American Options Based on Malliavin Calculus and Nonparametric Variance Reduction Methods

L. A. Abbas-Turki* B. Lapeyre†

March 10, 2015

Abstract

This paper is devoted to pricing American options using Monte Carlo and Malliavin calculus. We introduce two variance reduction techniques, the first one is based on conditioning and the second one relies on an appropriate choice of the number of simulated paths in the computation of the quotient of two expectations. We demonstrate our methods on two types of models, the multidimensional exponential models with deterministic volatility and the Heston model. Since our techniques are well-suited to parallel implementation, our numerical experiments are performed using multi-core CPU and many-core GPU environments.

Premia 17

Introduction and objectives

In this paper, we explore a Monte Carlo (**MC**) method based on **M**alliavin calculus (**MCM**) for pricing **A**merican **O**ptions (**AO**). Unlike usual American option algorithms as **L**ongstaff-**S**chwartz (**LS**) [16] or Malliavin calculus techniques based on localization, the method presented here does not need any parametric regression and higher dimensional problems can be dealt with more easily as the accuracy of results depends only on the number of simulated trajectories.

Assuming that the asset S follows a Markovian model, American contracts can be exercised at any trading date until maturity and their prices are given, at each time t, by (see [14]) $P_t(S_t)$ with

$$P_t(x) = \sup_{\theta \in \mathcal{T}_{t,T}} E_{t,x}\left(e^{-r(\theta-t)}\Phi(S_\theta)\right), \tag{1}$$

*Université Paris-Est, Laboratoire d'Analyse et de Mathématiques Appliquées, Champs-sur-Marne, 77454 Marne-la-Vallée Cedex2, France.

†Université Paris-Est, CERMICS, Projet MathFi, ENPC-INRIA-UMLV, 6 et 8 avenue Blaise Pascal, 77455 Marne La Vallée, Cedex 2, France.

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 given that $S_t = x$ and r and $\Phi(S_t)$ are respectively the instantaneous interest rate and the payoff of the contract.

In order to evaluate numerically the price (1), we first need to approach continuous stopping times in $\mathcal{T}_{t,T}$ with discrete stopping times taking values in the finite set $t = t_0 < t_1 < ... < t_n = T$ (Bermudan approximation). When we do this approximation, pricing American options can be reduced to the implementation of a discrete time dynamic programming algorithm (see [14]). Like the LS algorithm [16], we implement the dynamic programming principle in terms of the optimal stopping times τ_k, for each path, as follows

$$\begin{array}{c} \tau_n = T, \\ \forall k \in \{n-1, ..., 0\}, \quad \tau_k = t_k 1_{A_k} + \tau_{k+1} 1_{A_k^c}, \end{array} \tag{2}$$

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

$$C(S_{t_k}) = E\left(e^{-r\Delta t} P_{t_{k+1}}(S_{t_{k+1}}) \Big| S_{t_k}\right), \tag{3}$$

where $\Delta t = t_{k+1} - t_k$. Thus, to evaluate the price (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 conditional expectation (3). For example the authors of [19] perform a regression to estimate the continuation value, but unlike [16], 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 (2). We refer the reader to [10] for a presentation of the way this estimation is done for the LS algorithm and details on the convergence. Other methods use the Malliavin calculus with localization [6] or the quantization method [7] for $C(S_{t_k})$ computation.

In this work, we rewrite (3) using Malliavin calculus but unlike [6] we use the induction (2) for the implementation and we propose a new method of variance reduction, without using localization. Formally speaking, for a constant r, we can rewrite the conditional expectation using the Dirac distribution $\varepsilon_x(\cdot)$ at point x then using the Malliavin calculus for a large class of diffusion models, we get

$$C(x) = \frac{E\left(e^{-r\Delta t} P_{t_{k+1}}(S_{t_{k+1}}) \varepsilon_x(S_{t_k})\right)}{E\left(\varepsilon_x(S_{t_k})\right)} = \frac{E\left(e^{-r\Delta t} P_{t_{k+1}}(S_{t_{k+1}}) 1_{S_{t_k} \geq x} \Theta_{t_k, t_{k+1}}\right)}{E\left(1_{S_{t_k} \geq x} \Theta_{t_k, t_{k+1}}\right)}. \tag{4}$$

In this paper, we provide the value of $\Theta_{t_k, t_{k+1}}$ for two classes of models: **M**ultidimensional **E**xponential **D**iffusions with deterministic **C**oefficients (**MEDC**) and the **M**ulti-dimensional **H**eston (**MH**) model. In the MEDC models, the dynamics of the assets $\{S_t^i\}_{1 \leq i \leq d}$ is given by

$$\frac{dS_t^i}{S_t^i} = r_i dt + \sum_{j=1}^{i} \sigma_{ij}(t) dW_t^j, \quad S_0^i = z_i, \quad i = 1, .., d.$$

In the MH models, the dynamics of the assets $\{S_t^i\}_{1\leq i\leq d}$ is given by

$$for\ 1\leq i\leq d \qquad \begin{array}{c} d\nu_t^i=\kappa_i(\theta_i-\nu_t^i)dt+\eta_i\sqrt{\nu_t^i}d\widetilde{Z}_t^i, \quad \nu_0^i=y^i,\\ dS_t^i=S_t^i\left(r_idt+\sqrt{\nu_t^i}dZ_t^i\right), \quad S_0^i=z^i. \end{array}$$

For the conditional expectation (4) simulation, two variance reduction methods can be applied for MEDC and MH models as well as for other models. Without loss of generality, for the MEDC model, instead of simulating directly the last term in (4), we project $1_{S_{t_k}\geq x}(S_{t_k})\Theta_{t_k,t_{k+1}}$ using a conditioning as follows

$$C(x)=\frac{E\left(e^{-r\Delta t}P_{t_{k+1}}(S_{t_{k+1}})E\left[1_{S_{t_k}\geq x}\Theta_{t_k,t_{k+1}}\big|\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}\right]\right)}{E\left(E\left[1_{S_{t_k}\geq x}\Theta_{t_k,t_{k+1}}\big|\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}\right]\right)}. \quad (5)$$

Then, the second variance reduction method is applied by setting the continuation value to the approximation below

$$C(x)\approx\frac{\frac{1}{N'}\sum_{l=1}^{N'}e^{-r\Delta t}P_{t_{k+1}}^l(S_{t_{k+1}})h(x,\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l)}{\frac{1}{N}\sum_{l=1}^{N}h(x,\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}^l)}, \quad (6)$$

with $h(x,\{w_{ij}\}_{j\leq i})=E(1_{S_{t_k}\geq x}\Theta_{t_k,t_{k+1}}\big|\{\int_0^{t_{k+1}}\sigma_{ij}(u)dW_u^j\}_{1\leq j\leq i\leq d}=\{w_{ij}\}_{1\leq j\leq i\leq d})$ and $N\neq N'$. Thus, we improve the speed of the convergence using an appropriate relation between N and N' that reduces the variance of the quotient (6). Note that, even if one can also reduce the variance by an "appropriate" control variate, here we choose not to implement this kind of method because it is not standard for American options.

Regarding the numerical simulation, we test MCM on a multi-core CPU (Central Processing Unit) as well as a many-core GPU (Graphic Processing Unit). We will discuss the advantages of the parallel implementation 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.

The outline of this paper is as follows. In section 1 we establish the notations and the Malliavin calculus tools. We give in section 2 (see (14)) the value of $\Theta_{t_k,t_{k+1}}$ for the MEDC model and we extend it to the MH model in section 3. Section 4 is devoted to a variance reduction method based on conditioning and section 5 to the variance reduction method based on the appropriate relation between N and N' (6). In the last section, we show that the multidimensional MCM implementation on a many-core GPU is more than 60 times faster than its implementation on a multi-core CPU. We also provide the numerical comparison between LS and MCM. Finally, we study the results of using the two variance reduction methods (5) and (6) which allow to obtain accurate prices even when simulating only 2^{10} trajectories.

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 a 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 $t \in [0, T]$.

Throughout sections 2 and 3, we will use two operators: The Malliavin derivative D and the Skorohod integral δ and we define them in the same way as in [5]. 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 $\mathcal{S}(\mathbb{R}^{d \times 2^m})$ the Schwartz space of infinitely differentiable and rapidly decreasing functions on $\mathbb{R}^{d \times 2^m}$. Let $f \in \mathcal{S}(\mathbb{R}^{d \times 2^m})$, we define the set $\mathfrak{S}^m$ of simple functionals by

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

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

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

is a process in $L^2(\Omega \times [0, T])$. 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 $\forall F \in \mathbb{D}^{1,2}$

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

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

$$\forall F \in \mathbb{D}^{1,2}, \quad E\left(F \int_0^T U_t \cdot \delta W_t\right) = E\left(F\delta(U)\right) = E\left(\int_0^T U_t \cdot D_t F dt\right), \tag{7}$$

$(\cdot)$ 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 Itô 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 $s < t \in]0, T]$: Assume $F \in \mathbb{D}^{1,2}$, U is an adapted process in $Dom(\delta)$ and $FU \in Dom(\delta)$. For each $1 \leq i \leq d$ we have the following equality
$$E\left(\int_I F U_u \delta^i W_u\right) = E\left(F \int_I U_u dW_u^i\right) - E\left(\int_I U_u D_u^i F du\right). \tag{8}$$

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 i^{th} stock and the i^{th} coordinate of the positive vector x.

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

$$\mathcal{E}_b(\mathbb{R}^d) = \left\{ f \in \mathcal{M}(\mathbb{R}^d) \ : \ \exists C > 0 \ \textit{and} \ \ m \in \mathbb{N}; \ \ |f(y)| \leq C(1 + |y|_d)^m) \right\}, \tag{9}$$

where $\mathcal{M}(\mathbb{R}^d)$ is the set of measurable functions on $\mathbb{R}^d$ and $|\cdot|_d$ is the euclidean norm. 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 for multidimensional exponential diffusions with deterministic coefficients

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

$$\frac{dS_t^i}{S_t^i} = r_i dt + \sum_{j=1}^{i} \sigma_{ij}(t) dW_t^j, \quad S_0^i = z_i, \quad i = 1, .., d, \tag{10}$$

where r_i are constants and $\sigma(t) = \{\sigma_{ij}(t)\}_{1 \leq i,j \leq d}$ 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 ensures the existence of the inverse matrix $\rho(t) = \sigma^{-1}(t)$ and its boundedness. Dynamics (10) is widely used for equity models,

HJM interest rate models and variance swap models. Moreover, one should note that in the case where the dynamics of S is given by local volatility model, we can use a discretization scheme to reduce it to an SDE of type (10) on subintervals.

The first theorem of this section provides the expression of the continuation value (3) when using Malliavin calculus for MEDC models. This theorem can be considered as an extension of the results on the continuation value for the multidimensional model with constant parameters detailed in [6]. In Theorem 2, we provide a closed-form expression for $\Gamma^k_{s,t}$, introduced in Theorem 1. Corollary 1 treats the special case $\sigma_{ij}(t) = \sigma_i \delta(i-j)$ (σ_{ij} is a constant) that will be used, with other models, to test numerically our nonparametric variance reduction methods detailed in section 4 and section 5.

Theorem 1. *For any $s \in]0,t[$, $g \in \mathcal{E}_b(\mathbb{R}^d)$ and $x = (x_1, ..., x_d)$ with $x_i > 0$,*

$$E\left(g(S_t)\Big|S_s = x\right) = \frac{T_{s,t}[g](x)}{T_{s,t}[1](x)}, \tag{11}$$

where $T_{s,t}[f](x)$ is defined for every function[1] *$f \in \mathcal{E}_b(\mathbb{R}^d)$ by*

$$T_{s,t}[f](x) = E\left(f(S_t)\Gamma_{s,t}\prod_{k=1}^{d}\frac{H_k(S^k_s)}{S^k_s}\right), \tag{12}$$

$\Gamma_{s,t} = \Gamma^1_{s,t}$ and $\Gamma^1_{s,t}$ can be computed by the following induction scheme

$$\Gamma^d_{s,t} = \pi^{d,d}_{s,t},\ \textit{for } k \le d-1\text{: } \Gamma^k_{s,t} = \Gamma^{k+1}_{s,t}\pi^{k,d}_{s,t} - \sum_{j=k+1}^{d}\int_0^t D^j_u\Gamma^{k+1}_{s,t}D^j_u\pi^{k,d}_{s,t}du, \tag{13}$$

with

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

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

$H_k(S^k_s)$ is the Heaviside function of the difference between the k^{th} stock and the k^{th} coordinate of the positive vector x, $\mathcal{E}_b(\mathbb{R}^d)$ is defined in (9).

From this theorem we obtain

$$\Theta_{t_k,t_{k+1}} = \frac{\Gamma_{t_k,t_{k+1}}}{\prod_{i=1}^{d}S^i_{t_k}}. \tag{14}$$

To prove Theorem 1, we need the following two lemmas which are proved in the appendix. It follows from Lemma 1 that the sum $\sum_{i=k}^{d}\rho_{ik}(u)D^i_u g(S_t)$ does not depend on u.

[1] In our case $f = g$ or $f = 1$

Lemma 1. *For any $u \in]0,t[$, $f \in \mathcal{C}^1(\mathbb{R}^d)$ and S given by the SDE (10), we have*

$$\sum_{i=k}^{d} \rho_{ik}(u) D_u^i f(S_t) = S_t^k \partial_{x_k} f(S_t). \tag{15}$$

The second lemma is based on the duality property of the Malliavin calculus.

Lemma 2. *For any $I \subset]0,t[$, $h \in \mathcal{C}_b^\infty(\mathbb{R})$, $x \in \mathbb{R}_+^d$, $F \in \mathbb{D}^{1,2}$ and S given by the SDE (10), we have*

$$\begin{array}{rcl} E\left(\int_I \frac{F D_u^k h(S_s^k)}{\sigma_{kk}(u)} du\right) & = & E\left(h(S_s^k) F \sum_{i=k}^d \int_I \rho_{ik}(u) dW_u^i\right) \\ & - & E\left(h(S_s^k) \sum_{i=k}^d \int_I \rho_{ik}(u) D_u^i F du\right). \end{array} \tag{16}$$

Proof. of Theorem 1 We will prove that for $h_i \in \mathcal{C}_b^\infty(\mathbb{R})$, $0 \le k \le d$ and $f \in \mathcal{E}_b(\mathbb{R}^d)$

$$E\left(f(S_t) \prod_{i=1}^{d} h_i'(S_s^i)\right) = E\left(f(S_t) \Gamma_{s,t}^{k+1} \prod_{i=1}^{k} h_i'(S_s^i) \prod_{i=k+1}^{d} \frac{h_i(S_s^i)}{S_s^i}\right) \tag{17}$$

and that Theorem 1 is obtained directly from (17) by setting $k = 0$.

Step 1: ((17) with $k = 0$) $\Rightarrow$ (11).

Heuristically $E\left(g(S_t) \middle| S_s = x\right)$ can be viewed as $E\left(g(S_t)\varepsilon_x(S_s)\right) / E\left(\varepsilon_x(S_s)\right)$ where ε_x is the Dirac distribution at x and we know that $\varepsilon_{x_i} = H_i'$. In order to make our reasoning rigorous, we will replace the expectation of $\varepsilon_x(S_s)$ by the density of S_s evaluated at x.

Let $\phi \in \mathcal{C}_c^\infty(\mathbb{R})$ be a mollifier function with support equal to $[-1,1]$ and such that $\int_{\mathbb{R}} \phi(u) du = 1$, then for any $u \in \mathbb{R}$ we define

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

If the equality (17) is correct for any k, then it is correct for $k = 0$ which means

$$E\left(f(S_t) \prod_{k=1}^{d} h_{mk}'(S_s^k)\right) = E\left(f(S_t) \Gamma_{s,t} \prod_{k=1}^{d} \frac{h_{mk}(S_s^k)}{S_s^k}\right). \tag{18}$$

On the one hand, $h_{mk}(u)$ converges to $H_k(u)$ except at $u = x_k$ 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 $p \ge 1$. 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, $h'_{mk}(u_k) = \int_{\mathbb{R}} H_k(v_k)\phi'_m(u_k - v_k)dv_k = \phi_m(u_k - x_k)$. 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 log-normal joint density with respect to the Lebesgue measure on $\mathbb{R}_+^d \times \mathbb{R}_+^d$ we denote it by $p_{s,t}(u, v)$ with $u = (u_1, ..., u_d)$ and $v = (v_1, ..., v_d)$ such that

$$E\left(f(S_t)\prod_{k=1}^{d} h'_{mk}(S_s^k)\right) = \int_{\mathbb{R}^d} f(v)\left(\int_{\mathbb{R}^d}\prod_{k=1}^{d}\phi_m(u_k - x_k)p_{s,t}(u, v)du_1...du_d\right)dv_1...dv_d$$

Because $\int_{\mathbb{R}^d}\prod_{k=1}^{d}\phi_m(u_k - x_k)p_{s,t}(u, v)du_1...du_d$ converges to $p_{s,t}(x, v)$ and due to the regularity properties of the density and the growth condition on f, we easily have

$$E\left(f(S_t)\prod_{k=1}^{d} h'_{mk}(S_s^k)\right) \longrightarrow \int_{\mathbb{R}^d} f(v)p_{s,t}(x, v)dv_1...dv_d,$$

which concludes this step of the proof.

Step 2: We prove (17). Note that by a standard density argument of $\mathfrak{S}$ in $L^2(\Omega)$, we can assume $f \in \mathcal{C}^1(\mathbb{R}^d) \cap \mathcal{E}_b(\mathbb{R}^d)$.
We prove (17) by induction, we introduce the following notations:

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

When $k = d$, we have by the chain rule $h'_d(S_s^d) = \frac{D_u^d h_d(S_s^d)}{D_u^d S_s^d}$ and $D_u^d S_s^d = \sigma_{dd}(u)S_s^d$, thus

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

Using Lemma 2 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 d^{th} coordinate of the Brownian motion yields

$$\begin{aligned} &E\left(\tfrac{1}{s}\textstyle\int_0^s f(S_t)\widehat{h'}_{d-1}(S_s)\frac{D_u^d h_d(S_s^d)du}{\sigma_{dd}(u)S_s^d}\right) \\ &= E\left(Fh_d(S_s^d)\tfrac{1}{s}\textstyle\int_0^s \frac{dW_u^d}{\sigma_{dd}(u)}\right) - E\left(h_d(S_s^d)\tfrac{1}{s}\textstyle\int_0^s D_u^d\frac{\widehat{h'}_{d-1}(S_s)f(S_t)}{S_s^d}\frac{du}{\sigma_{dd}(u)}\right) \qquad (19) \\ &= E\left(Fh_d(S_s^d)\tfrac{1}{s}\textstyle\int_0^s \frac{dW_u^d}{\sigma_{dd}(u)}\right) - E\left(\widehat{h'}_{d-1}(S_s)h_d(S_s^d)\tfrac{1}{s}\textstyle\int_0^s D_u^d\frac{f(S_t)}{S_s^d}\frac{du}{\sigma_{dd}(u)}\right). \end{aligned}$$

Besides using Lemma 1 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 (19) is given by

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

And by duality (7) we remove the Malliavin derivative of $f(S_t)$ in the last term of the previous equality

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

Regrouping all terms together gives

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

Now, let us suppose that (17) is satisfied for k and prove it for $k-1$. We have by the chain rule $h_k'(S_s^k) = \frac{D_u^k h_k(S_s^k)}{D_u^k S_s^k}$ and $D_u^k S_s^k = \sigma_{kk}(u) S_s^k$, thus

$$\begin{aligned} E\left(f(S_t)\widehat{h'}_{d-1}(S_s)\right) &= E\left(f(S_t)\Gamma_{s,t}^{k+1}\widehat{h}_{k+1}^d(S_s)\widehat{h'}_k(S_s)\right) \\ &= E\left(\tfrac{1}{s}\int_0^s f(S_t)\Gamma_{s,t}^{k+1}\widehat{h}_{k+1}^d(S_s)\widehat{h'}_{k-1}(S_s)\tfrac{D_u^k h_k(S_s^k)}{\sigma_{kk}(u)S_s^k} du\right) \\ &= E\left(\tfrac{1}{s}\int_0^s \tfrac{f(S_t)\Gamma_{s,t}^{k+1}\widehat{h}_{k+1}^d(S_s)\widehat{h'}_{k-1}(S_s)}{S_s^k}\tfrac{D_u^k h_k(S_s^k)}{\sigma_{kk}(u)} du\right). \end{aligned}$$

Using Lemma 2 with

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

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

$$\begin{aligned} &E\left(\tfrac{1}{s}\int_0^s \tfrac{F D_u^k h_k(S_s^k)}{\sigma_{kk}(u)} du\right) = \textstyle\sum_{j=k}^d E\left(F h_k(S_s^k)\tfrac{1}{s}\int_0^s \rho_{jk}(u) dW_u^j\right) \\ &- \textstyle\sum_{j=k}^d E\left(h_k(S_s^k)\widehat{h'}_{k-1}(S_s)\tfrac{1}{s}\int_0^s D_u^j\left[\tfrac{f(S_t)\widehat{h}_{k+1}^d(S_s)\Gamma_{s,t}^{k+1}}{S_s^k}\right]\rho_{jk}(u) du\right). \end{aligned} \tag{20}$$

Besides, if for $x = (x_1, ..., x_d)$ we denote $\Pi(x) = \frac{\widehat{h}_{k+1}^d(x)}{x_k}$, the Malliavin derivative of the last term of (20) provides

$$\begin{aligned} D_u^j\left[\Gamma_{s,t}^{k+1}\Pi(S_s)f(S_t)\right] = D_u^j\Gamma_{s,t}^{k+1}\Pi(S_s)f(S_t) &+ \Gamma_{s,t}^{k+1} D_u^j \Pi(S_s) f(S_t) \\ &+ \Gamma_{s,t}^{k+1}\Pi(S_s) D_u^j f(S_t). \end{aligned}$$

Using Lemma 1 for the Malliavin derivative in the two last terms, we get

$$\sum_{j=k}^{d} \rho_{jk}(u) D_u^j \Pi(S_s) = S_s^k \partial_{x_k} \Pi(S_s) = -\Pi(S_s), \tag{21}$$

$$\sum_{j=k}^{d} \rho_{jk}(u) D_u^j f(S_t) = S_t^k \partial_{x_k} f(S_t). \tag{22}$$

From (21), we deduce that

$$\widehat{h'}_{k-1}(S_s) h_k(S_s^k) f(S_t) \Gamma_{s,t}^{k+1} \frac{1}{s} \int_0^s \sum_{j=k}^{d} \rho_{jk}(u) D_u^j \Pi(S_s) du = -\frac{\widehat{h'}_{k-1}(S_s) \widehat{h}_k^d(S_s) f(S_t) \Gamma_{s,t}^{k+1}}{S_s^k}.$$

Thus, introducing the random variable $\widetilde{\pi}_{s,t}^{k,d} = 1 + \frac{1}{s} \sum_{j=k}^{d} \int_0^s \rho_{jk}(u) dW_u^j$ and using (20)

$$\begin{aligned} E\left(\tfrac{1}{s}\int_0^s \tfrac{F D_u^k h_k(S_s^k)}{\sigma_{kk}(u)} du\right) &= E\left(\tfrac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s) f(S_t)\Gamma_{s,t}^{k+1}}{S_s^k} \widetilde{\pi}_{s,t}^{k,d}\right) \\ &- E\left(\tfrac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s) f(S_t)}{S_s^k} \tfrac{1}{s}\int_0^s \sum_{j=k}^{d} \rho_{jk}(u) D_u^j \Gamma_{s,t}^{k+1} du\right) \\ &- E\left(\tfrac{\widehat{h}_k^d(S_s)\widehat{h'}_{k-1}(S_s)\Gamma_{s,t}^{k+1}}{S_s^k} \tfrac{1}{t-s}\int_s^t \sum_{j=k}^{d} \rho_{jk}(u) D_u^j f(S_t) du\right), \end{aligned} \tag{23}$$

where we used the fact (22) 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 (23)

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

We applied (7) in the third equality to remove the Malliavin derivative of $f(S_t)$. We also used (8) 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 $\mathcal{F}_t^{k+1,\dots,d}$-measurable random variable ($\mathcal{F}_t^{i,\dots,d}$ defined in section 1)

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

Although $D_u^j\pi_{s,t}^{k,d} = \varphi_{jk}(u)$, note that the Malliavin derivative of $\Gamma_{s,t}^{k+1}$ intervenes in the induction (13) which is difficult to compute numerically. Consequently, we propose in Theorem 2 a new formula which enables us to get rid of the Malliavin derivatives and its computation can be easily done using (27).

We will use in Theorem 2 the set of the second order permutations $\overline{\mathcal{P}}_{k,d}$ defined as the following

$$\overline{\mathcal{P}}_{k,d} = \{p \in \mathcal{P}_{k,d};\ p \circ p = Id\}, \tag{24}$$

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

$$\overline{\mathcal{P}}_{k,d} = \{\tau_k^k \circ p;\ p \in \overline{\mathcal{P}}_{k+1,d}\} \cup \{\tau_k^l \circ p;\ p \in \overline{\mathcal{P}}_{k+1,d},\ p(l) = l,\ l \in \{k+1, ..., d\}\}, \tag{25}$$

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

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

where $\epsilon(p)$ is the signature of the permutation p.

Theorem 2. *Based on the assumptions of Theorem 1, for $k \in \{1, ..., d\}$ the value of $\Gamma_{s,t}^k$ is given by*

$$\Gamma_{s,t}^{k} = \sum_{p\in\overline{\mathcal{P}}_{k,d}} \epsilon(p)A_{k,p(k)}A_{k+1,p(k+1)}...A_{d,p(d)} = \sum_{p\in\overline{\mathcal{P}}_{k,d}} \epsilon(p)\prod_{i=k}^{d} A_{i,p(i)}, \tag{26}$$

with $\epsilon(p)$ as the signature of the permutation $p \in \overline{\mathcal{P}}_{k,d}$, $\overline{\mathcal{P}}_{k,d}$ defined in (24) 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 $\pi_{s,t}^{l,d}$.

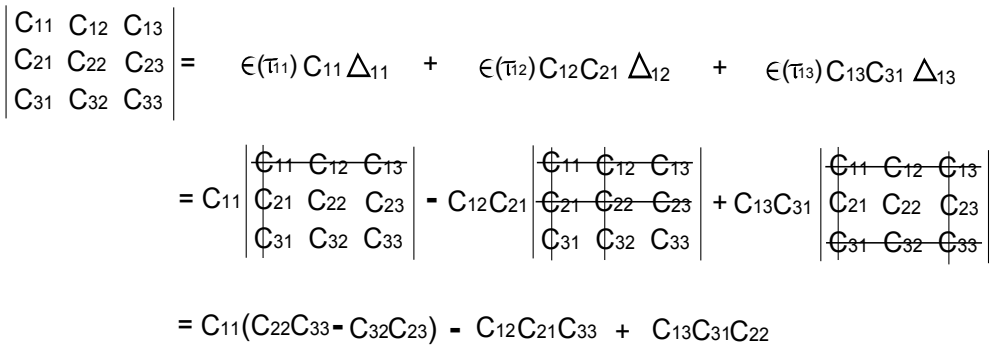

Figure 1: Illustration of the computation of Δ (27) for $d = 3$ and $k = 1$.

Remark: In the theorem above, $C_{k,l}$ admits a closed-form expression because $\pi^{k,d}_{s,t}$ and $\pi^{l,d}_{s,t}$ are two correlated Gaussian variables whose general value is given in Theorem 1. Please remark that $\Gamma^1_{s,t} = \Gamma_{s,t}$ is a determinant that involves only the permutations of $\overline{\mathcal{P}}_{1,d}$ and for a determinant Δ, associated to an arbitrary $(d-k+1)\times(d-k+1)$ matrix $C = \{C_{i,j}\}_{k\le i,j\le d}$, whose permutations are in $\overline{\mathcal{P}}_{k,d}$, we use (25) to prove easily that

$$\Delta = C_{k,k}\Delta_{k,k} + \sum_{i=k+1}^{d} \epsilon(\tau^i_k) C_{i,k} C_{k,i} \Delta_{k,i}, \tag{27}$$

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 (27) provides a recursive formula which is even more efficient than the determinant formula. Of course, we can generalize the relation (27) to the one that involves the development according to a j^{th} line or a j^{th} column with $k \le j \le d$. In Figure 1, we provide an illustration of the computation of Δ when C is a 3×3 matrix.

Proof. of Theorem 2 We prove (26) by a decreasing induction. For $k = d$, the expression (26) is clearly satisfied. We suppose that (26) is satisfied for $k+1$ and we prove it for k. According to Theorem 1, $\Gamma^k_{s,t} = \Gamma^{k+1}_{s,t}\pi^{k,d}_{s,t} - \sum_{j=k+1}^d \int_0^t D^j_u \Gamma^{k+1}_{s,t} D^j_u \pi^{k,d}_{s,t} du$, but

$$\begin{aligned} D^j_u \Gamma^{k+1}_{s,t} &= \sum_{l=k+1}^d \sum_{p\in\overline{\mathcal{P}}_{k+1,d}} \epsilon(p) \prod_{i=k+1, i\neq l}^d A_{i,p(i)} D^j_u A_{l,p(l)} \\ &= \sum_{l=k+1}^d \sum_{p\in\overline{\mathcal{P}}_{k+1,d}, p(l)=l} \epsilon(p) \prod_{i=k+1, i\neq l}^d A_{i,p(i)} D^j_u A_{l,l}, \end{aligned}$$

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

Subsequently

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

Finally

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

The last equality is due to the development of $\sum_{p\in\overline{\mathcal{P}}_{k,d}}\epsilon(p)\prod_{i=k}^{d}A_{i,p(i)}$ according to the k^{th} line of A which can be justified by (25).

As a corollary of Theorem 1 and Theorem 2, we obtain the following result for the multidimensional Black & Scholes model with independent coordinates

Corollary 1. *For any $s\in]0,t[$, $g\in\mathcal{E}_b(\mathbb{R}^d)$ and $x=(x_1,...,x_d)$ with $x_i>0$, if $\sigma_{ij}(t)=\sigma_i\delta(i-j)$ then*

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

with

$$T_{s,t}[f](x)=E\left(f(S_t)\prod_{k=1}^{d}\frac{H_k(S_s^k)W_{s,t}^k}{\sigma_k s(t-s)S_s^k}\right), \tag{28}$$

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 Extension to the multidimensional Heston model

In this section, we consider the multidimensional Heston model

$$\text{for } 1\le i\le d\qquad \begin{array}{c} d\nu_t^i=\kappa_i(\theta_i-\nu_t^i)dt+\eta_i\sqrt{\nu_t^i}d\widetilde{Z}_t^i,\quad \nu_0^i=y^i,\\ dS_t^i=S_t^i\left(r_idt+\sqrt{\nu_t^i}dZ_t^i\right),\quad S_0^i=z^i,\end{array} \tag{29}$$

where $(Z^1,...,Z^d,\widetilde{Z}^1,...,\widetilde{Z}^d)$ is a vector of correlated Brownian motions with R as a non-singular correlation matrix. The first step is to rewrite (29) using independent Brownian motions by the Cholesky decomposition of $R = LL'$ where L is a lower triangular matrix which provides

$$\begin{pmatrix} d\nu_t^1 \\ \vdots \\ d\nu_t^d \\ dS_t^1 \\ \vdots \\ dS_t^d \end{pmatrix} = \begin{pmatrix} \kappa_1(\theta_1-\nu_t^1) \\ \vdots \\ \kappa_d(\theta_d-\nu_t^d) \\ r_1 S_t^1 \\ \vdots \\ r_d S_t^d \end{pmatrix} dt + diag \begin{pmatrix} \eta_1\sqrt{\nu_t^1} \\ \vdots \\ \eta_d\sqrt{\nu_t^d} \\ \sqrt{\nu_t^1}S_t^1 \\ \vdots \\ \sqrt{\nu_t^d}S_t^d \end{pmatrix} L \begin{pmatrix} d\widetilde{W}_t^1 \\ \vdots \\ d\widetilde{W}_t^d \\ dW_t^1 \\ \vdots \\ dW_t^d \end{pmatrix}, \quad \begin{matrix} \nu_0^i = y^i \\ S_0^i = z^i \end{matrix}, \tag{30}$$

where $(\widetilde{W}^1,...,\widetilde{W}^d,W^1,...,W^d)$ is a vector of independent Brownian motions. Because the matrix L is a lower triangular matrix, conditionally to the Brownian motions $(\widetilde{W}^1,...,\widetilde{W}^d)$, the dynamics of the asset vector $S=(S^1,...,S^d)$ is similar to the one given in (10). This basic argument is the first we use to extend the results of the previous section to the multidimensional Heston model and it can also be used with other stochastic volatility models. Indeed, it allows us to use the Malliavin calculus directly on $(W^1,...,W^d)$ as in the previous section and to completely forget the dependence on $(\widetilde{W}^1,...,\widetilde{W}^d)$. The second argument used in our extension is based on the following result, proved in [12].

Lemma 3. *$E\left[\left(\int_0^t \nu_s^i ds\right)^r\right]$ is finite for all $r\in\mathbb{R}$ and $i\in\{1,...,d\}$.*

Before stating Theorem 3, we decompose the matrix L into three $d\times d$ blocks and we define the matrix $\sigma(u)$ using the third block σ.

$$L = \begin{pmatrix} \sigma' & 0 \\ \sigma'' & \sigma \end{pmatrix}, \quad \sigma(u) = \begin{pmatrix} \sqrt{\nu_u^1}\sigma_{11} & 0 & ... & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \sqrt{\nu_u^{d-1}}\sigma_{d-11} & ... & \sqrt{\nu_u^{d-1}}\sigma_{d-1d-1} & 0 \\ \sqrt{\nu_u^d}\sigma_{d1} & ... & \sqrt{\nu_u^d}\sigma_{dd-1} & \sqrt{\nu_u^d}\sigma_{dd} \end{pmatrix} \tag{31}$$

Theorem 3. *For any $s\in]0,t[$ let*

$$\Gamma_{s,t} = \sum_{p\in\overline{\mathcal{P}}_{1,d}} \epsilon(p) A_{1,p(1)}A_{2,p(2)}...A_{d,p(d)} = \sum_{p\in\overline{\mathcal{P}}_{1,d}} \epsilon(p)\prod_{i=1}^d A_{i,p(i)}, \tag{32}$$

with $\epsilon(p)$ as the signature of the permutation $p\in\overline{\mathcal{P}}_{1,d}$, $\overline{\mathcal{P}}_{1,d}$ defined in (24) and

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

$$\pi_{s,t}^{k,d} = 1 + \sum_{j=k}^{d} \int_0^t \varphi_{jk}(u) dW_u^j, \quad \varphi_{jk}(u) = \frac{1}{s} \frac{\rho_{jk}}{\sqrt{\nu_u^k}} 1_{u \in]0,s[} - \frac{1}{t-s} \frac{\rho_{jk}}{\sqrt{\nu_u^k}} 1_{u \in]s,t[}, \quad (33)$$

and

$$C_{k,l} = \sum_{j=k}^{d} \frac{\rho_{jk}\rho_{jl}}{s^2} \int_0^s \frac{du}{\sqrt{\nu_u^k \nu_u^l}} + \sum_{j=k}^{d} \frac{\rho_{jk}\rho_{jl}}{(t-s)^2} \int_s^t \frac{du}{\sqrt{\nu_u^k \nu_u^l}}. \quad (34)$$

ρ is the inverse matrix $\rho = \sigma^{-1}$ and σ is the third-block matrix in the decomposition (31). If there is $1 < q < \infty$ such that $\Gamma_{s,t} \in L^q(\Omega)$ then, for $g \in \mathcal{E}_b(\mathbb{R}^d)$ and $x = (x_1, ..., x_d)$ with $x_i > 0$

$$E\left(g(S_t) \middle| S_s = x\right) = \frac{T_{s,t}[g](x)}{T_{s,t}[1](x)}, \quad (35)$$

where $T_{s,t}[f](x)$ is defined for every function $f \in \mathcal{E}_b(\mathbb{R}^d)$ by

$$T_{s,t}[f](x) = E\left(f(S_t)\Gamma_{s,t} \prod_{k=1}^{d} \frac{H_k(S_s^k)}{S_s^k}\right). \quad (36)$$

Proof. of Theorem 3 To prove Theorem 3, it is sufficient to prove the following recursive relation for $k = 0$, $h_i \in \mathcal{C}_b^\infty(\mathbb{R})$ and $f \in \mathcal{E}_b(\mathbb{R}^d)$

$$E\left(E\left[f(S_t)\prod_{i=1}^{d} h_i'(S_s^i)\middle|\widetilde{\mathcal{F}}_t\right]\right) = E\left(E\left[f(S_t)\Gamma_{s,t}^{k+1}\prod_{i=1}^{k} h_i'(S_s^i)\prod_{i=k+1}^{d}\frac{h_i(S_s^i)}{S_s^i}\middle|\widetilde{\mathcal{F}}_t\right]\right), \quad (37)$$

where $\widetilde{\mathcal{F}}_t$ is the completed filtration generated by $(\widetilde{W}^1, ..., \widetilde{W}^d)$ until t. If we subdivide this proof into two steps, Step 2 is similar to the one in the proof of Theorem 1 because, as we said earlier, conditionally to $(\widetilde{W}^1, ..., \widetilde{W}^d)$, the processes $\{\nu_t^i\}_{1 \le i \le d}$ can be considered as deterministic. Moreover the expression of $\Gamma_{s,t}$ can be found in the same fashion as for Theorem 2.

Step 1: *((37) with $k = 0$) $\Rightarrow$ (35)*.
Let $\phi \in \mathcal{C}_c^\infty(\mathbb{R})$ be a mollifier function with support equal to $[-1, 1]$ and such that $\int_{\mathbb{R}} \phi(u)du = 1$, then for any $u \in \mathbb{R}$ we define

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

If the equality (37) is correct for any k, then it is correct for $k = 0$ which means

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

The proof of the convergence

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

is also similar to the one in the proof of Theorem 1, the only difference is due to the replacement of the Cauchy-Schwartz inequality by the Hölder inequality that uses the L^q-boundedness of $\Gamma_{s,t}$.

Besides, $h'_{mk}(u_k) = \int_{\mathbb{R}} H_k(v_k)\phi'_m(u_k - v_k)dv_k = \phi_m(u_k - x_k)$ and the distribution of the vector $(S^1_s, ..., S^d_s, S^1_t, ..., S^d_t)$, conditionally to $(\widetilde{W}^1, ..., \widetilde{W}^d)$, admits a log-normal joint density with respect to the Lebesgue measure on $\mathbb{R}^d_+ \times \mathbb{R}^d_+$, we denote it by $\widetilde{p}_{s,t}(u, v)$ with $u = (u_1, ..., u_d)$ and $v = (v_1, ..., v_d)$ where

$$\widetilde{p}_{s,t}(u, v) = \frac{1}{(\det(\Sigma_1)\det(\Sigma_2))^{\frac{1}{2}}} q(u, v)$$

and

$$q(u, v) = \frac{\prod_{i=1}^d 1_{u_i>0, v_i>0}}{(2\pi)^d \prod_{i=1}^d u_i \prod_{i=1}^d v_i} \exp\begin{pmatrix} -\frac{1}{2}(\ln u - d_1)'\Sigma_1^{-1}(\ln u - d_1) \\ -\frac{1}{2}(\ln v - \ln u - d_2)'\Sigma_2^{-1}(\ln v - \ln u - d_2) \end{pmatrix},$$

$d_1 = \left(sr_1 - \frac{1}{2}\Sigma_1^{11}, ..., sr_d - \frac{1}{2}\Sigma_1^{dd}\right)$, $d_2 = \left((t-s)r_1 - \frac{1}{2}\Sigma_2^{11}, ..., (t-s)r_d - \frac{1}{2}\Sigma_2^{dd}\right)$, $\Sigma_1 = \int_0^s \sigma(w)\sigma'(w)dw$, $\Sigma_2 = \int_s^t \sigma(w)\sigma'(w)dw$ and $\sigma(w)$ is given in (31), thus

$$E\left(f(S_t)\prod_{k=1}^d h'_{mk}(S^k_s)\Big|\widetilde{\mathcal{F}}_t\right) = \frac{\int_{\mathbb{R}^d} f(v)\Big(\int_{\mathbb{R}^d}\prod_{k=1}^d \phi_m(u_k - x_k)q(u, v)du_1...du_d\Big)dv_1...dv_d}{(\det(\Sigma_1)\det(\Sigma_2))^{\frac{1}{2}}}$$

To prove the convergence

$$E\left(f(S_t)\prod_{k=1}^d h'_{mk}(S^k_s)\right) \longrightarrow E\left(\int_{\mathbb{R}^d} f(v)\widetilde{p}_{s,t}(x, v)dv_1...dv_d\right),$$

we should first remove the term $(\det(\Sigma_1)\det(\Sigma_2))^{\frac{1}{2}}$ using the Cauchy-Schwarz inequality thanks to Lemma 3, then we use the convergence of $\int_{\mathbb{R}^d}\prod_{k=1}^d \phi_m(u_k - x_k)q(u, v)du_1...du_d$ to $q(x, v)$ as in *Step 1* of the proof of Theorem 1.

In Theorem 3, we made the assumption that $\Gamma_{s,t} \in L^q(\Omega)$ and one should find the parameters κ_i, θ_i and η_i of ν^i_u that fulfill this condition. In this article, we test only the one-dimensional Heston model for which the Feller conditions are sufficient to ensure that $\Gamma_{s,t} \in L^2(\Omega)$. Indeed, if $d = 1$, $\Gamma_{s,t} = \pi^{1,1}_{s,t}$ and it is sufficient to prove that $\int_0^t \frac{du}{\nu^1_u} \in L^1(\Omega)$ which is given in the following lemma. Because $d = 1$, in the lemma below, we remove the dimension index.

Lemma 4. *If $\kappa \geq 0$ and $2\kappa\theta \geq \eta^2$ then $E\left(\int_0^t \frac{du}{\nu_u}\right)$ is finite.*

Proof. of Lemma 4 According to Lemma A.2. in [8]

$$E\exp\left(\frac{\eta^2\left(\frac{2\kappa\theta}{\eta^2} - 1\right)^2}{8}\int_0^t \frac{du}{\nu_u}\right) < \infty$$

and the finiteness of $E\left(\int_0^t \frac{du}{\nu_u}\right)$ follows directly from the application of the Jensen's inequality on the logarithmic function.

4 Variance reduction method based on conditioning

As was said in the previous section, conditionally to the Brownian motions that generate the volatilities, studying the stochastic volatility model (30) is equivalent to studying the MEDC model (10). Thus, except for Theorem 6, in this section we suppose that the price of the asset S_t is given by (10) for which 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 the closed-form expression of $T_{s,t}[1](x)$.

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

$$h(x, w_{ij}) = E\left(\Gamma_{s,t}\prod_{k=1}^d \frac{H_k(S_s^k)}{S_s^k}\Big|\left\{\int_0^t \sigma_{ij}(u)dW_u^j\right\}_{1\le j\le i\le d} = \{w_{ij}\}_{1\le j\le i\le d}\right) \quad (38)$$

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

The computation of $h(x, \{w_{ij}\}_{1\le j\le i\le d})$ 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}

$$\int_0^t \varphi_{jk}(u)dW_u^j = \sum_{i=j}^d a_{i,k}^j Y_{ij} + X_{jk}, \quad (39)$$

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

where $\{X_{jk}\}_{1\leq k\leq j\leq d}$ is a Gaussian vector $\mathcal{N}(0, C_X)$ orthogonal to Y. Using Itô 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}

$$\int_0^s \sigma_{kj}(u)dW_u^j = \sum_{i=j}^{d} b_{i,k}^j Y_{ij} + Z_{kj}, \tag{40}$$

where $\{Z_{kj}\}_{1\leq j\leq k\leq d}$ is a Gaussian vector $\mathcal{N}(0, C_Z)$ orthogonal to Y. Using Itô 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 (39), (40) and the value of A and B, the covariance matrices C_X, C_Z and $C_{XZ} = E(XZ)$ are given by ($\Phi_{jt}^{i,k} = \int_0^t \varphi_{ji}(u)\varphi_{jk}(u)du$)

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

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

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

Employing (39) and (40), 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 $h(x, w_{ij})$ [3]. In Theorem 4, we give an explicit expression of $T_{s,t}[1](x)$ and $h(x, w_{ij})$ in the case of multidimensional B&S models with independent coordinates.

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

Regarding the model (10), we see that now that we know the explicit value of $T_{s,t}[1](x)$ and $h(x,\{w_{ij}\}_{1\leq j\leq i\leq d})$, subsequently, we should choose between the simulation of:
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 value to

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

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 value to

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

Based on a variance reduction argument, Proposition 1 will indicate asymptotically, for some cases, the preferable method to use. In the case of the multidimensional Heston model, please note that $T_{s,t}[1](x)$ is not given explicitly, however we explicitly know the value of the function

$$h(x,\{w_{ij}\}_{j\leq i})=E\left(1_{S_s\geq x}\Gamma_{s,t}\prod_{k=1}^d\frac{H_k(S_s^k)}{S_s^k}|\widetilde{\mathcal{F}}_t\bigvee\{\int_0^t\sqrt{\nu_u^i}dW_u^j\}_{j\leq i}=\{w_{ij}\}_{j\leq i}\right), \quad (41)$$

for $1\leq j,i\leq d$. Thus, we will exclusively use a **P2)** alike procedure, that is to say, simulate N' paths of $g(S_t)h\left(x,\{\int_0^t\sqrt{\nu_u^i}dW_u^j\}_{i,j}\right)$ and N paths of $h\left(x,\{\int_0^t\sqrt{\nu_u^i}dW_u^j\}_{i,j}\right)$ then set the continuation value to

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

In this approximation of the continuation value, the trajectory index l is on the function h because it resulted from a conditioning according to $\widetilde{\mathcal{F}}_t$ and consequently h is not deterministic.

We provide in Theorem 4, Theorem 5 and Theorem 6 the expression of the conditioning for three cases that will be tested in section 6. The proofs of these theorems are given in the appendix. Unlike in the Theorem 4, in Theorem 5 and Theorem 6 we only give the expression of the function h because we will only use the procedure **P2)** for the simulation.

Theorem 4. *We suppose that S_t has the dynamics (10) and $\sigma_{ij}(t)=\sigma_{ij}\delta(i-j)$ then, by conditioning, the function h defined in (38) and the denominator $T_{s,t}[1](x)$ given in Theorem 1 have the following values*

$$T_{s,t}[1](x)=\prod_{k=1}^d\frac{e^{(\sigma_k^2-r_k)s}}{\sigma_k S_0^k}\frac{1}{\sqrt{s2\pi}}e^{-\frac{\widetilde{d}_{x_k}^2}{2}},\quad \widetilde{d}_{x_k}=\frac{\ln\left(\frac{x_k}{S_0^k}\right)-r_k s+\frac{3\sigma_k^2 s}{2}}{\sigma_k\sqrt{s}}$$

and

$$h(x,\{w_k\}_{1\le k\le d}) = \prod_{k=1}^{d} \frac{e^{(\sigma_k^2-r_k)s}}{\sigma_k S_0^k}\sqrt{\frac{t}{s(t-s)2\pi}}\exp\left(\frac{-s\sigma_k}{t}\left(\frac{s\sigma_k}{2}+w_k\right)-\frac{(d_{x_k}(w_k))^2}{2}\right),$$

with

$$d_{x_k}(w_k) = \left(\ln\left(\frac{x_k}{S_0^k}\right) - r_k s + \frac{3\sigma_k^2 s}{2} - (s\sigma_k + w_k)\frac{s\sigma_k}{t}\right) \Big/ \left(\sigma_k\sqrt{s(t-s)/t}\right).$$

Theorem 5. *We suppose that S_t has the dynamics (10) and that $d = 2$ with ρ, σ_1 and σ_2 three constants such that*

$$\sigma(u) = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2 \end{pmatrix}, \quad |\rho| < 1.$$

By conditioning, the expression of the function h defined in (38) is given by

$$\begin{aligned} h(x,w_1,w_2) &= E\left(\Gamma_{s,t}\prod_{k=1}^{2}\frac{H_k(S_s^k)}{S_s^k}\Big|W_t^1 = w_1, W_t^2 = w_2\right) \\ &= \frac{e^{\left(\frac{(\sigma_1^2+\sigma_2^2)}{2}-r_1-r_2\right)s+\frac{s(t-s)(\sigma_1^2+2\rho\sigma_1\sigma_2+\sigma_2^2)}{2t}-\frac{(\sigma_1+\rho\sigma_2)sw_1+\sigma_2\sqrt{1-\rho^2}sw_2}{t}}}{S_0^1 S_0^2}\Lambda_{x_1,x_2,w_1,w_2}, \end{aligned} \tag{42}$$

with

$$\begin{aligned} \Lambda_{x_1,x_2,w_1,w_2} &= \frac{t\rho(\sigma_2-\sigma_1)}{s(t-s)(1-\rho^2)\sigma_1^2\sigma_2^2}\Lambda^1_{x_1,x_2,w_1,w_2} + \frac{t\left((1-\rho^2)\sigma_2+\rho^2\sigma_1\right)}{s(t-s)\sigma_1\sigma_2^2 2\pi\sqrt{1-\rho^2}}\Lambda^2_{x_1,x_2,w_1,w_2} \\ &+ \sqrt{\frac{t}{s(t-s)}}\frac{\rho(\sigma_1-\sigma_2)}{\sigma_1\sigma_2}\left[1+\frac{d_2\sqrt{1-\rho^2}}{\sigma_2}\sqrt{\frac{t}{s(t-s)}}\right]\frac{1}{\sqrt{2\pi}}\Lambda^3_{x_1,x_2,w_1,w_2}, \end{aligned}$$

and

$$\Lambda^1_{x_1,x_2,w_1,w_2} = \frac{1}{2\pi\sqrt{1-\rho^2}}\int_{-\infty}^{d_1}\int_{-\infty}^{d_2\sqrt{1-\rho^2}} e^{-\frac{u_1^2+u_2^2-2\rho u_1 u_2}{2(1-\rho^2)}}du_1du_2,$$

$$\Lambda^3_{x_1,x_2,w_1,w_2} = e^{-\frac{(1-\rho^2)d_2^2}{2}}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\frac{d_1}{\sqrt{1-\rho^2}}-\rho d_2} e^{-\frac{u^2}{2}}du,\ \Lambda^2_{x_1,x_2,w_1,w_2} = e^{-\frac{d_1^2}{2(1-\rho^2)}+\frac{\rho d_1 d_2}{\sqrt{1-\rho^2}}-\frac{d_2^2}{2}},$$

where d_1 and d_2 are functions of x_1,x_2,w_1 and w_2

$$d_1(x_1,w_1) = \frac{\ln\left(\frac{S_0^1}{x_1}\right)+r_1 s+\frac{\sigma_1 s w_1}{t}-\frac{s(3t-2s)\sigma_1^2}{2t}-\frac{s(t-s)\rho\sigma_1\sigma_2}{t}}{\sigma_1\sqrt{\frac{s(t-s)}{t}}},$$

$$d_2(x_2,w_1,w_2) = \frac{\ln\left(\frac{S_0^2}{x_2}\right)+r_2 s+\frac{\sigma_2\rho s w_1}{t}+\frac{\sigma_2\sqrt{1-\rho^2}sw_2}{t}-\frac{s(3t-2s)\sigma_2^2}{2t}-\frac{s(t-s)\rho\sigma_1\sigma_2}{t}}{\sqrt{1-\rho^2}\sigma_2\sqrt{\frac{s(t-s)}{t}}}.$$

Theorem 6. *For $d = 1$, we suppose that S_t has the dynamics (29), $\sigma(u) = \sqrt{1-\rho^2}\sqrt{\nu_u}$ with $|\rho| < 1$ and $\kappa > 0$, $2\kappa\theta > \eta^2$. By conditioning, the expression of the function h defined in (41) is given by*

$$h(x,w) = \frac{F\left(\int_0^t \nu_u du, \int_0^s \nu_u du\right)}{S_0\sqrt{2\pi(1-\rho^2)}} \exp\left(\begin{array}{c} -d_x^2(w) - rs - \sqrt{1-\rho^2}\dfrac{\int_0^s \nu_u du}{\int_0^t \nu_u du} w - \rho\int_0^s \sqrt{\nu_u}d\widetilde{W}_u \\ + \left(1 - \frac{\rho^2}{2}\right)\int_0^s \nu_u du - \frac{1-\rho^2}{2}\dfrac{\left(\int_0^s \nu_u du\right)^2}{\int_0^t \nu_u du} \end{array} \right),$$

with

$$d_x(w) = \frac{\ln\left(\frac{S_0}{x}\right) + rs + \sqrt{1-\rho^2}\left(\dfrac{\int_0^s \nu_u du}{\int_0^t \nu_u du}\right) w + \rho\int_0^s \sqrt{\nu_u}d\widetilde{W}_u + \left(\rho^2 - \frac{3}{2}\right)\int_0^s \nu_u du + \dfrac{(1-\rho^2)\left(\int_0^s \nu_u du\right)^2}{\int_0^t \nu_u du}}{\sqrt{1-\rho^2}\sqrt{\left(\int_0^t \nu_u du\right)\left(\int_0^s \nu_u du\right) - \left(\int_0^s \nu_u du\right)^2}/\sqrt{\int_0^t \nu_u du}},$$

and

$$F\left(\int_0^t \nu_u du, \int_0^s \nu_u du\right) = \frac{\sqrt{\int_0^t \nu_u du}}{\sqrt{\int_0^t \nu_u du \int_0^s \nu_u du - \int_0^s \nu_u du \int_0^s \nu_u du}}.$$

5 Advanced variance reduction method

In this section, we present a less intuitive variance reduction method that is based on an appropriate relation between N and N' in (6). This method can be applied independently of conditioning detailed in the previous section.

Let us denote Q as the quotient given by

$$Q = \frac{\frac{1}{N'}\sum_{i=1}^{N'} X_i}{\frac{1}{N}\sum_{i=1}^{N} Y_i} \tag{43}$$

where $\{X_i\}_{1\le i\le N'}$ and $\{Y_i\}_{1\le i\le N}$ are respectively independent copies of the square integrable random variables X, Y. If $|E(Y_i)| \ge \varepsilon > 0$, Q converges to $E(X_i)/E(Y_i)$. In the following two theorems we will prove asymptotically that we can speed up the convergence when acting on the relation between N and N'.

We will use the notations

$$A = E(X),\ B = E(Y),\ \sigma_1^2 = Var(X),\ \sigma_2^2 = Var(Y) \text{ and } \rho = Cov(X,Y)/(\sigma_1\sigma_2), \tag{44}$$

and we consider the two cases:

case 1: $N' = \lceil \lambda_1 N \rceil$ with $\lambda_1 \in]0,1[$, then (43) becomes

$$Q = \frac{\frac{1}{N'}\sum_{i=1}^{N'} X_i}{\frac{1}{N}\left(\frac{N'}{N'}\sum_{i=1}^{N'} Y_i + \frac{N-N'}{N-N'}\sum_{i=N'+1}^{N} Y_i\right)} = g_1(A_{N'}, B_{N'}, B_{N,N'}),$$

where

$$\begin{array}{c} g_1(x,y,z) = x/(\lambda_1 y + (1-\lambda_1)z), \\ A_{N'} = \frac{1}{N'}\sum_{i=1}^{N'} X_i, \quad B_{N'} = \frac{1}{N'}\sum_{i=1}^{N'} Y_i, \quad B_{N,N'} = \frac{1}{N-N'}\sum_{i=N'+1}^{N} Y_i. \end{array} \tag{45}$$

case 2: $N = \lceil \lambda_2 N' \rceil$ with $\lambda_2 \in]0,1[$, then (43) becomes

$$Q = \frac{\frac{1}{N'}\left(\frac{N}{N}\sum_{i=1}^{N} X_i + \frac{N'-N}{N'-N}\sum_{i=N+1}^{N'} X_i\right)}{\frac{1}{N}\sum_{i=1}^{N} Y_i} = g_2(A_N, A_{N',N}, B_N),$$

where

$$\begin{array}{c} g_2(x,y,z) = (\lambda_2 x + (1-\lambda_2)y)/z, \\ A_N = \frac{1}{N}\sum_{i=1}^{N} X_i, \quad A_{N',N} = \frac{1}{N'-N}\sum_{i=N+1}^{N'} X_i, \quad B_N = \frac{1}{N}\sum_{i=1}^{N} Y_i. \end{array} \tag{46}$$

Theorem 7. *Based on the notations (44) and the variables defined in (45) and (46), if $|B| > 0$ then as $N \to \infty$ and $N' \to \infty$*

$$g_1(A_{N'}, B_{N'}, B_{N,N'}) \xrightarrow{a.s.} \frac{A}{B}, \quad \sqrt{N'}\left(g_1(A_{N'}, B_{N'}, B_{N,N'}) - \frac{A}{B}\right) \xrightarrow{law} \mathcal{N}(0, \Sigma_1(\lambda_1)),$$

$$\Sigma_1(\lambda_1) = \frac{\sigma_1^2}{B^2} + \lambda_1 \frac{4A^2\sigma_2^2}{B^4}\left(\frac{1}{4} - \frac{B\sigma_1\rho}{2A\sigma_2}\right). \tag{47}$$

and

$$g_2(A_N, A_{N',N}, B_N) \xrightarrow{a.s.} \frac{A}{B}, \quad \sqrt{N}\left(g_2(A_N, A_{N',N}, B_N) - \frac{A}{B}\right) \xrightarrow{law} \mathcal{N}(0, \Sigma_2(\lambda_2)),$$

$$\Sigma_2(\lambda_2) = \frac{A^2\sigma_2^2}{B^4} + \lambda_2 \frac{4\sigma_1^2}{B^2}\left(\frac{1}{4} - \frac{A\sigma_2\rho}{2B\sigma_1}\right). \tag{48}$$

Asymptotically (N and N' sufficiently large), Theorem 7 tells us that one should use either $\lambda_1 < 1$ or $\lambda_2 < 1$ depending on the positivity of the terms $\left(\frac{1}{4} - \frac{B\sigma_1\rho}{2A\sigma_2}\right)$ and $\left(\frac{1}{4} - \frac{A\sigma_2\rho}{2B\sigma_1}\right)$ in (47) and (48). In this paper, we will take

$$\lambda_1 = \frac{1}{2} + \frac{B\sigma_1\rho}{2A\sigma_2} \;\; if \;\; A^2\sigma_2^2 \geq B^2\sigma_1^2 \quad and \quad \lambda_2 = \frac{1}{2} + \frac{A\sigma_2\rho}{2B\sigma_1} \;\; if \;\; A^2\sigma_2^2 \leq B^2\sigma_1^2. \tag{49}$$

This choice allows $\lambda_1 \in]0,1[$ and $\lambda_2 \in]0,1[$ depending on the value of ρ. Moreover, if we replace the (49) given value of λ_1 and λ_2 in $\Sigma_1(\lambda_1)$ and $\Sigma_2(\lambda_2)$ respectively, we obtain

$$\begin{array}{l} \Sigma_1(\lambda_1) = \frac{(1-\rho^2)\sigma_1^2}{B^2} + \frac{A^2\sigma_2^2}{2B^4} - \frac{A\sigma_1\sigma_2\rho}{2B^3}, \\ \Sigma_2(\lambda_2) = \frac{(1-\rho^2)A^2\sigma_2^2}{B^4} + \frac{\sigma_1^2}{2B^2} - \frac{A\sigma_1\sigma_2\rho}{2B^3}. \end{array} \tag{50}$$

From (50), we conclude that if

$$\rho^2 \leq \frac{1}{2} \tag{51}$$

then $\Sigma_1(\lambda_1) \leq \Sigma_2(\lambda_2)$ if and only if $A^2\sigma_2^2 \geq B^2\sigma_1^2$, which means that one should use λ_1 for the simulation if $A^2\sigma_2^2 \geq B^2\sigma_1^2$ and λ_2 otherwise. We also point out that the condition (51) is numerically always satisfied in our simulations.

Proof. of Theorem 7 Since the computations are similar for the **case 2**, we give only the proof associated to the **case 1**. First, the variables $A_{N'}$, $B_{N'}$ and $B_{N,N'}$ are square integrable thanks to the fact that the variables X, Y that intervene in Q are square integrable. The almost sure convergence of $g_1(A_{N'}, B_{N'}, B_{N,N'})$ follows from the law of large numbers and from the continuity of g_1 at (A, B, B). For the same reasons, the gradient vector $\nabla g_1(A_{N'}, B_{N'}, B_{N,N'})$ converges a.s. to $\nabla g_1(A, B, B)$. Besides

$$\begin{aligned}
&\sqrt{N'}(g_1(A_{N'}, B_{N'}, B_{N,N'}) - g_1(A, B, B)) \\
&\quad = \quad \sqrt{N'}(A_{N'} - A)\tfrac{\partial g_1}{\partial x}(A, B, B) + \sqrt{N'}(B_{N'} - B)\tfrac{\partial g_1}{\partial y}(A, B, B) \\
&\quad + \quad \sqrt{\tfrac{N'}{N-N'}}\sqrt{N - N'}(B_{N,N'} - B)\tfrac{\partial g_1}{\partial z}(A, B, B) + \sqrt{N'}(A_{N'} - A)\epsilon(A_{N'} - A) \\
&\quad + \quad \sqrt{N'}(B_{N'} - B)\epsilon(B_{N'} - B) + \sqrt{N'}(B_{N,N'} - B)\epsilon(B_{N,N'} - B).
\end{aligned}$$

Using the Slutsky Theorem, with $G \sim \mathcal{N}(0, C)$ and the continuity of $(x, y) \mapsto xy$ and $(x, y) \mapsto x + y$ provide

$$\sqrt{N'}(g_1(A_{N'}, B_{N'}, B_{N,N'}) - g_1(A, B, B)) \xrightarrow{law} \begin{pmatrix} \frac{\partial g_1}{\partial x}(A, B, B) \\ \frac{\partial g_1}{\partial y}(A, B, B) \\ \sqrt{\frac{\lambda_1}{1-\lambda_1}}\frac{\partial g_1}{\partial z}(A, B, B) \end{pmatrix} \cdot G,$$

where

$$C = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & 0 \\ \sigma_1\sigma_2\rho & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_2^2 \end{pmatrix},$$

which allows us to compute $\Sigma_1(\lambda_1)$.

In order to use the values (49), we should have a "sufficiently good" approximation of σ_1, σ_2, A, B and ρ. Consequently, we can implement one of the two methods below:

M0) If $B = T_{s,t}[1](x)$ and σ_2 are explicitly known, using all the simulated paths N_{max}, we approximate the values of σ_1, A and ρ then we compute λ_1 or λ_2 given in (49) that we use to re-simulate Q.

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 or λ_2 given in (49) that we use to re-simulate Q.

Procedure **P2** (see section 4) is implemented using **M0)** in subsection 6.2, when it is implemented using **M1)** in subsections 6.3 and 6.4.

When the model allows the choice between **P1** and **P2**, the following theorem tells us which one to take.

Proposition 1. *Based on the assumption (51) and the values (50), if*
1. $A^2\sigma_2^2 \geq B^2\sigma_1^2$ *and* $\rho > -1/2$ *then* $\left(B^2\Sigma_1(\lambda_1) - \sigma_1^2\right) < 0$.
2. $8A^2\sigma_2^2 \leq 7B^2\sigma_1^2$ *then* $\left(B^2\Sigma_2(\lambda_2) - \sigma_1^2\right) < 0$.

Proposition 1 tells us that, when $T_{s,t}[1](x)$ is known explicitly, one can accelerate the convergence when using the quotient of two Monte Carlo estimators. The proof of Proposition 1 is straightforward.

6 Simulation and numerical results

In this section, we perform three sets of tests that involve the results of Theorem 4, Theorem 5 and Theorem 6. But before that, we study the parallel adaptability of MCM as well as some considerations that one should respect when using GPUs to reduce the execution time.

6.1 Parallel considerations

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 Monte Carlo for multi-core and many-core architectures using CPUs and GPUs (Graphics Processing Units) in [2] and [1]. 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 Longstaff and Schwartz algorithm [16]. The results of this study prove that we can greatly decrease the execution time and the energy consumed during the simulation. Unlike the LS method that uses a regression phase which is difficult to parallelize according to [1], the MCM is a squared[4] Monte Carlo method which is more adapted to multi-core and many-core environments than the LS method. Moreover, since using MCM without localization does not involve any parametric regression, higher dimensional problems can be dealt with more easily and the accuracy of results depends only on the number of simulated trajectories.

Let us study the parallel adaptability of MCM for parallel architectures. In Figure 2, we present the speedup of parallelizing[5] 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 simulated trajectories and it reaches a saturation state for > 9000 trajectories. For a large dimensional problem, the maximum speedup obtained is greater than the number of physical

[4] What we mean by squared 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 value) and this can be done by using the same set of trajectories as the first Monte Carlo simulation.

[5] We use OpenMP directives.

cores[6] 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[7], so the speedup is invariable according to dimensions and time steps.

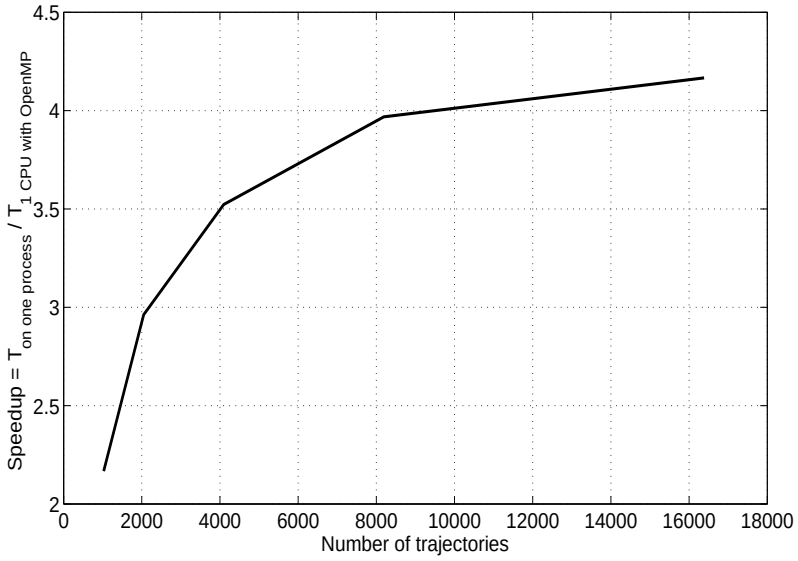

Figure 2: The speedup of using all the CPU cores according to the number of trajectories.

Regarding GPU implementation, we also use a path parallelization of simulations. In Figure 3, we present the speedup of parallelizing[8] 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 [18]. The speedups provided in Figure 3 prove, once again, the high adaptability of MCM on parallel architectures.

MCM is well suited to parallel architecture because it is completely based on Monte Carlo, unlike the Longstaff-Schwartz algorithm that performs a regression which cannot be efficiently parallelized. Indeed, for a regression that uses less than 10 polynomials, the Longstaff-Schwartz algorithm have almost the same behavior on the CPU as the one described for MCM in Figure 2. However, the many-core GPU implementation of the Longstaff-Schwartz algorithm is at most two times faster than its multi-core CPU implementation. For more details, we

[6] which is due to hyper-threading.
[7] which is the most natural procedure of parallelizing Monte Carlo.
[8] We use CUDA language.

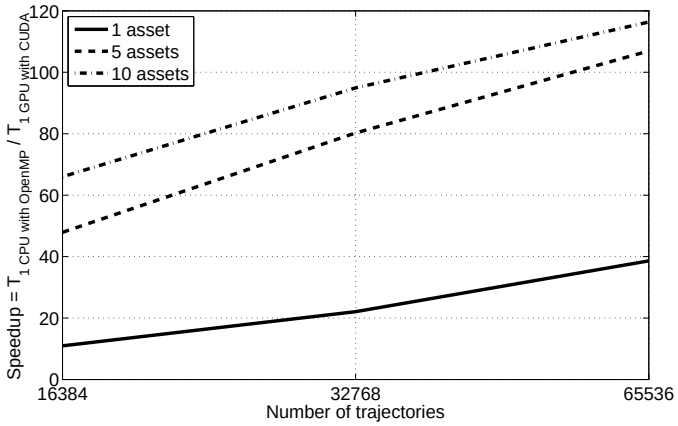

Figure 3: The speedup of using the GPU instead of the CPU cores according to the number of trajectories.

refer the reader to [1] which compares the GPU implementation of the Longstaff-Schwartz algorithm to its one-core CPU implementation.

Although MCM is based on Monte Carlo, one should, at least, respect the four points below when programming MCM on GPUs:

- Reduce the communication between CPU and GPU to its minimum, even more when implementing MCM whose trajectories are coupled, that is to say, one needs the other trajectories' data to simulate one value associated to one trajectory.
- Ensure the maximum coalescence of the data on the GPU because it affects greatly the execution time.
- Find the right compromise between the number of threads on each block and the number of blocks on the GPU when implementing MCM using less than 2^{16} trajectories.
- Saturate the GPU with as many instructions as possible thanks to the use of multi-streaming.

We refer the reader to [18] and [17] for more details on programming GPUs using CUDA.

6.2 Geometric average on independent B&S model

In this part, we simulate the prices associated to Theorem 4 and we test our simulations on a geometric average payoff that has the following expression

$$\Phi^d_{geo}(S_T) = \left(K - \prod_{i=1}^{d} (S^i_T)^{1/d} \right)_+ . \tag{52}$$

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 true values, to which we compare our simulation results, are set using the one-dimensional equivalence and a tree method [9], available in Premia [15].

In Figure 4, we compare the **P2** ($\neq$) 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 improvement of the prices tested.

In Figure 4, 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 high variance of the ten dimension contract and that one should simulate more trajectories, on the one hand, to obtain 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 value. 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.

In Table 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($\neq$)**, 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 19 and 20). First, due to the high variance, we notice that **P2(=)** is not stable in the multidimensional case and can give wrong results if the time steps > 10. This kind of bad results are also obtained for different values of the model parameters. However the **P2** method is stabilized when we implement the version $N \neq N'$ of the advanced variance reduction method detailed in section 5. Also when we use 2^{10} trajectories, **P1** and **P2($\neq$)** are almost similar. Nevertheless, with 2^{14} trajectories, **P2($\neq$)** outperforms **P1** which indicates that the conditions of Proposition 1 are fulfilled 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($\neq$)** is not very different from **P1** when we use 2^{10} trajectories. In addition, using 2^{14} trajectories, the computations of the relation between N and N' is performed on the GPU independently on each trajectory and **P2($\neq$)** is $< 5\%$ slower than **P1** for the tests that we have implemented.

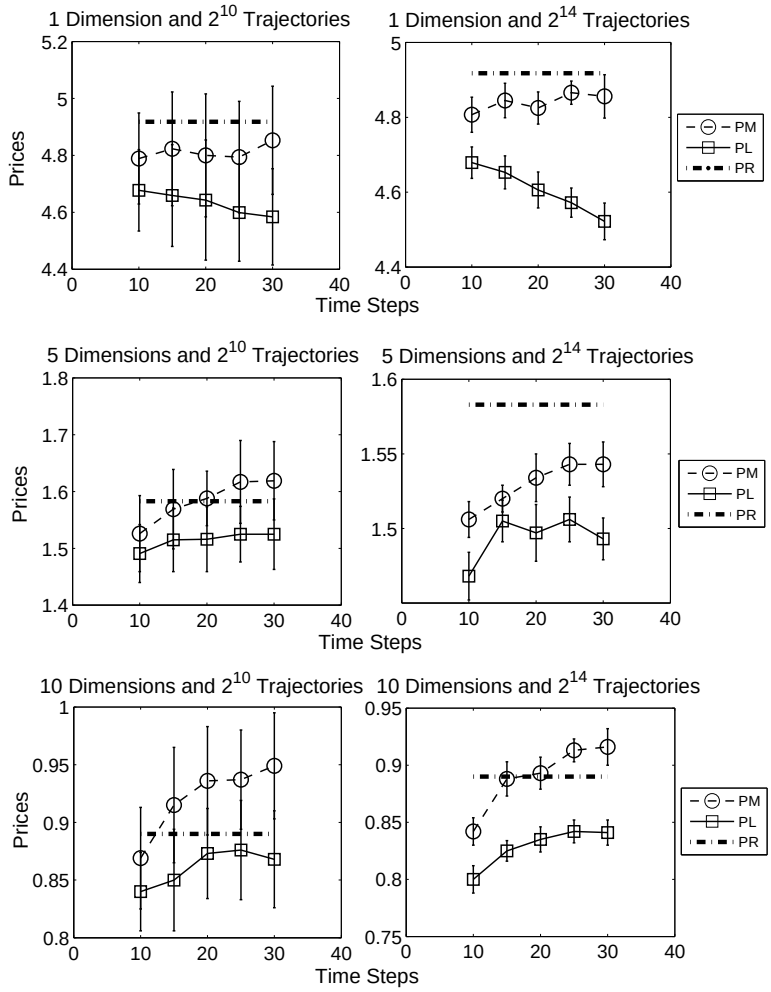

Figure 4: *MCM Vs. LS for* $\Phi^d_{geo}(S_T)$*: PR is the real price. PM and PL are the prices obtained respectively by MCM and LS represented with their standard deviations.*

6.3 Call on max and put on min on two-dimensional B&S model

In this part, we simulate the prices associated to Theorem 5 and we test our simulations on the American put on minimum and on the American call on maximum that have the following payoffs

$$\Phi_{min}(S_T) = \left(K - \min(S_T^1, S_T^2)\right)_+, \quad \Phi_{max}(S_T) = \left(\max(S_T^1, S_T^2) - K\right)_+. \quad (53)$$

Table 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 d	Time Steps	Price **P1**	Price **P2(=)**	Price **P2(≠)**	Std Deviation **P1**	Std Deviation **P2(=)**	Std Deviation **P2(≠)**
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

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, $S^i_0 = 100$.

The true values, to which we compare our simulation results, are set using the Premia implementation of a finite difference algorithm [20] in two dimensions. Besides, we use the approximation presented in [11] for the bivariate cumulative distribution in the expression of $\Lambda^1_{x_1,x_2,w_1,w_2}$ (Theorem 5). For higher dimensions, we refer the reader to [13] for the approximation of the multivariate normal cumulative distribution.

Because of the bad results obtained previously with **P2(=)**, we eliminate this method and we only consider **P2(≠)** and **P1**. In Table 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(≠)** outperforms **P1** even when we use only 2^{10}. Regarding Φ_{max}, **P1** performs better than **P2(≠)** 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(≠)** for Φ_{max}.

In Table 3, we show that our results are accurate even when $\rho \neq 0$ and when simulating only 2^{10} trajectories with **P2(≠)**.

Table 2: **P1** Vs. **P2** for Φ_{min} and Φ_{max}: Simulations for $\rho = 0$ and $\sigma_1 = \sigma_2 = 0.2$. The real values are equal to 8.262 and 21.15 respectively

Simulated Paths	The Payoff	Time Steps	Price **P1**	Price **P2**($\neq$)	Std Deviation **P1**	Std Deviation **P2**($\neq$)
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

Table 3: Φ_{min} and Φ_{max}: Simulations for $\rho \neq 0$ and $\sigma_1 = \sigma_2 = 0.2$ using 2^{10} trajectories

Real values	The Payoff	Time Steps	Price $\rho = 0.5$	Price $\rho = -0.5$	Std Deviation $\rho = 0.5$	Std Deviation $\rho = -0.5$
(+)7.23	Φ_{min}	10	7.31	9.10	0.06	0.03
	Φ_{min}	20	7.47	9.29	0.07	0.06
(−)9.05	Φ_{min}	30	7.64	9.48	0.09	0.08
(+)18.74	Φ_{max}	10	18.78	23.23	0.53	0.35
	Φ_{max}	20	19.03	23.57	0.37	0.12
(−)23.08	Φ_{max}	30	19.29	23.94	0.19	0.20

Table 4: Φ_{max}: Simulations for $\rho \neq 0$, $\sigma_1 = 0.1$ and $\sigma_2 = 0.2$ using 2^{10} trajectories

Time Steps	Price $\rho = 0.3$	Price -0.3	Price 0.7	Price -0.7	Std Deviation 0.3	Std Deviation -0.3	Std Deviation 0.7	Std Deviation -0.7
10	17.17	19.07	15.44	20.20	0.29	0.24	0.29	0.22
20	17.17	19.24	15.37	20.36	0.20	0.23	0.27	0.27
30	17.22	19.30	15.40	20.44	0.15	0.18	0.25	0.19
Real values	17.27	19.11	15.70	20.21				

When $\sigma_1 = \sigma_2$, the terms $\Lambda^1_{x_1,x_2,w_1,w_2}$ and $\Lambda^3_{x_1,x_2,w_1,w_2}$ of (42) do not intervene, thus in Table 4, we show that our results are also accurate even when $\sigma_1 \neq \sigma_2$ and when simulating only 2^{10} trajectories with **P2**($\neq$). We also considered, in Table 3, the case of highly correlated assets $|\rho| = 0.7$.

6.4 Put on Heston model

In this part, we simulate the prices associated to Theorem 6. Unlike the previous tests, in this part we do not know the real price of the American options. Thus, we will test the coherency of the results obtained with MCM to the results obtained using two different algorithms in Premia [15] (version 13). The methods to which we compare MCM are: The Longstaff-Schwartz(LS) method [16] implemented with a second order discretization scheme for the CIR process [3] and the Andersen-Broadie(AB) method [4] also implemented with a second order discretization scheme for the CIR process. We take the default parameters given in Premia for the two methods and we have tested various model parameters configurations including the one associated to Table 5: Dimension $d = 1$, maturity $T = 1$, strike $K = S_0 = 100$, $\nu_0 = 0.01$, $\kappa = 2$, $\theta = 0.01$, $\eta = 0.2$ and $r = \ln(1.1)$. Moreover, we have implemented MCM using the Milstein scheme for the CIR process and we used only 2^{10} trajectories with $\mathbf{P2(\neq)}$ because we judged it sufficient for our simulations.

According to Table 5, the results obtained with MCM are coherent to the one obtained with LS and AB. In Table 5, we only present the results for the put option, but we obtained the same kind of coherence for the call option and even for high values of ν_0, θ and η but always under the Feller conditions.

Table 5: Put option using 2^{10} trajectories and 50 time steps

Correlation(ρ)	Price(MCM)	Std Deviation(MCM)	LS	AB
-0.5	1.79	0.05	1.78	1.74
0.0	1.60	0.05	1.61	1.59
0.5	1.41	0.05	1.41	1.35

7 Conclusion and Future Work

In this article we provided, on the one hand, theoretical results that deal with the computation of the continuation value using the Malliavin calculus and how one can reduce the Monte Carlo variance when simulating this value. 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 and many-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 multidimensional models with deterministic and nonconstant triangular matrix $\sigma(t)$ as well as for the multidimensional Heston model. Moreover, we pointed out that one can judiciously 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 instantaneous simulations on the CPU can be obtained using only 2^{10} trajectories and the results got with

MCM are sufficient and better than with LS. 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 CPUs and GPUs. Indeed, for all the implemented tests, the MCM simulation of 2^{14} trajectories using the GTX 480 GPU can be performed within seconds ($< 5s$).

As future work, we plan to extend the results presented for the multidimensional Heston model to other stochastic volatility models. We will also look for a weaker and sufficient condition than the one presented in Lemma 4 for the Heston model and to extend it for the multidimensional case. Regarding the parallelization aspects, we are working on the parallelization of MCM on a CPU/GPU cluster using MPI+OpenMP+CUDA.

Appendix

Proof. of Lemma 1 The equality (15) can be easily proved. Indeed, using the chain rule

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

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

Proof. of Lemma 2 Using duality (7) we have

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

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

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.

Proof. of Theorem 4 Let us begin with $T_{s,t}[1](x)$, by independence of the coordinates we obtain

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

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

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

By computing the expectation, we obtain the requested result.

Regarding function h, we condition according to $W_t^k = w_k$ and we use the independence of coordinates

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

with

$$h_k(x_k,w_k) = E\left(H_k(S_s^k)W_{s,t}^k/S_s^k \middle| W_t^k = w_k\right).$$

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 $W_{s,t}^k$: $W_{s,t}^k \doteq \sigma_k s(t-s) + \sqrt{ts(t-s)}G$ and $S_s^k \doteq S_0^k \exp\left(\left(r_k - \frac{\sigma_k^2}{2}\right)s + \sigma_k\frac{sw_k}{t} + \sigma_k\sqrt{\frac{s(t-s)}{t}}G\right)$. Then we compute $h_k(x_k, w_k)$ which yields:

$$h_k(x_k,w_k) = \frac{e^{(\sigma_k^2 - r_k)s}}{S_0^k}\sqrt{\frac{ts(t-s)}{2\pi}}\exp\left(\frac{-s\sigma_k}{t}\left(\frac{s\sigma_k}{2}+w_k\right) - \frac{(d_{x_k}(w_k))^2}{2}\right).$$

Proof. of Theorem 5 For this model $\Gamma_{s,t} = \pi_{s,t}^{1,2}\pi_{s,t}^{2,2} - C_{1,2}$ with

$$\pi_{s,t}^{1,2} = 1 + \frac{\sqrt{1-\rho^2}W_s^1 - \rho W_s^2}{s\sigma_1\sqrt{1-\rho^2}} - \frac{\sqrt{1-\rho^2}(W_t^1 - W_s^1) - \rho(W_t^2 - W_s^2)}{(t-s)\sigma_1\sqrt{1-\rho^2}},$$

$$\pi_{s,t}^{2,2} = 1 + \frac{W_s^2}{s\sigma_2\sqrt{1-\rho^2}} - \frac{(W_t^2 - W_s^2)}{(t-s)\sigma_2\sqrt{1-\rho^2}}, \quad C_{1,2} = \frac{-t\rho}{s(t-s)(1-\rho^2)\sigma_1\sigma_2}.$$

For $k = 1, 2$, knowing $W_0^k = 0$ and $W_t^k = w_k$, when we fix s the random variable $W_s^k \doteq \frac{sw_k}{t} - \sqrt{\frac{s(t-s)}{t}}G_k$ where G_1 and G_2 are two independent with

standard normal distribution. In addition, we obtain the following equalities in distribution

$$\frac{\prod_{i=1}^{2} S_s^i}{S_0^1 S_0^2} \doteq e^{\left(\frac{(-\sigma_1^2-\sigma_2^2)}{2}+r_1+r_2\right)s+\frac{(\sigma_1+\rho\sigma_2)sw_1+\sigma_2\sqrt{1-\rho^2}sw_2}{t}-\sqrt{\frac{s(t-s)}{t}}\left((\sigma_1+\rho\sigma_2)G_1+\sigma_2\sqrt{1-\rho^2}G_2\right)}$$

$$\Gamma_{s,t} \doteq \left(1+\sqrt{\frac{t}{s(t-s)}}\left[\frac{\rho G_2}{\sigma_2\sqrt{1-\rho^2}}-\frac{G_1}{\sigma_1}\right]\right)\left(1-\sqrt{\frac{t}{s(t-s)}}\frac{G_2}{\sigma_2\sqrt{1-\rho^2}}\right)-C_{1,2},$$

which allows us to compute $h(x, w_1, w_2)$ and obtain the result of Theorem 5.

Proof. of Theorem 6 We perform the regression presented in pages 15 and 16 that provides the following equalities in distribution when we condition according to $\widetilde{\mathcal{F}}_t$ and $\int_0^t \sqrt{\nu_u} dW_u = w$

$$\frac{1}{s\sqrt{1-\rho^2}}\int_0^s \frac{dW_u}{\sqrt{\nu_u}} - \frac{1}{(t-s)\sqrt{1-\rho^2}}\int_s^t \frac{dW_u}{\sqrt{\nu_u}} \doteq \frac{-1}{s(t-s)\sqrt{1-\rho^2}}\sqrt{(t-s)^2\int_0^s \frac{du}{\nu_u}+s^2\int_s^t \frac{du}{\nu_u}}G_1,$$

$$\int_0^s \sqrt{\nu_u} dW_u \doteq \frac{\int_0^s \nu_u du}{\int_0^t \nu_u du} w - \frac{\sqrt{\left(\int_0^t \nu_u du\right)\left(\int_0^s \nu_u du\right)-\left(\int_0^s \nu_u du\right)^2}}{\sqrt{\int_0^t \nu_u du}} G_2,$$

where the Gaussian vector $(G_1, G_2) \sim \mathcal{N}(0, \Gamma)$ with

$$\Gamma = \begin{pmatrix} 1 & R \\ R & 1 \end{pmatrix}, \quad R = \frac{s(t-s)\sqrt{\int_0^t \nu_u du}}{\sqrt{\left(\int_0^t \nu_u du\right)\left(\int_0^s \nu_u du\right)-\left(\int_0^s \nu_u du\right)^2}\sqrt{(t-s)^2\int_0^s \frac{du}{\nu_u}+s^2\int_s^t \frac{du}{\nu_u}}}.$$

Thus $T_{s,t}[g](x) = E\left(E\left(g(S_t)h\left(x, \int_0^t \sqrt{\nu_u} dW_u\right)\middle|\widetilde{\mathcal{F}}_t\right)\right)$ with

$$h(x,w) = E\left(\frac{H(S_s)}{S_s}\left(1-\frac{1}{s(t-s)\sqrt{1-\rho^2}}\sqrt{(t-s)^2\int_0^s \frac{du}{\nu_u}+s^2\int_s^t \frac{du}{\nu_u}}G_1\right)\middle|\widetilde{\mathcal{F}}_t \bigvee \int_0^t \sqrt{\nu_u} dW_u = w\right)$$

and after Gaussian computations we get the requested result.

Acknowledgment: We started this work in the ANR GCPMF project, and it is supported now by CreditNext project. The authors want to thank Professor Damien Lamberton for his review of our work and Professor Vlad Bally for his valuable advice.

References

[1] 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. 24, 26

[2] 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. 24

[3] Aurélien Alfonsi. High order discretization schemes for the CIR process: Application to affine term structure and heston models. *Mathematics of Computation*, 79:209–237, 2010. 31

[4] L. Andersen and M. Broadie. A primal-dual simulation algorithm for pricing multidimensional american options. *Management Science*, 50(9):1222–1234, 2004. 31

[5] V. Bally. An elementary introduction to Malliavin calculus. *INRIA Rapport de Recherche*, 4718, 2003. 4

[6] V. Bally, L. Caramellino, and A. Zanette. Pricing American options by Monte Carlo methods using a Malliavin calculus approach. *Monte Carlo Methods and Applications*, 11:97–133, 2005. 2, 6

[7] V. Bally and G. Pagès. A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. *Bernoulli*, 9(6):1003–1049, 2003. 2

[8] M. Bossy and A. Diop. An efficient discretisation scheme for one dimensional sdes with a diffusion coefficient function of the form $|x|^\alpha$, $\alpha \in [1/2, 1)$. *Rapport de recherche INRIA*, (5396), July 2007. 16

[9] 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*, 9:1221–1250, 1996. 27

[10] E. Clément, D. Lamberton, and P. Protter. An analysis of a least squares regression algorithm for American option pricing. *Finance and Stochastics*, 17:448–471, 2002. 2

[11] Z. Drezner and G. O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35:101–107, 1989. 29

[12] D. Dufresne. The integrated square-root process. *Research paper number 90, Center for Actuarial Studies, University of Melbourne*, Novembre 2001. 14

[13] A. Genz. A comparison of methods for numerical computation of multivariate normal probabilities. *Computing Science and Statistics*, 25, 1993. 29

[14] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Applications of Mathematics. Springer, 2003. 1, 2

[15] http://www roc.inria.fr/mathfi/Premia. 27, 31

[16] F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001. 1, 2, 24, 31

[17] Nvidia. *NVIDIA CUDA C Best Practices Guide*. 2010. 26

[18] Nvidia. *NVIDIA CUDA C Programming Guide*. 2010. 25, 26

[19] J. Tsitsiklis and B. Van Roy. Regression methods for pricing complex American-style options. *IEEE Transactions on Neural Networks*, 12(4):694–703, 2001. 2

[20] S. Villeneuve and A. Zanette. Parabolic ADI methods for pricing American options on two stocks. *Mathematics of Operations Research*, 27, 2002. 29