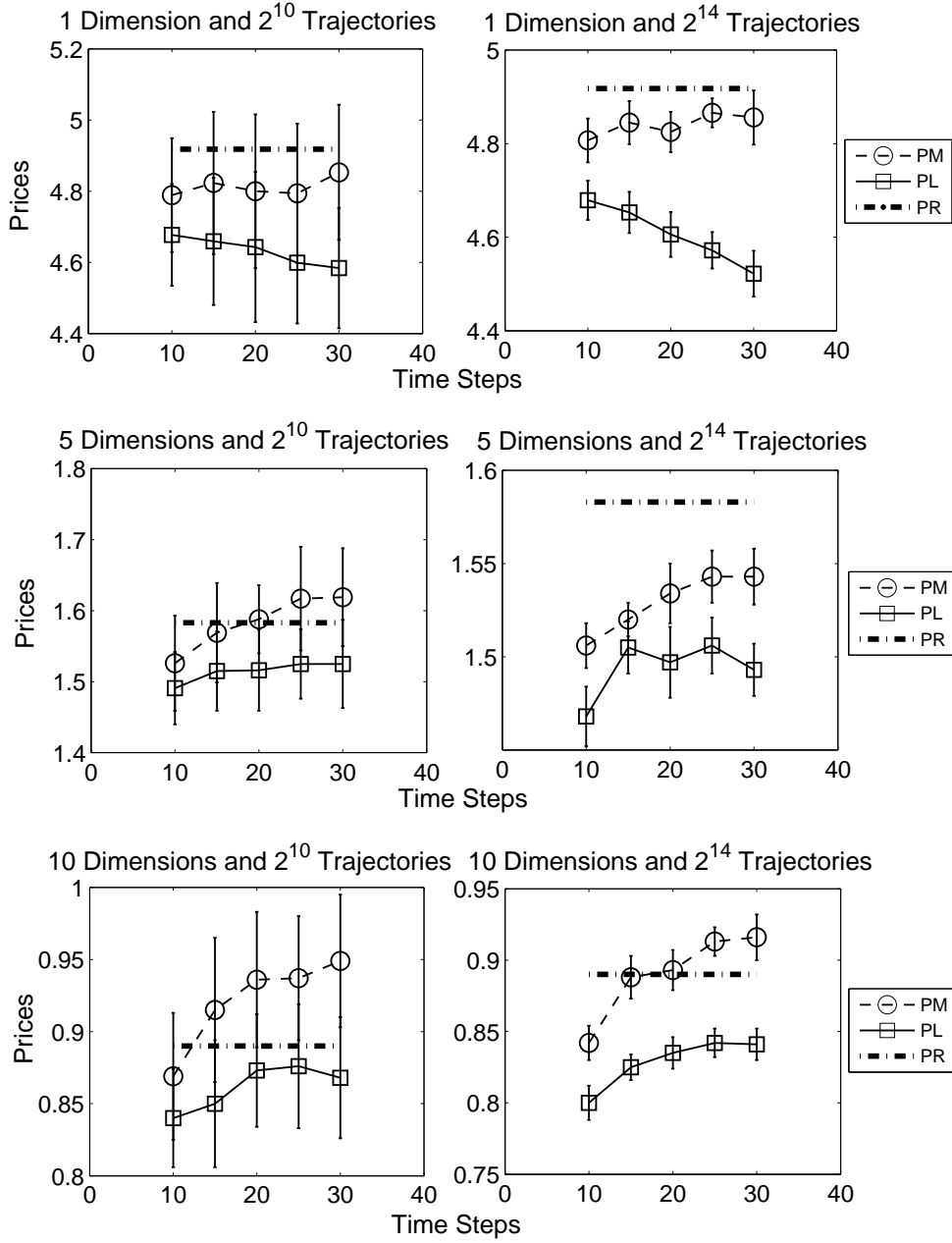
FIG. 5.1. *MCM Vs. LS for* $\Phi^d_{geo}(S_T)$*: PR is the real price. PM and PL are the prices obtained respectively by MCM and LS represented with their standard deviations.*

In Table 5.1, we remain with the same payoff $\Phi^d_{geo}(S_T)$ but this time we compare the different nonparametric methods of implementing MCM. In **P2(=)** and **P2(Opt)**, we use the same **P2** method but with $N = N'$ for the first one and $N \neq N'$ for the second (The relation between N and N' is detailed in pages 16 and 17). First, we remark that **P2(=)** is not stable in the multidimensional case and can give wrong results if the time steps > 10. However the **P2** method is stabilized when we im-

plement the version $N \neq N'$ of the advanced variance reduction method detailed in section 4. Also when we use 2^{10} trajectories, **P1** and **P2(Opt)** are almost similar. Nevertheless, with 2^{14} trajectories, **P2(Opt)** outperforms **P1** which indicates that we fill the conditions of Theorem 4.3 and we have an asymptotically good approximation of the relation between N and N'. As far as the execution time is concerned, the time consumed by **P2(Opt)** is not much different from **P1** when we use 2^{10} trajectories. In addition, using 2^{14} trajectories, the computations of the relation between N and N' can be performed on the CPU when the rest of the simulation is done on the GPU. The latter fact allows a similar overall execution time for **P2(Opt)** and **P1** (within seconds).

TABLE 5.1

P1 Vs. P2 for $\Phi^d_{geo}(S_T)$: The real values are equal to 4.918, 1.583 and 0.890 for dimensions one, five and ten respectively

Simulated Paths	Dim n	Time Steps	Price			Std Deviation		
			P1	**P2(=)**	**P2(Opt)**	**P1**	**P2(=)**	**P2(Opt)**
2^{10}	1	10	4.750	4.826	4.789	0.213	0.167	0.160
2^{10}	1	20	4.729	4.880	4.800	0.270	0.226	0.216
2^{10}	1	30	4.679	4.909	4.853	0.270	0.179	0.190
2^{10}	5	10	1.548	1.681	1.526	0.071	0.073	0.067
2^{10}	5	20	1.632	> 2.0	1.588	0.070		0.048
2^{10}	5	30	1.650	> 2.3	1.619	0.074		0.069
2^{10}	10	10	0.900	1.112	0.869	0.039	0.045	0.044
2^{10}	10	20	0.921	> 1.3	0.936	0.043		0.047
2^{10}	10	30	0.908	> 1.5	0.949	0.035		0.046
2^{14}	1	10	4.738	4.812	4.807	0.057	0.046	0.047
2^{14}	1	20	4.675	4.869	4.825	0.047	0.044	0.043
2^{14}	1	30	4.638	4.876	4.856	0.072	0.059	0.058
2^{14}	5	10	1.487	1.526	1.506	0.057	0.012	0.012
2^{14}	5	20	1.504	1.639	1.534	0.047	0.021	0.016
2^{14}	5	30	1.508	> 1.8	1.543	0.072		0.015
2^{14}	10	10	0.845	0.938	0.842	0.013	0.015	0.012
2^{14}	10	20	0.901	> 1.2	0.893	0.012		0.014
2^{14}	10	30	0.923	> 1.3	0.916	0.015		0.016

Because of the bad results obtained previously with **P2(=)**, we eliminate this method and we only consider **P2(Opt)** and **P1**. In Table 5.2, we analyze the American put on minimum and the American call on maximum in two dimensions. As far as Φ_{min} is concerned, **P2(Opt)** outperforms **P1** even when we use only 2^{10}.Regarding Φ_{max}, **P1** performs better than **P2(Opt)** for 2^{10} trajectories which indicates that, because of the big variance produced by Φ_{max} relatively to Φ_{min}, the relation between N and N' is not well estimated. Simulating 2^{14} trajectories, we obtain similar results for **P1** and **P2(Opt)** for Φ_{max}.

Let us now study the parallel adaptability of MCM for parallel architectures. In Figure 5.2, we present the speedup of parallelizing[4] MCM on the four cores of the CPU instead of implementing it on only one core. We notice that the speedup increases quickly according to the number of the simulated trajectories and it reaches

---

[4]We use OpenMP directives.

P1 Vs. P2 for Φ_{min} and Φ_{max}: The real values are equal to 8.262 and 21.15 respectively

| Simulated | The | Time | Price | | Std Deviation | |
Paths	Payoff	Steps	**P1**	**P2(Opt)**	**P1**	**P2(Opt)**
2^{10}	Φ_{min}	10	7.734	7.986	0.190	0.248
2^{10}	Φ_{min}	20	7.618	7.895	0.257	0.270
2^{10}	Φ_{min}	30	7.564	7.920	0.224	0.263
2^{10}	Φ_{max}	10	21.03	20.33	0.66	0.86
2^{10}	Φ_{max}	20	20.46	19.38	0.61	0.73
2^{10}	Φ_{max}	30	19.73	18.13	0.73	0.93
2^{14}	Φ_{min}	10	7.755	8.088	0.058	0.067
2^{14}	Φ_{min}	20	7.584	8.098	0.098	0.052
2^{14}	Φ_{min}	30	7.467	8.087	0.082	0.043
2^{14}	Φ_{max}	10	20.96	20.91	0.09	0.24
2^{14}	Φ_{max}	20	20.58	20.56	0.16	0.16
2^{14}	Φ_{max}	30	20.36	20.05	0.15	0.22
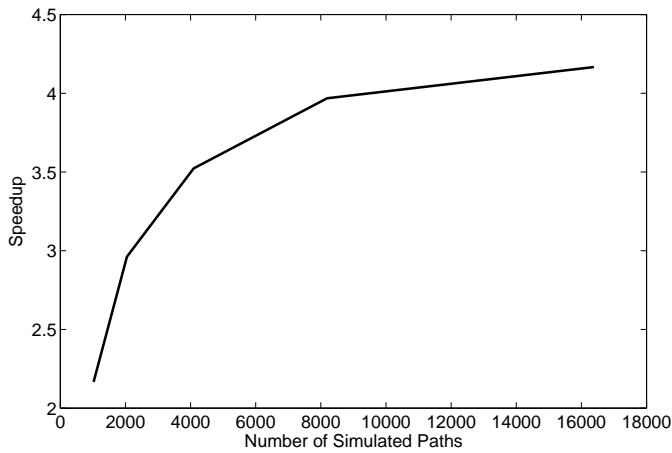
FIG. 5.2. *The speedup of using all the CPU cores according to the number of trajectories.*

a saturation state for > 9000 trajectories. For a large dimensional problem, the maximum speedup obtained is approximately equal to the number of logical cores on the CPU which indicates that MCM is very appropriate for parallel architectures. We point out, however, that our parallelization of MCM is done on the trajectories[5], so the speedup is invariable according to dimensions and time steps.

Regarding GPU implementation, we also use a path parallelization of simulations. In Figure 5.3, we present the speedup of parallelizing[6] MCM on the GPU instead of implementing it on the four cores of the CPU. The speedup increases quickly not only according to the number of simulated trajectories, but also according to the dimension of the contract. The latter fact can be easily explained by the memory hierarchy of the GPU. The speedups provided in Figure 5.3 prove, once again, the high adaptability of MCM on parallel architectures.

---

[5]which is the most natural procedure of parallelizing Monte Carlo.
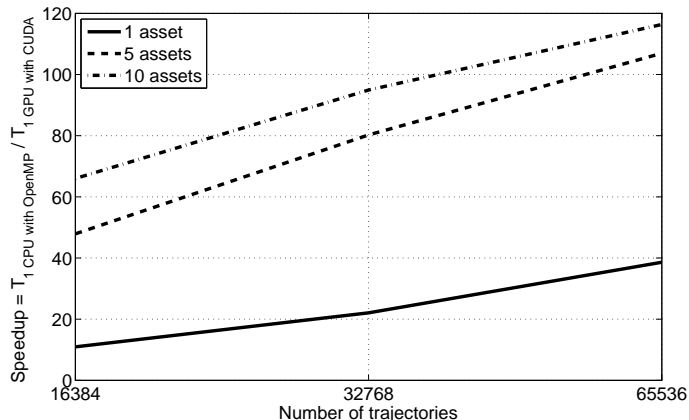[6]We use CUDA language.

Fig. 5.3. *The speedup of using the GPU instead of the CPU cores according to the number of trajectories.*

6. Conclusion. In this article we provided, on the one hand, theoretical results that deal with the continuation computations using the Malliavin calculus and how one can reduce the Monte Carlo variance when simulating this continuation. On the other hand, we presented numerical results related to the accuracy of the prices obtained and the parallel adaptability of the MCM method on multi-core architectures.

As far as the theoretical results are concerned, based on the Malliavin calculus, we provided a generalization of the value of the continuation for the multi-dimensional models with deterministic and non a constant triangular matrix $\sigma(t)$. Moreover, we pointed out that one can effectively reduce the variance by a simple conditioning method. Finally, we presented a less intuitive but very effective variance reduction method based on an appropriate choice of the number of trajectories used to approximate the quotient of two expectations.

Regarding the numerical part, we proved that the one who looks for instantaneous simulations can obtain better and sufficiently good prices with MCM than with LS using only 2^{10} trajectories. Also, unlike LS, our nonparametric variance reduction implementation of MCM does not require parametric regression. Thus we improve the results of the simulation by only increasing the number of trajectories. Finally, increasing the number of trajectories is time consuming but MCM can be effectively parallelized on multi-core CPUs and GPUs. Indeed, the MCM simulation of 2^{14} trajectories using the GTX 480 GPU can be performed within seconds ($< 5s$).

Appendix. Proof of Lemma 2.2. The equality (2.4) can be easily proved. Indeed, using the chain rule

$$D_u^k f(S_t) = \sum_{p=k}^{n} \sigma_{pk}(u) S_t^p \partial_{x_p} f(S_t)$$

Besides, we assumed that $\rho(u) = \sigma^{-1}(u)$ which completes the proof.

$\square$

Proof of Lemma 2.3. Using duality (1.2) we have

$$E\left(h(S_s^k)F\sum_{i=k}^n\int_I\rho_{ik}(u)dW_u^i\right)=E\left(\sum_{i=k}^n\int_I D_u^i\left[h(S_s^k)F\right]\rho_{ik}(u)du\right)$$

$$=E\left(h(S_s^k)\sum_{i=k}^n\int_I D_u^i F\rho_{ik}(u)du\right)+E\left(F\sum_{i=k}^n\int_I h'(S_s^k)\sigma_{ki}(u)\rho_{ik}(u)S_s^k du\right)$$

Moreover, the fact that $\sigma(u)$ and $\rho(u)$ are two triangular matrices such that $\rho_{kk}(u)=1/\sigma_{kk}(u)$ simplifies the last term which can be also rewritten using the Malliavin derivative

$$E\left(F\int_I h'(S_s^k)S_s^k du\right)=E\left(F\int_I\frac{D_u^k h(S_s^k)}{\sigma_{kk}(u)}du\right)$$

This provides the required result.

$\square$

REFERENCES

[1] L. A. Abbas-Turki, S. Vialle, B. Lapeyre, and P. Mercier, "High dimensional pricing of exotic european contracts on a GPU cluster, and comparison to a CPU cluster," *Parallel and Distributed Computing in Finance, in IEEE International Parallel & Distributed Processing Symposium*, May 2009.

[2] L. A. Abbas-Turki and B. Lapeyre, "American options pricing on multicore graphic cards," *IEEE The Second International Conference on Business Intelligence and Financial Engineering*, July 2009.

[3] F. A. Longstaff and E. S. Schwartz, "Valuing American options by simulation: A simple least-squares approach," *The Review of Financial Studies*, vol. 14, no. 1, pp. 113–147, 2001.

[4] P. Glasserman, *Monte Carlo Methods in Financial Engineering.* Applications of Mathematics, Springer, 2003.

[5] E. Clément, D. Lamberton, and P. Protter, "An analysis of a least squares regression algorithm for American option pricing," *Finance and Stochastics*, vol. 17, pp. 448–471, 2002.

[6] J. Tsitsiklis and B. V. Roy, "Regression methods for pricing complex American-style options," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 694–703, 2001.

[7] V. Bally, L. Caramellino, and A. Zanette, "Pricing American options by Monte Carlo methods using a Malliavin calculus approach," *Monte Carlo Methods and Applications*, vol. 11, pp. 97–133, 2005.

[8] V. Bally and G. Pagès, "A quantization algorithm for solving multidimensional discrete-time optimal stopping problems," *Bernoulli*, vol. 9, no. 6, pp. 1003–1049, 2003.

[9] V. Bally, "An elementary introduction to Malliavin calculus," *INRIA Rapport de Recherche*, vol. 4718, 2003.

[10] M. Broadie and J. Detemple, "American option valuation: new bounds, approximations, and a comparison of existing methods securities using simulation," *The Review of Financial Studies*, vol. 9, pp. 1221–1250, 1996.

[11] "http://www-roc.inria.fr/mathfi/Premia/,"

[12] S. Villeneuve and A. Zanette, "Parabolic ADI methods for pricing American options on two stocks," *Mathematics of Operations Research*, vol. 27, pp. 121–149, 2002.