

First order schemes in the numerical quantization method

V. Bally*, G. Pagès† and J. Printems‡

Abstract

The numerical quantization method is a grid method which relies on the approximation of the solution of a nonlinear problem by piecewise constant functions. Its purpose is to compute a large number of conditional expectations along the path of the associated diffusion process. We give here an improvement of this method by describing a first order scheme based on piecewise *linear* approximations. Main ingredients are correction terms in the transition probability weights. We emphasize the fact that in the case of optimal quantization, many of these correcting terms vanish. We think that this is a strong argument to use it. The problem of pricing and hedging American options is investigated and *a priori* estimates of the errors are proposed.

Keywords : Numerical quantization, american options, Malliavin calculus.

AMS 2000 Classification : 60G40, 65C05, 65C20, 65N50, 90A09.

1 Introduction

The numerical quantization method has been introduced in [1, 3, 4]. It is a grid method which is conceived in order to solve non linear problems in large dimension. Since the problems which we have in mind have a *P.D.E.* formulation, analytical methods like finite differences or finite elements are candidates in order to solve such problems but it is well known that the implementation of this type of methods is rather difficult in dimension larger than three. So one would like to use some probabilistic methods of Monte Carlo type (which have the advantage of being dimension free). But this may not be done directly for non linear problems because the resolution of such problems suppose the computation of a large number of conditional expectations and not only of a single expectation. The numerical quantization method is in-between the analytical approach and the Monte Carlo method. One uses some grids and some weights (like in the finite element method) but the weights are computed using a Monte Carlo method. Although the error depends on the dimension as in the analytical methods, the advantage of using Monte Carlo is that one

*Université du Mans et projet MathFi INRIA, bally@ccr.jussieu.fr

†CNRS UMR 7599, Université Paris 6, gpa@ccr.jussieu.fr

‡corresponding author : Centre de Mathématique & CNRS UMR 8050, Université de Paris 12, 61, avenue du Général de Gaulle, 94010 CRÉTEIL, FRANCE. printems@univ-paris12.fr

may implement such algorithms in dimension larger than 3 (typically up to 10 - beyond, the number of points needed in the grids becomes huge).

The aim of this paper is to give a more efficient version of this algorithm. Roughly speaking, in [1, 2, 3, 4] we have studied approximation schemes of order zero and now we give approximation schemes of order one. Basically the schemes of order zero produce piecewise constant approximations of the functions at hand and consequently use the information at one point only: the center of the cell on which the approximation of the function is constant. The schemes of order one use linear interpolation and so put to work several points: the center of the cell but also the centers of its neighbors. For example the basic finite element method is an algorithm of order one because it is based on linear interpolations. If one uses polynomial interpolations then one obtains schemes of higher order (but of course the algorithm becomes much more complex). In our frame we use the Malliavin integration by parts formula in order to compute some correctors which produce piecewise linear interpolations. The attractive thing in our approach is that although the scheme becomes more complicated as we pass from a 0^{th} -order scheme to a 1^{st} -order scheme, the complexity of the algorithm remains of the same order and the correctors which come in are of the same nature. Consequently they may be computed rather simply by the Monte Carlo method with the sample used to compute the weights coming on in the original 0^{th} -order method.

As emphasized in Section 3, there are two types of projection errors coming in our algorithm, say *top* and *bottom* errors. So, a priori we need two types of correctors concerning each of these errors. But it turns out that, if we use optimal grids (in the quantization sense) the bottom correctors naturally vanish. This is an enlightening fact concerning optimal quantization and a strong argument to use it.

The numerical quantization represents a quite general approach to non-linear problems because its main purpose is to compute a large number of conditional expectations along the path of a diffusion process (see Section 3.3). But our specific initial motivation comes from pricing American options, which is an optimal stopping problem and so a typical non-linear problem. In this paper we also focus on this problem in order to illustrate the method.

Finally, we mention that the first order correctors are closely related to the hedging strategy, so we can produce as well some proxy of this hedging strategy as a by-product of this first order scheme (see [2] for an extensive discussion of hedging by quantization).

2 The basic algorithm for pricing American options

2.1 The problem

We consider a market model containing a risk-less asset S_t^0 and a d -dimensional risky asset $S_t \in \mathbb{R}^d$, $0 \leq t \leq T$, whose dynamics read

$$\begin{aligned} dS_t^0 &= S_t^0 r dt, \quad S_0^0 = 1, \\ dS_t &= \text{Diag}(S_t)(r \mathbf{1} dt + \bar{\sigma}(t, S_t) dB_t), \quad S_0 = 1. \end{aligned}$$

Here r denotes the interest rate, $\bar{\sigma}$ is the volatility function and B_t is a standard d -dimensional Brownian Motion on some probability space (Ω, \mathcal{F}, P) and $(\mathcal{F}_t)_{0 \leq t \leq T}$ its standard filtration. T is a fixed time (maturity). In order to avoid some rather complicated formulae in our computations it is convenient to work with $X_t := \log S_t$ instead of S_t itself. It is easy to see that the dynamics of X obey

$$dX_t = \sigma(t, X_t) dB_t + b(t, X_t) dt, \quad X_0 = x := \log s_0$$

with $\sigma(t, x) = \bar{\sigma}(t, e^x)$, $b(t, x) = r - \frac{1}{2} \text{Tr}(\sigma \sigma^*)(t, e^x)$.

Moreover, we consider a payoff function $h : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ and we want to price an American option with payoff h . The price at time $t \in [0, T]$ is given by

$$Y_t = \text{esssup}_{\tau \in \mathcal{T}_{t,T}} \mathbb{E}(h(\tau, X_\tau) | \mathcal{F}_t)$$

where $\mathcal{T}_{t,T}$ denotes the set of all $[t, T]$ -valued stopping times. This is the Snell envelope of the semi-martingale $h(t, X_t)$ (if h is sufficiently smooth). It is well known that no closed formula holds for Y so we will consider a discrete approximation. To process, we consider the Euler scheme of step $\frac{T}{n}$,

$$\bar{X}_{t_{k+1}} := \bar{X}_{t_k} + b(t_k, \bar{X}_{t_k}) \frac{T}{n} + \sigma(t_k, \bar{X}_{t_k}) \Delta_{k+1}, \quad \bar{X}_0 := x,$$

where $t_k := \frac{k}{n}T$ and $\Delta_{k+1} := B_{t_{k+1}} - B_{t_k}$. In order to simplify the notation we put $X_k := \bar{X}_{t_k}$, $\sigma_k(x) := \sigma(t_k, x)$ and $b_k(x) := b(t_k, x)$ so that we are concerned with the $(\mathcal{F}_{t_k})_{0 \leq k \leq n}$ -Markov chain $(X_k)_{0 \leq k \leq n}$ recursively defined by

$$X_{k+1} = X_k + b_k(X_k) \frac{T}{n} + \sigma_k(X_k) \Delta_{k+1}, \quad X_0 = x.$$

Then the discrete version of Y is defined by $\bar{Y}_{t_k} = \text{esssup}_{\bar{\tau} \in \bar{\mathcal{T}}_{t_k, T}} \mathbb{E}(h_k(X_{\bar{\tau}}) | \mathcal{F}_{t_k})$ where $h_k(x) := h(t_k, x)$ and $\bar{\mathcal{T}}_{t_k, T}$ denotes the set of all the $(\mathcal{F}_{t_i})_{0 \leq i \leq n}$ discrete stopping times which take values in $\{k, \dots, n\}$. We will work under two different sets of hypothesis:

$$(H_1) \left\{ \begin{array}{l} (i) \quad b, \sigma \text{ are continuously differentiable with bounded derivatives and } h \text{ is} \\ \quad \text{Lipschitz continuous functions.} \\ (ii) \quad \sigma \sigma^* \geq c I_d \text{ where } c > 0 \text{ and } I_d \text{ is the identity matrix.} \end{array} \right.$$

or

$$(H_2) \left\{ \begin{array}{l} (i) \quad b, \sigma \text{ are continuously differentiable with bounded derivatives and } h \text{ is} \\ \quad \text{is Lipschitz continuous in } t \text{ and semi-convex in } x. \\ (ii) \quad \sigma\sigma^* \geq cI_d, c > 0. \end{array} \right.$$

A precise definition of semi-convex functions is given in [3] (in particular Lipschitz continuous convex functions are semi-convex as well as twice differentiable functions with bounded derivatives). It is proved in [4] that under these hypothesis

$$\left(\mathbb{E} \max_{k \leq n} |Y_{t_k} - \bar{Y}_{t_k}|^2 \right)^{1/2} \leq \frac{C}{n^\alpha} \quad (2.1)$$

with $\alpha = \frac{1}{2}$ under (H_1) and $\alpha = 1$ under (H_2) (provided one uses the true “sampled” diffusion $(X_{t_k})_{0 \leq k \leq n}$ as the Markov chain to be quantized instead of its Euler scheme).

Now we compute \bar{Y}_{t_k} using the dynamical programming principle:

$$\begin{aligned} \bar{Y}_{t_n} &= h_n(X_n) \\ \bar{Y}_{t_k} &= \max(h_k(X_k), \mathbb{E}(\bar{Y}_{t_{k+1}} | \mathcal{F}_{t_k})). \end{aligned}$$

The analytical counterpart of this scheme is obtained in the following way. One constructs recursively the functions u_k by

$$\begin{aligned} u_n(x) &= h_n(x) \\ u_k(x) &= \max(h_k(x), \mathbb{E}(u_{k+1}(X_{k+1}) | X_k = x)). \end{aligned}$$

Then $\bar{Y}_{t_k} = u_k(X_k)$ and consequently, up to the approximation of Y by \bar{Y} , the price at time zero is given by $u_0(x) = u_0(\log s_0)$.

2.2 The basic algorithm

We want to produce an algorithm in order to compute u_k . The difficult point will be of course to design an efficient method to compute $\mathbb{E}(u_{k+1}(X_{k+1}) | X_k = x)$, $k = 0, \dots, n$. It is clear that we cannot do it for every point $x \in \mathbb{R}^d$, so we will settle some space grids $\Gamma_k = \{x_k^1, \dots, x_k^{N_k}\} \subset \mathbb{R}^d$, $k = 0, \dots, n$, one for each epoch $t_k = \frac{kT}{n}$. The way we choose the size N_k of the grid and the location of the points x_k^i of the grid Γ_k play a crucial part and the numerical efficiency of the algorithm heavily depends on this choice. But these problems have already been extensively discussed in [3] and so we leave them out here. So, in this paper the grids $\Gamma_k, k = 0, \dots, n$ are some exogenously designed objects. Moreover we define the Voronoi tessell of x_k^i by

$$C_k^i := \{u \in \mathbb{R}^d / |u - x_k^i| \leq \inf_{0 \leq j \leq N_k} |u - x_k^j|\}$$

and we denote by Π_k the projection on the grid Γ_k *i.e.* $\Pi_k(u) := \sum_{i=1}^{N_k} x_k^i \mathbf{1}_{C_k^i}(u)$. Note that $C_k^i, i = 1, \dots, n$, is not a true partition of \mathbb{R}^d because the different tessels have

boundary hyperplanes in common. But this is just a formal disagreement because the laws of the random variables we work with are absolutely continuous and so give no mass to hyperplanes. The basic idea is to approximate

$$\mathbb{E}(u_{k+1}(X_{k+1})|X_k = x_k^i) \approx \mathbb{E}(u_{k+1}(\Pi_{k+1}(X_{k+1})|\Pi_k(X_k) = x_k^i). \quad (2.2)$$

Note that

$$\begin{aligned} \mathbb{E}(u_{k+1}(\Pi_{k+1}(X_{k+1})) | \Pi_k(X_k) = x_k^i) &= \sum_{j=1}^{N_{k+1}} u_{k+1}(x_{k+1}^j) \mathbb{E}(\mathbf{1}_{C_{k+1}^j}(X_{k+1})|\mathbf{1}_{C_k^i}(X_k)) \\ &= \sum_{j=1}^{N_{k+1}} u_{k+1}(x_{k+1}^j) \pi_k^{ij} \\ \text{with } \pi_k^{ij} &:= \frac{\mathbb{P}(X_{k+1} \in C_{k+1}^j, X_k \in C_k^i)}{\mathbb{P}(X_k \in C_k^i)}. \end{aligned}$$

The π_k^{ij} 's are the weights in the algorithm and we compute them using a Monte Carlo simulation. The important point here is that we may compute all π_k^{ij} , $i = 1, \dots, N_k$, $j = 1, \dots, N_{k+1}$, $k = 0, \dots, n$ using the same sample $(X_k^1)_{0 \leq k \leq n}, \dots, (X_k^M)_{0 \leq k \leq n}$ of the chain $(X_k)_{0 \leq k \leq n}$ (see [3]). In fact

$$\pi_k^{ij} \sim \widehat{\pi}_k^{ij} =: \frac{\sum_{m=1}^M \mathbf{1}_{C_{k+1}^j}(X_{k+1}^m) \mathbf{1}_{C_k^i}(X_k^m)}{\sum_{m=1}^M \mathbf{1}_{C_k^i}(X_k^m)}. \quad (2.3)$$

So we avoid using different Monte Carlo procedures in order to compute the conditional expectation at each point, which would be extremely expensive. In this sense our algorithm may be seen as a *compressed* Monte Carlo Method. Now the algorithm reads

$$\begin{aligned} \widehat{u}_n(x_n^i) &= h_n(x_n^i), & i = 1, \dots, N_n, \\ \widehat{u}_k(x_k^i) &= \max \left(h_k(x_k^i), \sum_{j=1}^{N_{k+1}} \pi_k^{ij} \widehat{u}_{k+1}(x_{k+1}^j) \right), & k = 0, \dots, n. \end{aligned} \quad (2.4)$$

Of course in true applications we do not know π_k^{ij} , so we use $\widehat{\pi}_k^{ij}$. This introduces one further error – the statistical error – which is not discussed here (see [4]).

This is our basic algorithm. It is an algorithm of order zero because we replace $u_{k+1}(x)$ by $u_{k+1}(\Pi_{k+1}(x)) = \sum_{j=1}^{N_{k+1}} x_{k+1}^j \mathbf{1}_{C_{k+1}^j}(x)$ and X_k (with respect to which one takes conditional expectation) by $\Pi_k(X_k) = \sum_{i=1}^{N_k} x_k^i \mathbf{1}_{C_k^i}(X_k)$. So we work with piecewise constant functions.

2.3 Optimal grids and error estimates

We give now some error evaluations which are obtained in [4]. First, one proves that

$$\left(\mathbb{E} \max_{k \leq n} |Y_{t_k} - \widehat{u}_k(\Pi_k(X_k))|^2 \right)^{1/2} \leq \frac{C}{n^\alpha} + \sum_{k=0}^n C_k \left(\mathbb{E} |X_k - \Pi_k(X_k)|^2 \right)^{1/2} \quad (2.5)$$

with $\alpha = \frac{1}{2}$ under (H_1) and $\alpha = 1$ under (H_2) .

The grids we use are optimal in the following sense. One defines the distortion of a grid $\Gamma := \{x^1, \dots, x^{N_k}\}$ (with obvious notations) as

$$(D_{X_k}(\Gamma))^2 := \mathbb{E} |X_k - \Pi_\Gamma(X_k)|^2 = \sum_{i=1}^{N_k} \mathbb{E} (|X_k - x^i|^2 \mathbf{1}_{C_\Gamma^i}(X_k)). \quad (2.6)$$

A grid Γ_k is *optimal* if

$$D_{X_k}(\Gamma_k) = \inf_{\Gamma, |\Gamma| \leq N_k} D_{X_k}(\Gamma).$$

A basic result from the quantization theory (the Bucklew & Wise Theorem - see [3, 5, 6] for the precise result) asserts that, if the grid is optimal, then there exists a real constant C_2 such that

$$(D_{X_k}(\Gamma_k))^{1/2} = (\mathbb{E} (|X_k - \Pi_{\Gamma_k}(X_k)|^2))^{1/2} \leq \frac{C_2}{N_k^{1/d}}. \quad (2.7)$$

Plugging (2.7) in (2.5) and using the structure of the constants C_k in (2.5) make it possible to tune the N_k 's in an optimal way: this is achieved in [4]. Since in this paper we are simply interested in the asymptotic order, we leave out this slightly more sophisticated analysis and express the above error in terms of $N := \max_{0 \leq k \leq n} N_k$.

Proposition 1 *Assume that (2.7) holds true and $N_k \leq N$, $k = 1, \dots, n$. Then*

$$\left(\mathbb{E} \max_{k \leq n} |Y_{t_k} - \widehat{u}_k(\Pi_k(X_k))|^2 \right)^{1/2} \leq C \left(\frac{1}{n^\alpha} + \frac{n}{N^{1/d}} \right) \quad (2.8)$$

with $\alpha = \frac{1}{2}$ under (H_1) and $\alpha = 1$ under (H_2) .

Let us take one step beyond into the numerical properties of optimal grids. Since $\Gamma_k := \{x_k^1, \dots, x_k^{N_k}\}$ achieves the minimum, formal derivation in (2.6) (see [6] for the complete argument) yields

$$\frac{\partial}{\partial x_k^i} (D_{X_k}(\Gamma_k))^2 = 2 \mathbb{E} ((X_k - x_k^i) \mathbf{1}_{C_k^i}(X_k)) = 0. \quad (2.9)$$

We will show in the next section that optimal grids produce an error of order $N^{-2/d}$ instead of $N^{-1/d}$ and the relation (2.9) represents the key argument: it says that, if the grid is optimal, then the terms of order one in a certain Taylor expansion of order two vanish.

3 Correctors of order one

In the approximation presented in (2.2) there are two different projection errors corresponding to Π_k (“bottom”) and Π_{k+1} (“top”). The aim of this section is to produce some correctors which reduce these errors. In order to enlighten the notations we put $\widehat{X}_k = \Pi_k(X_k)$.

3.1 The basic integration by parts formula

In our very elementary setting the Malliavin integration by parts formula reads as follows. Given two differentiable functions $f, g : \mathbb{R}^d \rightarrow \mathbb{R}$, and any real constant C ,

$$\mathbb{E} \left(\frac{\partial f}{\partial x_i}(\Delta_{k+1})g(\Delta_{k+1}) \right) = -\mathbb{E} \left((f(\Delta_{k+1}) + C) \left(\frac{\partial g}{\partial x_i}(\Delta_{k+1}) - \frac{n}{T}\Delta_{k+1}^i g(\Delta_{k+1}) \right) \right). \quad (3.1)$$

The proof is obtained by a usual integration by parts (with respect to the Gaussian distribution). It represents the starting point of the Malliavin calculus - which goes far away - but we stop here.

It seems natural to take $C = 0$ in the above formula because anyway $\nabla(f + C) = \nabla f$. But in our frame f appears as an a priori given function and the fact that we have the freedom of choosing any C is crucial for simulation. The practical way of using this formula is to employ the Monte Carlo method for computing the expectation in the right hand side in order to obtain the expectation in the left hand side. So we would like to simulate the expectation of some variable with a small variation and consequently we would choose $C = -\mathbb{E}(f(\Delta_{k+1}))$ (if we know it) for example.

Now let $U : \mathbb{R}^d \rightarrow \mathbb{R}$ be a measurable function with polynomial growth (so that $U(X_{k+1})$ is integrable). We define

$$P_k U(x) := \mathbb{E}(U(X_{k+1})|X_k = x)$$

which represents the transition kernel of the Euler Scheme. The problem is to compute the derivatives of $P_k U$. One defines

$$\theta_k(x, y) := x + b_k(x)\frac{T}{n} + \sigma_k(x)y \quad \text{so that} \quad X_{k+1} = \theta_k(X_k, \Delta_{k+1})$$

and $P_k U(x) = \mathbb{E}(U(\theta_k(x, \Delta_{k+1})))$. Moreover, since σ_k is invertible, one may define

$$\begin{aligned} \lambda_k(x, y) &:= \sigma_k^{-1}(x) \times \nabla_x \theta_k(x, y) \quad \text{and} \\ \rho_k^\ell(x, y) &:= -\sum_{\ell'=1}^d \left(\frac{\partial \lambda_k^{\ell'\ell}}{\partial y^{\ell'}}(x, y) - \frac{n}{T} y^{\ell'} \lambda_k^{\ell'\ell}(x, y) \right), \quad \ell = 1, \dots, d. \end{aligned} \quad (3.2)$$

Lemma 1 *The partial derivatives of $P_k U$ are given by*

$$\frac{\partial P_k U}{\partial x_\ell}(x) = \mathbb{E}((U(\theta_k(x, \Delta_{k+1})) - C(x))\rho_k^\ell(x, \Delta_{k+1})), \quad \ell = 1, \dots, d, \quad (3.3)$$

where C is any real function.

Proof. One may assume w.l.g. that U is smooth. Note that $\nabla_y(U \circ \theta_k) = (\nabla U) \circ \theta_k \times \nabla_y \theta_k = (\nabla U) \circ \theta_k \times \sigma_k$ and so $(\nabla U) \circ \theta_k = \nabla_y(U \circ \theta_k) \times \sigma_k^{-1}$. It follows that

$$\nabla P_k U(x) = \mathbb{E}((\nabla U)(\theta_k(x, \Delta_{k+1}))\nabla_x \theta_k(x, \Delta_{k+1})) = \mathbb{E}(\nabla_y(U \circ \theta_k)(x, \Delta_{k+1})\lambda_k(x, \Delta_{k+1}))$$

and now (3.3) follows from (3.1) with $C = C(x)$. \square

Remark. Let us emphasize the simple but important example of constant volatility σ_k and constant drift coefficient b_k (the *log-normal* model corresponding to the classical Black-Scholes model). Then $\theta_k(x, y) = x + b_k \frac{T}{n} + \sigma_k y$, $\lambda_k(x, y) = \sigma_k^{-1}$ and so $\rho_k^\ell(x, y) = \frac{n}{T}(y\sigma_k^{-1})_\ell$.

Finally we give some *a priori* estimates used to evaluate errors.

Proposition 2 *Assume that U is $[U]_1$ -Lipschitz continuous. Then,*

$$\left\| \frac{\partial P_k U}{\partial x_\ell} \right\|_\infty \leq C[U]_1 \quad \text{and} \quad \left\| \frac{\partial^2 P_k U}{\partial x_\ell \partial x_{\ell'}} \right\|_\infty \leq C[U]_1 \sqrt{n}, \quad \ell, \ell' \in \{1, \dots, d\}. \quad (3.4)$$

Proof. The first inequality in (3.4) is obtained by direct calculation and the second one is obtained using integration by parts once. integration by parts once \square

3.2 The Π_k projection error (or *bottom* error)

We consider the same measurable function U as in the previous subsection and we want to approximate $P_k U(x_k^i)$. So, we define

$$\phi_U^k(x_k^i) := \frac{\mathbb{E}(U(X_{k+1}) \mathbf{1}_{C_k^i}(X_k))}{\mathbb{P}(X_k \in C_k^i)}.$$

Note that, if $U(x) = \mathbf{1}_{C_{k+1}^j}(x)$ then $\phi_U^k(x_k^i) = \pi_k^{ij}$, *i.e.* the standard weight we use in our algorithm. We write

$$\mathbb{E}(U(X_{k+1}) \mathbf{1}_{C_k^i}(X_k)) = \mathbb{E}(P_k U(X_k) \mathbf{1}_{C_k^i}(X_k)) = P_k U(x_k^i) \mathbb{P}(X_k \in C_k^i) + \varepsilon_k^i$$

with

$$\sum_{i=1}^{N_k} |\varepsilon_k^i| \leq [P_k U]_1 \mathbb{E} |X_k - \Pi_k(X_k)| \leq \frac{C[P_k U]_1}{N^{1/d}}$$

the last inequality being a consequence of (2.7). In particular the above relation gives $P_k U(x_k^i) = \phi_U^k(x_k^i) - \varepsilon_k^i / \mathbb{P}(X_k \in C_k^i)$ and so

$$\begin{aligned} \mathbb{E} \left| P_k U(\widehat{X}_k) - \phi_k^U(\widehat{X}_k) \right| &= \sum_{i=1}^{N_k} \mathbb{E} \left(\left| P_k U(\widehat{X}_k) - \phi_k^U(\widehat{X}_k) \right| \mathbf{1}_{\{X_k \in C_k^i\}} \right) \\ &= \sum_{i=1}^{N_k} |P_k U(x_k^i) - \phi_k^U(x_k^i)| \mathbb{P}(X_k \in C_k^i) \leq \frac{C[P_k U]_1}{N^{1/d}}. \end{aligned}$$

The aim of this section is to prove that, as a consequence of the optimality of the grid, the above error is of order $\frac{\sqrt{n}}{N^{2/d}}$ instead of $\frac{1}{N^{1/d}}$.

Proposition 3 *Assume that U is Lipschitz continuous and that the grid Γ_k is optimal, so that (2.9) holds true. Then*

$$\mathbb{E} \left| P_k U(\widehat{X}_k) - \phi_k^U(\widehat{X}_k) \right| \leq \frac{C[U]_1 \sqrt{n}}{N^{2/d}}. \quad (3.5)$$

Proof. We use the Markov property and a Taylor expansion of $P_k U$:

$$\begin{aligned} \mathbb{E}(U(X_{k+1})\mathbf{1}_{C_k^i}(X_k)) &= \mathbb{E}(P_k U(X_k)\mathbf{1}_{C_k^i}(X_k)) = P_k U(x_k^i)\mathbb{P}(X_k \in C_k^i) \\ &\quad + \sum_{\ell=1}^d \frac{\partial P_k U}{\partial x_\ell}(x_k^i)\mathbb{E}((X_k - x_k^i)_\ell \mathbf{1}_{C_k^i}(X_k)) + R_k^i \end{aligned}$$

with (see (3.4))
$$\sum_{i=1}^{N_k} |R_k^i| \leq \|\partial^2 P_k U\|_\infty \mathbb{E}|X_k - \Pi_k(X_k)|^2 \leq \frac{C[U]_1 \sqrt{n}}{N^{2/d}}.$$

Since the grid is optimal, $\mathbb{E}((X_k - x_k^i)^\ell \mathbf{1}_{C_k^i}(X_k)) = 0$ and the proof is complete. \square

3.3 The Π_{k+1} projection error (or *top* error)

The aim of this section is to compute $\phi_k^U(x_k^i)$. We cannot solve our problem for a general measurable function U so we need U to have the special form

$$U(x) = P_{k+1}V(x) := \mathbb{E}(V(X_{k+2})|X_{k+1} = x).$$

Moreover, in order to obtain reasonable error estimates, we assume that $V = P_{k+2}W$ for some bounded measurable function W . This may be seen as a regularity property for V , (in particular V is Lipschitz continuous). Assume temporarily that W is Lipschitz continuous as well. Then, as a consequence of (3.4),

$$\left\| \frac{\partial^2 U}{\partial x_\ell \partial x_{\ell'}} \right\|_\infty \leq C[V]_1 \sqrt{n} \quad \text{and} \quad \left\| \frac{\partial V}{\partial x_\ell} \right\|_\infty \leq C[W]_1, \quad \ell, \ell' \in \{1, \dots, d\}. \quad (3.6)$$

We define the new weights by setting

$$\pi_k^{\ell, ijr} := \frac{\mathbb{E}(\rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2})(X_{k+1} - x_{k+1}^j)_\ell \mathbf{1}_{C_k^i \times C_{k+1}^j \times C_{k+2}^r}(X_k, X_{k+1}, X_{k+2}))}{\mathbb{P}(X_k \in C_k^i)}, \quad (3.7)$$

$$\ell = 1, \dots, d, \quad i = 1, \dots, N_k, \quad j = 1, \dots, N_{k+1}, \quad r = 1, \dots, N_{k+2},$$

and
$$P_{V,k}U(x_k^i) := \sum_{j=1}^{N_{k+1}} \pi_k^{ij} U(x_{k+1}^j) + \sum_{\ell=1}^d \sum_{j=1}^{N_{k+1}} \sum_{r=1}^{N_{k+2}} \pi_k^{\ell, ijr} (V(x_{k+2}^r) - C_{V,\ell}^j) \quad (3.8)$$

where $C_{V,\ell}^j$ are arbitrary real constants to be settled in order to reduce the variance.

Lemma 2 *Suppose that $U = P_{k+1}V$ with $V = P_{k+1}W$ for some Lipschitz continuous function W . Then*

$$\mathbb{E} \left| P_k U(\widehat{X}_k) - P_{V,k} U(\widehat{X}_k) \right| \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}} \quad (3.9)$$

where C depends on the diffusion coefficients.

Proof. The idea is similar to that in the previous section: we localize on C_{k+1}^j and we use a Taylor expansion

$$\phi_k^U(x_k^i) \times \mathbb{P}(X_k \in C_k^i) = \mathbb{E}(U(X_{k+1})\mathbf{1}_{C_k^i}(X_k)) = \sum_{j=1}^{N_{k+1}} \mathbb{E}(U(X_{k+1})\mathbf{1}_{C_{k+1}^j}(X_{k+1})\mathbf{1}_{C_k^i}(X_k))$$

$$\begin{aligned}
&= \sum_{j=1}^{N_{k+1}} U(x_{k+1}^j) \mathbb{E}(\mathbf{1}_{C_{k+1}^j}(X_{k+1}) \mathbf{1}_{C_k^i}(X_k)) + \\
&\quad + \sum_{j=1}^{N_{k+1}} \sum_{\ell=1}^d \mathbb{E} \left(\frac{\partial U}{\partial x_\ell}(X_{k+1})(X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_{k+1}^j}(X_{k+1}) \mathbf{1}_{C_k^i}(X_k) \right) + Q_k^i
\end{aligned}$$

$$\begin{aligned}
\text{with } Q_k^i &:= \sum_{j=1}^{N_{k+1}} \sum_{\ell, \ell'=1}^d \mathbb{E} \left(\frac{\partial^2 U}{\partial x_\ell \partial x_{\ell'}}(X_{k+1}^j)(X_{k+1} - x_{k+1}^j) \ell (X_{k+1} - x_{k+1}^j) \ell' \mathbf{1}_{C_{k+1}^j}(X_{k+1}) \mathbf{1}_{C_k^i}(X_k) \right) \\
&\quad + \sum_{j=1}^{N_{k+1}} \sum_{\ell=1}^d \mathbb{E} \left(\left(\frac{\partial U}{\partial x_\ell}(x_{k+1}^j) - \frac{\partial U}{\partial x_\ell}(X_{k+1}) \right) (X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_{k+1}^j}(X_{k+1}) \mathbf{1}_{C_k^i}(X_k) \right)
\end{aligned}$$

We use (3.6) and we obtain

$$\sum_{i=1}^{N_k} \mathbb{E} |Q_k^i| \leq C \|\partial^2 U\|_\infty \mathbb{E} |X_{k+1} - \widehat{X}_{k+1}|^2 \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}}. \quad (3.10)$$

In order to compute $\frac{\partial U}{\partial x_\ell}(X_{k+1})$ we use (3.3) with $x = X_{k+1}$ and we obtain

$$\begin{aligned}
\frac{\partial U}{\partial x_\ell}(X_{k+1}) &= \mathbb{E} \left((V(\theta_{k+1}(X_{k+1}, \Delta_{k+2})) - C_{V,\ell}^j \rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2}) | X_{k+1}) \right) \\
&= \mathbb{E} \left((V(X_{k+2}) - C_{V,\ell}^j \rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2}) | X_{k+1}) \right).
\end{aligned}$$

Moreover, using the Markov property first and then localization for X_{k+2} yield

$$\begin{aligned}
&\mathbb{E} \left(\frac{\partial U}{\partial x_\ell}(X_{k+1})(X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_k^i \times C_{k+1}^j}(X_k, X_{k+1}) \right) \\
&= \mathbb{E} \left((V(X_{k+2}) - C_{V,\ell}^j \rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2}))(X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_k^i \times C_{k+1}^j}(X_k, X_{k+1}) \right) \\
&= \sum_{r=1}^{N_{k+2}} \mathbb{E} \left((V(X_{k+2}) - C_{V,\ell}^j \rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2}))(X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_k^i \times C_{k+1}^j \times C_{k+2}^r}(X_k, X_{k+1}, X_{k+2}) \right) \\
&= \sum_{r=1}^{N_{k+2}} (V(x_{k+2}^r) - C_{V,\ell}^j) \mathbb{E}(\rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2}))(X_{k+1} - x_{k+1}^j) \ell \mathbf{1}_{C_k^i \times C_{k+1}^j \times C_{k+2}^r}(X_k, X_{k+1}, X_{k+2}) \\
&\quad + H_k^{\ell, ij}.
\end{aligned}$$

Note that both assumptions (H_1) and (H_2) imply that

$$|\rho_k^\ell(x, y)| \leq C \left(1 + \frac{n}{T} |y| (1 + |y|) \right), \quad (3.11)$$

hence $\|\rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2})\|_2 \leq C \sqrt{n/T}$ since $\sqrt{\frac{T}{n}} \Delta_{k+2}$ is a standard normal distribution.

Keeping in mind that V is Lipschitz continuous yields

$$\begin{aligned}
\sum_{i=1}^{N_k} \sum_{j=1}^{N_{k+1}} |H_k^{\ell, ij}| &\leq C[V]_1 \sqrt{n} \sum_{i=1}^{N_k} \mathbb{E} \left(|X_{k+2} - \widehat{X}_{k+2}| |X_{k+1} - \widehat{X}_{k+1}| \mathbf{1}_{C_k^i}(X_k) \right) \\
&\leq C \sqrt{n} \mathbb{E} \left(|X_{k+2} - \widehat{X}_{k+2}|^2 + |X_{k+1} - \widehat{X}_{k+1}|^2 \right) \leq \frac{C \sqrt{n}}{N^{2/d}}.
\end{aligned}$$

Finally, using the result from the previous section

$$\begin{aligned}
& \mathbb{E} \left| P_k U(\widehat{X}_k) - P_{V,k} U(\widehat{X}_k) \right| \\
& \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}} + \mathbb{E} \left| \phi_k^U(\widehat{X}_k) - P_{V,k} U(\widehat{X}_k) \right| \\
& \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}} + \sum_{i=1}^{N_k} \left| \phi_k^U(x_k^i) - P_{V,k} U(x_k^i) \right| \times \mathbb{P}(X_k \in C_k^i) \\
& \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}} + \sum_{i=1}^{N_k} \mathbb{E} |Q_k^i| + \sum_{\ell=1}^d \sum_{i=1}^{N_k} \sum_{j=1}^{N_{k+1}} \left| H_k^{\ell,ij} \right| \leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}}. \quad \square
\end{aligned}$$

In the algorithm we have in mind we want to compute $P_k U$ but we do not have access to the true value neither of U nor of V but only to some approximations $U' : \Gamma_{k+1} \rightarrow R$ and $V' : \Gamma_{k+2} \rightarrow R$. So we are interested to evaluate the impact of the error $U - U'$ and $V - V'$. The functions U' and V' are not related by $U' = P V'$ and this relation makes actually no sense because these functions are only defined on grids. Anyway, one may define $P_{V',k} U'(x_k^i)$ by (3.8).

Lemma 3 *Let $U, U' : \Gamma_{k+1} \rightarrow \mathbb{R}$, $V, V' : \Gamma_{k+2} \rightarrow \mathbb{R}$ be some functions. For every $p \in [1, 2)$, there is a real constant C_p , depending on p , on the diffusion process coefficients and on the real constant in (2.7), such that*

$$\begin{aligned}
\mathbb{E} |(P_{V,k} U - P_{V',k} U')(\widehat{X}_k)| & \leq \mathbb{E} |(U - U')(\widehat{X}_{k+1})| + \frac{C_p}{n} \mathbb{E} (|(V - V')(\widehat{X}_{k+2})|) \quad (3.12) \\
& + C_p n^{\frac{3}{p}-1} \|(V - V')(\widehat{X}_{k+2})\|_{\frac{p}{p-1}} \|X_{k+1} - \widehat{X}_{k+1}\|_2^{2/q} \quad (3.13)
\end{aligned}$$

Proof. We assume that $C_{V,\ell}^j = C_{V',\ell}^j$ so that these terms disappear when taking the difference. Moreover, having in mind the expression (3.7) of the weights, one obtains

$$\begin{aligned}
\mathbb{E} |(P_{V,k} U - P_{V',k} U')(\widehat{X}_k)| & = \sum_{i=1}^{N_k} \left| (P_{V,k} U - P_{V',k} U')(x_k^i) \right| \mathbb{P}(X_k \in C_k^i) \\
& \leq \mathbb{E} |(U - U')(\widehat{X}_{k+1})| + \underbrace{\mathbb{E} \left(|(V - V')(\widehat{X}_{k+2})| \sum_{\ell=1}^d |\rho_{k+1}^\ell(X_{k+1}, \Delta_{k+2})| \|X_{k+1} - \widehat{X}_{k+1}\| \right)}_{=:A}.
\end{aligned}$$

Set $Z_{k+2} := \sqrt{\frac{n}{T}} \Delta_{k+2} \stackrel{\mathcal{L}}{\sim} \mathcal{N}(0; I_d)$. Then (3.11) yields

$$A \leq C \sqrt{\frac{n}{T}} \mathbb{E} (|(V - V')(\widehat{X}_{k+2})| \|X_{k+1} - \widehat{X}_{k+1}\| (1 + |Z_{k+2}|)^2).$$

Let $p \in [1, 2)$ and $q := \frac{p}{p-1} \in (2, +\infty]$. It follows from Holder and Markov inequalities that

$$\begin{aligned}
A & \leq C \mathbb{E} (|(V - V')(\widehat{X}_{k+2})|) \\
& + C \sqrt{\frac{n}{T}} \|(V - V')(\widehat{X}_{k+2})\|_q \|X_{k+1} - \widehat{X}_{k+1}\| (1 + |Z_{k+2}|)^2 \mathbf{1}_{\{|X_{k+1} - \widehat{X}_{k+1}\| (1 + |Z_{k+2}|)^2 \geq n^{-\frac{3}{2}}\}} \|p \\
A & \leq \frac{C}{n} \mathbb{E} (|(V - V')(\widehat{X}_{k+2})|) + n^{\frac{3}{p}-1} \|(V - V')(\widehat{X}_{k+2})\|_q \|X_{k+1} - \widehat{X}_{k+1}\|_2^{2/p} \|(1 + |Z_{k+2}|)^4\|_p.
\end{aligned}$$

The last inequality uses that Z_{k+2} is independent of $\mathcal{F}_{t_{k+1}}$. \square

As an immediate consequence of the above lemmas one obtains

Proposition 4 *Let $U = P_k V$ with $V := P_{k+1} W$ where W is a Lipschitz continuous function on \mathbb{R}^d and let $U' : \Gamma_{k+1} \rightarrow \mathbb{R}$ and $V' : \Gamma_{k+2} \rightarrow \mathbb{R}$ be some real functions. Then, for every $p \in [1, 2)$, there is a real constant C_p , depending on p , on the diffusion process coefficients and on the constant in (2.7), such that*

$$\begin{aligned} \mathbb{E}|(P_k U - P_{V',k} U')(\widehat{X}_k)| &\leq \frac{C[W]_1 \sqrt{n}}{N^{2/d}} + \mathbb{E}|(U - U')(\widehat{X}_{k+1})| + \frac{C_p}{n} \mathbb{E}|(V - V')(\widehat{X}_{k+2})| \\ &\quad + C_p n^{\frac{3}{p}-1} \|(V - V')(\widehat{X}_{k+2})\|_q \|X_{k+1} - \widehat{X}_{k+1}\|_2^{2/p}. \end{aligned}$$

3.4 The algorithm for the Snell envelope

In this section we give the analogue of the algorithm in (2.4):

$$\begin{aligned} \widehat{u}_n(x_n^i) &:= h_n(x_n^i), & i = 1, \dots, N_n, & \quad (3.14) \\ \widehat{u}_k(x_k^i) &:= \max\left(h_k(x_k^i), \widehat{P}_k \widehat{u}_{k+1}(x_k^i)\right) & \text{with } \widehat{P}_k \widehat{u}_{k+1} &:= P_{\widehat{u}_{k+2},k} \widehat{u}_{k+1}, \quad 0 \leq k \leq n-2 \end{aligned}$$

and $\widehat{P}_{n-1} \widehat{u}_n(x_k^i) := \sum_{j=1}^{N_n} \pi_{n-1}^{ij} h(x_n^j)$. (Keep in mind that

$$\widehat{P}_k \widehat{u}_{k+1}(x_k^i) = \sum_{j=1}^{N_{k+1}} \pi_k^{ij} u_{k+1}(x_{k+1}^j) + \sum_{\ell=1}^d \sum_{j=1}^{N_{k+1}} \sum_{r=1}^{N_{k+2}} \pi_k^{\ell,ijr} (u_{k+1}(x_{k+2}^r) - C_{u_{k+1},\ell}^j), \quad 0 \leq k \leq n-2.$$

so that the definition of $\pi_k^{\ell,ijr}$ makes no sense for $k = n-1$). The constants $C_{u_{k+1},\ell}^j$ are to be chosen in order to reduce the variance. Now we would like to evaluate the error induced by our algorithm. However, we deal with a nonlinear problem since $u_k = \max(h_k, P_k u_{k+1}) \neq P_k u_{k+1}$ and some trouble appears – at least theoretically – in the computation of $\partial P_k u_{k+1}$ when hitting the obstacle. It seems difficult at this stage to provide an accurate description of this error although this difficulty occurs rather seldom. Furthermore the derivatives appear in the correctors, so they are already multiplied by small quantities. Anyway, numerical evidence show that things work well.

Whatsoever, to carry out our error estimates rigorously, we will behave as if we were solving a linear problem (namely computing $\mathbb{E}(h_n(X_n))$) using a “linear” dynamic programming formula *i.e.* removing the max).

A standard argument shows that, for every $q \in (2, +\infty)$, $\sup_{0 \leq k \leq n, n \in \mathbb{N}} \|u_{k+1}(\widehat{X}_{k+1})\|_q < +\infty$ if h has linear growth (this holds for $q = +\infty$ if h is bounded). Then Proposition 4 applied with $U := u_{k+1}$, $V := u_{k+2}$, $U' := \widehat{u}_{k+1}$, $V' := \widehat{u}_{k+2}$ leads to

$$\begin{aligned} \mathbb{E}|u_k(\widehat{X}_k) - \widehat{u}_k(\widehat{X}_k)| &\leq \mathbb{E}|P_k u_{k+1}(\widehat{X}_k) - \widehat{P}_k \widehat{u}_{k+1}(\widehat{X}_k)| = \mathbb{E}|P_k u_{k+1}(\widehat{X}_k) - P_{\widehat{u}_{k+2},k} \widehat{u}_{k+1}(\widehat{X}_k)| \\ &\leq \frac{C\sqrt{n}}{N^{2/d}} + \mathbb{E}|(u_{k+1} - \widehat{u}_{k+1})(\widehat{X}_{k+1})| + \frac{C_p}{n} \mathbb{E}|(u_{k+2} - \widehat{u}_{k+2})(\widehat{X}_{k+2})| \\ &\quad + C_p n^{\frac{3}{p}-1} \|X_{k+1} - \widehat{X}_{k+1}\|_2^{2/p} \end{aligned}$$

(where C_p also depends on h now). Iterating the above inequality yields,

$$\forall k \in \{0, \dots, n\}, \quad \mathbb{E}|u_k(\widehat{X}_k) - \widehat{u}_k(\widehat{X}_k)| \leq \frac{Cn\sqrt{n}}{N^{2/d}} + C_p n^{\frac{3}{p}-1} \sum_{k'=0}^n \|X_{k'+1} - \widehat{X}_{k'+1}\|_2^{2/p}.$$

Let us evaluate the impact of this inequality. If the marginal distributions are appropriately dominated (see [3]) – or at least heuristically – the optimality of the grids implies that $\sup_{0 \leq k \leq n, n \in \mathbb{N}} \|X_{k+1} - \widehat{X}_{k+1}\|_2 \leq CN^{-\frac{1}{d}}$. This finally yields the following global error bound: for every $\varepsilon := \frac{p-1}{p} \in (0, 1/2)$ (including $\varepsilon = 0$ if h is bounded),

$$|Y_0 - \widehat{u}_0(x_0)| \leq \frac{C}{n^\alpha} + |u_0(x_0) - \widehat{u}_0(x_0)| \leq \frac{C}{n^\alpha} + C_\varepsilon \left(\frac{n^3}{N^{2/d}} \right)^{1-\varepsilon} \quad (3.15)$$

for some real constant $C_\varepsilon > 0$. If h is semi-convex, then $\alpha = 1$ and if h is simply Lipschitz continuous, $\alpha = \frac{1}{2}$. This leads to the conjecture that the following error bounds hold true:

CONJECTURE: *Let $\varepsilon \in (0, 1/2)$ and suppose that $N \geq n^{(\frac{3+\alpha}{2}+\varepsilon)d}$. Then*

$$|Y_0 - \widehat{u}_0(x_0)| \leq \frac{C_\varepsilon}{n^\alpha} \quad (3.16)$$

where $\alpha = 1/2$ under (H_1) and $\alpha = 1$ under (H_2) . When h is also bounded, one may take $\varepsilon = 0$. The constant C_ε depends on the diffusion coefficients and on the constant in (2.7).

Comments. With the original 0th order algorithm, one needs $N = n^{2d+1}$ in order to get an error of order $1/n$ (when (H_2) holds true and the true diffusion (X_{t_k}) is quantized, see Theorem 5 in [3]). Still with this original algorithm, one needs $N = n^{\frac{3}{2}d+\frac{1}{2}}$ in order to get an error of order $1/\sqrt{n}$ (when (H_1) holds true and the Euler scheme is quantized, see [3]). So the theoretical gain derived from the above estimate may look poor. In fact, the above theoretical error bounds are probably not very sharp, especially because of Lemma 3. However, in view of Proposition 3 and Lemma 2 which evaluate “locally” the improvements brought by the 1st-order algorithm, one may hope to replace n^3 in (3.15) by a smaller coefficient, possibly $n\sqrt{n}$. Anyway, one verifies on numerical experiments that there is definitely a significant gain, especially for medium dimensions ($d \leq 6$).

3.5 Geometrical interpretation

We mentioned above that the 1–schemes correspond to the linear interpolation for the function u_{k+1} . The aim of this subsection is to make this assertion more precise. For simplicity we consider the one dimensional case only. So the points $x_k^j \in \mathbb{R}$ and we denote $I_k^i := [x_k^i, x_k^{i+1}]$. We also denote

$$u_{k+1}'^{r}(x_{k+1}^j) := \frac{u_{k+1}(x_{k+1}^{j+1}) - u_{k+1}(x_{k+1}^j)}{x_{k+1}^{j+1} - x_{k+1}^j}, \quad u_{k+1}'^{\ell}(x_{k+1}^j) := \frac{u_{k+1}(x_{k+1}^j) - u_{k+1}(x_{k+1}^{j-1})}{x_{k+1}^j - x_{k+1}^{j-1}}$$

and we think of $u_{k+1}'^{r}(x_{k+1}^j)$ (respectively of $u_{k+1}'^{\ell}(x_{k+1}^j)$) as an approximation of the right hand side (respectively of the left hand side) derivative of u_{k+1} at x_{k+1}^j . The linear

interpolation for $\{u_{k+1}(x_{k+1}^j), j = 1, \dots, N_{k+1}\}$ on $[x_{k+1}^1, x_{k+1}^{N_{k+1}}]$ is given by

$$\tilde{u}_{k+1}(y) = \sum_{1 \leq j \leq N_{k+1}} \mathbf{1}_{I_{k+1}^j}(y) \left(u_{k+1}(x_{k+1}^j) + u'_{k+1}(x_{k+1}^j)(y - x_{k+1}^j) \right)$$

In order to express this in terms of Voronoi tessels we denote $\overrightarrow{C}_{k+1}^j = [x_{k+1}^j, \frac{x_{k+1}^j + x_{k+1}^{j+1}}{2})$ and $\overleftarrow{C}_{k+1}^j = (\frac{x_{k+1}^{j-1} + x_{k+1}^j}{2}, x_{k+1}^j)$ so that $C_{k+1}^j = \overleftarrow{C}_{k+1}^j \cup \overrightarrow{C}_{k+1}^j$ and $I_{k+1}^j = \overrightarrow{C}_{k+1}^j \cup \overleftarrow{C}_{k+1}^{j+1}$. With this notation

$$\begin{aligned} \tilde{u}_{k+1}(y) &= \sum_{1 \leq j \leq N_{k+1}} \mathbf{1}_{C_{k+1}^j}(y) u_{k+1}(x_{k+1}^j) \\ &+ \sum_{1 \leq j \leq N_{k+1}} (\mathbf{1}_{\overrightarrow{C}_{k+1}^j}(y) u'_{k+1}(x_{k+1}^j)(y - x_{k+1}^j) + \mathbf{1}_{\overleftarrow{C}_{k+1}^j}(y) u'_{k+1}(x_{k+1}^j)(x_{k+1}^j - y)). \end{aligned}$$

As long as we are far from the free boundary, the above expressions yield

$$\begin{aligned} u_k(x_k^i) &= \mathbb{E}(\tilde{u}_{k+1}(B_{t_{k+1}})|B_{t_k} = x_k^i) = \sum_{1 \leq j \leq N_{k+1}} u_{k+1}(x_{k+1}^j) \mathbb{E}(\mathbf{1}_{C_{k+1}^j}(B_{t_{k+1}})|B_{t_k} = x_k^i) \\ &+ \sum_{1 \leq j \leq N_{k+1}} \mathbb{E} \left(\mathbf{1}_{\overrightarrow{C}_{k+1}^j}(B_{t_{k+1}})(B_{t_{k+1}} - x_{k+1}^j) | B_{t_k} = x_k^i \right) u'_{k+1}(x_{k+1}^j) \\ &+ \sum_{1 \leq j \leq N_{k+1}} \mathbb{E} \left(\mathbf{1}_{\overleftarrow{C}_{k+1}^j}(B_{t_{k+1}})(x_{k+1}^j - B_{t_{k+1}}) | B_{t_k} = x_k^i \right) u'_{k+1}(x_{k+1}^j). \end{aligned}$$

So the linear interpolation may be seen as a Taylor expansion of order one, with the derivative approximated by finite differences (in a different way in the left hand side and in the right hand side). This is exactly what we are doing in the 1- schemes. The only difference concerns the approximation that we use for the first order derivatives. The reason for which we do not use finite differences approximations is because this kind of scheme is not available in the multi-dimensional case, when the grid is not regular (we mean hypercubes): optimal grids are never regular.

Let us now come back to our way of computing derivatives. We stress that this is based on the fact that $u_{k+1}(x_{k+1}^j)$ is itself an expectation. The formula

$$u'_{k+1}(x_{k+1}^j) = \mathbb{E} \left((u_{k+2}(\theta_k(x_{k+1}^j, \Delta_{k+2})) - C) \rho_k^1(x_{k+1}^j, \Delta_{k+2}) \right)$$

gives a pathwise interpretation of the derivative and this is the basic fact which allows us to compute the derivatives using a Monte Carlo method.

There is one more difference between our method and the linear interpolation method. In the computation of $u'_{k+1}(x_{k+1}^j)$ using the finite difference method one uses two values of u_{k+1} : at x_{k+1}^j and at x_{k+1}^{j+1} (respectively in x_{k+1}^{j-1}). In our method we do not use the values of u_{k+1} but of u_{k+2} . Moreover, we use all the values $(u_{k+2}(x_{k+2}^j))_{1 \leq j \leq N_{k+2}}$. Finally we stress that our interpolation is piecewise linear but not continuous.

3.6 Numerical experiments

We now present numerical experiments on pricing American exchange style options based on algorithms (2.4) and (3.14) in dimension 4, 6 and 10. The obstacle chosen here is the following exchange style pay-off

$$h(t, S_t) = \left(S_t^1 \dots S_t^{d/2} - S_t^{d/2+1} \dots S_t^d \right)_+,$$

where $t \in [0, T]$ and where the d -dimensional price process $\{S_t\}_{t \in [0, T]}$ follows a standart Blake-Sholes model with null interest rate, a dividend rate vector $\mu = [5\%, 0, \dots, 0]^t$ and a diagonal constant volatility matrix $\sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$ with $\sigma_i = \sqrt{2/d} 20\%$. The initial conditions are $S_0^i = (40)^{2/d}$ for $i \in \{1, \dots, d/2\}$ and $S_0^i = (36)^{2/d}$ for $i \in \{d/2 + 1, \dots, d\}$.

In the following simulations, we have set the maximal maturity time $T = 1$ year and the time step T/n with $n = 24$ (dimension 4 and 6), $n = 48$ (dimension 10). The number of points on the top layer is $N_{24} = 500$ (dimension 4), $N_{24} = 1000$ (dimension 6) and $N_{48} = 1000$ (dimension 10). The number of points per time layer is then fixed as explained in Section 2.3. We have to point out here that due to the fact that the price process is an explicit function of the Brownian motion, the computations of the weights (2.3) and (3.7) are done for the d -dimensional Brownian motion. This part of the computation is then parameter free and have to be done once for all. Therefore algorithms (2.4) and (3.14) can be used for every choice of parameter (dividend rate, volatilities, initial conditions) with the same weights. We denote by $AM_0(0, T)$ (resp. $AM_1(0, T)$) the price at time $t = 0$ and maturity T computed with (2.4) (resp. with (3.14)). Finally, we denote by $AM_{ref}(0, T)$ a reference price performed in dimension 2 by a finite difference scheme on the associated PDE formulation [7] with $\sigma_1 = \sigma_2 = 20\%$, $\mu_1 = 5\%$, $\mu_2 = 0$, $S_0^1 = 40$ and $S_0^2 = 36$.

In Table 1, the relative error of $AM_0(0, T)$ and $AM_1(0, T)$ with respect to $AM_{ref}(0, T)$ are displayed. First we observe that in every case the relative errors do not overcome 4% for 12 months and 2% for 6 months. For 6 months of maturity, the corrected weights seems to be very pertinent in every dimension since it reduces the relative errors by a factor of 30 up to 50. As maturity grows up (9 and 12 months), the correction remains usefull for $d = 4$ and $d = 6$ in order to reduce below 1% the relative error. The use of corrected terms in dimension 10 for these maturities seems not to be very relevant. Indeed, in this case, the optimization of the grids is very hard to achieve. Therefore the approximation of the weights is no more consistent because we have to take into account the *bottom* error (see section 3.2). But taking into account those corrections is pratically untractable in this dimension due to the high cost of the storage. This shows the great importance of optimal grids in high dimension.

References

- [1] V. Bally, G. Pagès, J. Printems (2001): A stochastic quantization method for non linear problems, *Monte Carlo Methods and Applications*, **7**, n^o 1-2, pp.21-34.
- [2] V. Bally, G. Pagès, J. Printems: (2002): A quantization tree method for pricing and hedging multi-dimensional American options, preprint de l'INRIA & pre-print, Labo. de Proba. & Modèles aléatoires, Univ. Paris 6 (France).
- [3] V. Bally, G. Pagès: (2000): A quantization algorithm for solving multi-dimensional optimal stopping problems, pre-print n^o 628, Labo. de Proba. & Modèles aléatoires, Univ. Paris 6 (France).
- [4] V. Bally, G. Pagès (2001): Error analysis of the quantization algorithm for obstacle problems, pre-print n^o642, Laboratoire de Probabilités & Modèles aléatoires, Université Paris VI (France), forthcoming in *Stoch. Proc. and their Appl.*.
- [5] S. Graf, H. Luschgy (2000): Foundations of quantization for probability distributions, Lecture Notes in Mathematics n^o1730, Springer, 230p.
- [6] G. Pagès (1997): A space vector quantization method for numerical integration. *Journal of Applied and Computational Mathematics*, 89, pp.1-38.
- [7] S. Villeneuve, A. Zanette (2002): Parabolic A.D.I. methods for pricing american option on two stocks, forthcoming in *Mathematics of Operation Research*.

Maturity	3 months		6 months		9 months		12 months	
AM_{ref}	4.4110		4.8969		5.2823		5.6501	
	Price	Error (%)						
$d = 4$								
AM_0	4.4076	0.08	4.9169	0.34	5.3284	0.82	5.7366	1.39
AM_1	4.4058	0.1	4.8991	0.04	5.2881	0.08	5.6592	0.13
$d = 6$								
AM_0	4.4156	0.1	4.9276	0.63	5.3550	1.38	5.7834	2.20
AM_1	4.4099	0.02	4.8975	0.01	5.3004	0.34	5.6557	0.10
$d = 10$								
AM_0	4.4317	0.47	4.9945	2.00	5.4350	2.89	5.8496	3.53
AM_1	4.4194	0.19	4.8936	0.07	5.1990	1.58	5.4486	3.56

Table 1: Relative errors of AM_0 and AM_1 with respect to a reference price for different maturities and dimensions.