Méthodes de Monte Carlo et Algorithmes Stochastiques

BERNARD LAPEYRE

Septembre 2018

Table des matières

Chapitre 2

Monte–Carlo methods for pricing American options.

We present some recent Monte–Carlo algorithms devoted to the pricing of American options. This part follows the work done by Pierre Cohort (see [Cohort(2001)]). We give the description of two class of algorithm. The first class use an approximation of conditional expectation (Longstaff and Schwartz [Longstaff and Schwartz(1998)] and Tsitsiklis and VanRoy [Tsitsiklis and Roy(2000)]). The second one is based on an approximation a the underlying Markov chain (quantization algorithms [Pagès and Bally(2000), Pagès et al.(2001)], Broadie and Glassermann [Broadie and Glassermann(1997a), Broadie and Glassermann(1997b)] and Barraquand and Martineau [Barraquand and Martineau(1995)]).

2.1 Introduction

We assume that an American option can only be exercised at a given set of time $(t_0,\dots,t_N = T)$. In this case we say that the option is a Bermudean option.

The asset price at time t_n is supposed to be given by a Markov chain $(X_n, 0 \le n \le N)$ with a transition matrix between n and $n+1$ given by $P_n(x,dy)$. This means that for every bounded f

$$\mathbb{E}\left(f(X_{n+1})|\mathscr{F}_n\right) = P_n f(X_n) := \int f(y)P_n(X_n,dy),$$

where $\mathscr{F}_n = \sigma\left(X_k, k \le n\right)$.

We assume that the payoff at time t_n is given by $\varphi(X_n)$ where φ is a bounded function and that the actualization between time t_j and t_{j+1} can be expressed as $1/(1+R_j)$, R_j being a constant positive interest rate between time t_j and t_{j+1}. This allows us to define an actualization coefficient B between arbitrary time j and k, when $j < k$, by setting

$$B(j,k) = \prod_{l=j}^{k-1} \frac{1}{1+R_l}.$$

By convention we set $B(j,j) = 1$. Note that $B(j,j+1) = 1/(1+R_j)$. With these notations, the price at time 0 of this option is

$$Q_0 = \sup_{\tau \in \mathscr{T}_{0,N}} \mathbb{E}\left(B(0,\tau)\,\varphi(X_\tau)\right) \tag{2.1}$$

where $\mathscr{T}_{j,N}$ is the set of $\mathscr{F}_n$-stopping times taking values in $\{j,\ldots,N\}$.

Remark 2.1.1. For the d-dimensional Black and Scholes model the assets prices $(S_t^{x,i}, 1 \le i \le d)$ can be written as

$$S_t^{x,i} = x_i e^{\left(r - \frac{1}{2}\sum_{1\le j\le p} \sigma_{ij}^2\right)t + \sum_{1\le j\le p} \sigma_{ij} W_t^j}$$

where $(W_t, t \ge 0)$ is a p–dimensional Brownian motion, σ is a $d \times p$ matrix, x belongs to $\mathbb{R}^d$, $\sigma_i^2 = \sum_{1\le j\le p} \sigma_{ij}^2$ and $r > 0$ is the risk-less interest rate. So P_n is defined by

$$P_n f(x) = \mathbb{E}\left(f(S^x_{t_{n+1}-t_n})\right).$$

Moreover, we have

$$1 + R_n = \exp(r(t_{n+1} - t_n)).$$

Standard theory of optimal stopping proves that $Q_0 = u(0, X_0)$, where u is the solution of the dynamic programming algorithm

$$\begin{cases} u(N,x) = \varphi(x) \\ u(j,x) = \max\left(\varphi(x), \tilde{P}_j u(j+1,x)\right), \quad 0 \le j \le N-1. \end{cases} \tag{2.2}$$

and $\tilde{P}_j(x,dy)$ is the discounted transition kernel of the Markov chain $(X_j, j = 0,\ldots,N)$ between j and $j+1$ given by

$$\tilde{P}_j f(x) = B(j,j+1) P_j f(x).$$

The stopping time

$$\tau^* = \inf\left\{j \ge 0, u(j,X_j) = \varphi\left(X_j\right)\right\}. \tag{2.3}$$

is optimal, that is to say

$$Q_0 = u(0,X_0) = \mathbb{E}\left(B(0,\tau^*)\,\varphi(X_{\tau^*})\right).$$

Moreover, the price Q_j of this American option at time j given by

$$Q_j = \sup_{\tau \in \mathscr{T}_{j,N}} \mathbb{E}\left(B(j,\tau)\,\varphi(X_\tau)\,|\mathscr{F}_j\right),$$

can be computed as $u(j,X_j)$ and and optimal stopping time at time j is given by τ_j^* where

$$\tau_j^* = \inf\{i \ge j; u(i,X_i) = \varphi(X_i)\}.$$

2.2 The Longstaff and Schwartz algorithm.

This algorithm approximate the optimal stopping time τ^* on M given paths by random variables $\tau^{(m)}$ depending on the path m and then estimate the price Q_0 according to a Monte-Carlo formula

$$Q_0^M = \frac{1}{M} \sum_{1<i<M} B(0,\tau^{(m)})\varphi\left(X_{\tau^{(m)}}^{(m)}\right). \tag{2.4}$$

The construction of $(\tau^m, 1 \leq m \leq M)$ is presented according to the following steps. First we write the dynamic programming algorithm directly on the optimal stopping time τ^*. Then we present the Longstaff and Schwartz approximation of this recursion. Finally we give a Monte-Carlo version, leading to $\tau^{(m)}$.

A dynamical programming algorithm for optimal stopping time. The main feature of the Longstaff and Schwartz algorithm is to use a dynamic principle for optimal stopping times rather than for the value function. Note that that τ^* can be computed using the sequence $(\tau_j^*, 0 \leq j \leq N)$ defined by the backward induction

$$\begin{cases} \tau_N^* = N, \\ \tau_j^* = j\mathbf{1}_{\varphi(X_j) \geq u(j,X_j)} + \tau_{j+1}^* \mathbf{1}_{\varphi(X_j) < u(j,X_j)}. \end{cases} \tag{2.5}$$

Note that

$$\begin{aligned} \left\{\varphi(X_j) < u(j,X_j)\right\} &= \left\{\varphi(X_j) < \tilde{P}_j u(j+1,X_j))\right\} \\ &= \left\{\varphi(X_j) < \mathbb{E}\left(B(j,\tau_{j+1}^*)\varphi(X_{\tau_{j+1}^*})\middle| X_j\right)\right\}. \end{aligned} \tag{2.6}$$

With this definition for τ_j^*, it is easy to check recursively that

$$\tau_j^* = \min\left\{i \geq j, \varphi\left(X_i\right) = u(i,X_i)\right\}. \tag{2.7}$$

Thus τ_j^* (and thus $\tau^* = \tau_0^*$) are optimal stopping times at time j. In order to estimate τ^* we have to find a way to approximate the conditional expectation

$$\mathbb{E}\left(B\left(j,\tau_{j+1}^*\right)\varphi\left(X_{\tau_{j+1}^*}\right)|X_j\right). \tag{2.8}$$

The basic idea of Longstaff and Schwartz is to introduce a least square regression method to perform this approximation.

The regression method We denote by Z_{j+1} the random variable

$$Z_{j+1} = B\left(j,\tau_{j+1}^*\right)\varphi\left(X_{\tau_{j+1}^*}\right).$$

Obviously, as B and φ are bounded, Z_{j+1} is also a bounded random variable. Let us recall that the conditional expectation

$$\mathbb{E}\left(Z_{j+1}|X_j\right) \tag{2.9}$$

can be expressed as $\psi_j(X_j)$, where ψ_j minimizes

$$\mathbb{E}\left(\left[X_{j+1} - f(X_j)\right]^2\right)$$

among all functions f such that $\mathbb{E}\left(f(X_j)^2\right) < +\infty$. This come from the definition of the conditional expectation as a L^2-projection on the set of the $\sigma(X_j)$-measurable random variables and the well known property that all these random variables can be writen as $f(X_j)$.

Assume now that we have a sequence of functions $(g_l, l \geq 1)$ which is a total basis of $L^2_j = L^2\left(\mathbb{R}^d, \text{law of } X_j\right)$ for every time $j, 1 \leq j \leq N$. For all time index j we are thus able to express ψ_j as

$$\psi_j = \sum_{l\geq 1} \alpha_l g_l,$$

where the convergence of the series has to be understood in L^2_j. So we have a way to compute recursively the optimal exercise time

1. initialize $\tau_N = N$. Then inductively
2. define $\alpha^j = (\alpha^j_l, l \geq 0)$ as the sequence which minimizes
$$\mathbb{E}\left(\left[B\left(j,\tau^*_{j+1}\right)\varphi\left(X_{\tau^*_{j+1}}\right) - (\alpha\cdot g)(X_j)\right]^2\right)$$
where $\alpha\cdot g = \sum_{l\geq 1}\alpha_l g_l$.
3. define $\tau^*_j = j\mathbf{1}_{\{\varphi(X_j)\geq(\alpha^j\cdot g)(X_j)\}} + \tau^*_{j+1}\mathbf{1}_{\{\varphi(X_j)<(\alpha^j\cdot g)(X_j)\}}$.

This program is not really implementable since we have to perform a minimization in an infinite dimensional space and we are not able to compute the expectation involved in the minimization problem in step 2.

In order to implement the algorithm, the basic idea is to truncate the series at index k. This leads to the following modified program.

1. initialize $\hat{\tau}_N = N$. Then inductively
2. define $\widehat{\alpha}^{j,k} = (\widehat{\alpha}^{j,k}_l, 0 \leq l \leq k)$ as the vector which minimizes
$$\mathbb{E}\left(\left[B\left(j,\hat{\tau}_{j+1}\right)\varphi\left(X_{\hat{\tau}_{j+1}}\right) - (\hat{\alpha}^{j,k}\cdot g)(X_j)\right]^2\right) \tag{2.10}$$
where $(\widehat{\alpha}^{j,k}\cdot g) = \sum_{l=1}^k \widehat{\alpha}^{j,k}_l g_l$.
3. define
$$\widehat{\tau}_j = j\mathbf{1}_{\varphi(X_j)\geq(\widehat{\alpha}^{j,k}\cdot g)(X_j)} + \widehat{\tau}_{j+1}\mathbf{1}_{\varphi(X_j)<(\widehat{\alpha}^{j,k}\cdot g)(X_j)}.$$

To make the previous program implementable, we introduce a Monte-Carlo version of (2.10).

A Monte-Carlo approach to the regression problem Let $(X^{(m)}_n, 0 \leq n \leq N)$, for $1 \leq m \leq M$ be M paths sampled along the law of the process $(X_n, 0 \leq n \leq N)$. We replace the minimization problem 2.10 by

1. initialize $\tau^M_N = N$. Then inductively
2. define $\alpha^{j,k}_M = (\alpha^{j,k}_M, 0 \leq j \leq k)$ as the vector which minimizes
$$\frac{1}{M}\sum_{1\leq m\leq M}\left((\alpha\cdot g)\left(X^{(m)}_j\right) - B\left(j,\tau_{j+1}\right)\varphi\left(X^{(m)}_{\tau_{j+1}}\right)\right)^2 \tag{2.11}$$

3. define for each trajectory m

$$\tau_j^{(m)} = j\mathbf{1}_{\left\{\varphi(X_j)\geq(\alpha_M^{j,k}\cdot g)(X_j)\right\}} + \tau_{j+1}^{(m)}\mathbf{1}_{\left\{\varphi(X_j)<(\alpha_M^{j,k}\cdot g)(X_j)\right\}}.$$

This algorithm is now fully implementable and that the estimator of the price is given by

$$\frac{1}{M}B(0,\tau^{(m)})\varphi(X_{\tau^{(m)}}^{(m)}).$$

Remark 2.2.1. — The minimization problem (2.11) is a standard least square approximation problem. An algorithm for solving it can be found for instance in [Press et al.(1992)].

— The basis $(g_k, k \geq 1)$ is supposed to be independent of the time index j. This is just for sake of simplicity and one can change the basis at each time index if needed for numerical efficiency. For instance, it is quite usual to add the payoff function in the basis, at least when j approaches N.

— Very often for financial products such as call or put options,

$$\mathbb{P}\left(\varphi(X_j) = 0\right) > 0,$$

for every time j. Since $\left\{\varphi\left(X_j\right) = 0\right\} \subset A_j$, it is useless to compute

$$\mathbb{E}\left(B\left(j,\tau_{j+1}^*\right)\varphi\left(X_{\tau_{j+1}^*}\right)\middle|X_j\right)$$

on the set $\left\{\varphi\left(X_j\right) = 0\right\}$ and we can thus restrict ourself in the regression to trajectories such that $\left\{\varphi\left(X_j\right) > 0\right\}$.

— A rigorous proof of the convergence of the Longstaff Schwartz algorithm is given in [Clément et al.(2002)].

2.3 The Tsitsiklis–VanRoy algorithm.

This algorithm use a revised dynamical programming algorithm similar to (2.2) but involving

$$\widetilde{u}(j,x) = \tilde{P}_j u(j+1,x)$$

instead of $u(j,x)$. An approximation of $\widetilde{u}$ is then performed by using a regression method.

Another form of the dynamic programming algorithm. Using $\widetilde{u}$, the algorithm (2.2) can be rewriten as

$$\begin{cases} \widetilde{u}(N,x) = \tilde{P}_{n-1}\varphi(x) \\ \text{For } 0 \leq j \leq N-2, \\ \qquad \widetilde{u}(j,x) = \tilde{P}_j \max\left(\varphi(.),\tilde{u}(j+1,.)\right)(x). \end{cases} \tag{2.12}$$

So that $Q_0 = \max(\varphi(x_0),\widetilde{u}(0,x_0))$.

For every $1 \leq j \leq N-1$, let $(g_l, l \geq 1)$ be a basis of L_j^2. At time $N-1$, the authors approximate $\tilde{u}(j,x)$ by its projection on the subspace spanned by $\{g_1,\ldots,g_k\}$, according to the L_j^2 norm. They iterate this procedure backward in time, projecting at time j the function

$$B\left(j,j+1\right)\max\left(\varphi(x),(\alpha^{j+1}\cdot g)(x)\right)$$

on the vector space generated by $g_1,\dots,g_k$ according to the L^2_j norm. The resulting approximated dynamical programming algorithm is then

$$\left\{\begin{array}{l} \alpha^{N-1}=D_{N-1}^{-1}\mathbb{E}\left(B\left(N-1,N\right)\varphi\left(X_N\right)^t g^{N-1}\left(X_{N-1}\right)\right) \\ \text{For } 1\leq j\leq N-2 \\ \quad \alpha^j=D_j^{-1}\mathbb{E}\left(B\left(j,j+1\right)\max\left(\varphi\left(X_{j+1}\right),\left(\alpha^{j+1}\cdot g\right)\left(X_{j+1}\right)\right)^t g\left(X_j\right)\right) \\ \alpha=\mathbb{E}\left(B\left(0,1\right)\max\left(\varphi\left(X_1\right),\left(\alpha^1\cdot g\right)\left(X_1\right)\right)\right) \end{array}\right. \tag{2.13}$$

where D_j^M is the covariance matrix of the vector $g(X_j)$ defined as

$$(D_j)_{k,l}=\mathbb{E}\left(g_k(X_j)g_l(X_j)\right)-\mathbb{E}\left(g_k(X_j)\right)\mathbb{E}\left(g_l(X_j)\right).$$

At time 0, the proposed approximation of the price is given by

$$\max\left(\varphi\left(x_0\right),\alpha\right).$$

The Monte-Carlo algorithm. In order to obtain an implementable algorithm we use a set of trajectories denoted by $(X^{(m)},1\leq m\leq M)$ sampled along the law of X. The Monte-Carlo version of (2.13) is

$$\left\{\begin{array}{l} \alpha_M^{N-1}=(D_{N-1}^M)^{-1}\frac{1}{M}\sum_{1\leq l\leq M}B\left(N-1,N\right)\varphi\left(X_N^{(m)}\right)^t g\left(X_{N-1}^{(m)}\right) \\ \text{For } 1\leq j\leq N-2, \\ \quad \alpha_M^j=(D_j^M)^{-1}B\left(j,j+1\right)\times \\ \qquad \times\frac{1}{M}\sum_{1\leq l\leq M}\max\left(\varphi\left(X_{j+1}^{(m)}\right),\left(\alpha_M^{j+1}\cdot g\right)\left(X_{j+1}^{(m)}\right)\right)^t g\left(X_j^{(m)}\right) \\ \alpha_M^0=B\left(0,1\right)\frac{1}{M}\sum_{1\leq l\leq M}\max\left(\varphi\left(X_1^{(m)}\right),\left(\alpha_M^1\cdot g\right)\left(X_1^{(m)}\right)\right), \end{array}\right. \tag{2.14}$$

where D_j denotes the empirical dispersion matrix of the set of points

$$(g(X_j^{(m)}),1\leq m\leq M)$$

of $\mathbb{R}^k$.The price at time 0 is approximated by $\max\left(\varphi\left(X_0\right),\alpha_M^0\right)$.

2.4 The quantization algorithms.

Algorithms based on quantization methods have been introduced by Bally and Pagès in [Pagès and Bally(2000)]. The principle of these algorithms is to discretize, at time j, the underlying Markov process X_j by a set of "quantization levels" $y^j=\left(y_i^j\right)_{1\leq i\leq n_j}$ and then to approximate the transition kernel of X at time j by a discrete kernel $\widehat{P}_j=(\widehat{P}_j(y_k^j,y_l^{j+1}),1\leq k\leq n_j,l\leq n_{j+1})$.

More precisely, let $P_j\left(x,dy\right)$ be the transition kernel between times j and $j+1$ of a Markov chain $(X_n,0\leq n\leq N)$ taking its values in $\mathbb{R}^n$ and starting from x_0.

For $1\leq j\leq N$, let y^j be a set of vectors of $\mathbb{R}^d$

$$y^j=\left\{y_1^j,\dots,y_{n_j}^j\right\}.$$

Note that $X_0 = x_0$ is supposed to be constant and need not to be discretized. We recall that

$$\sup_{\tau \in \mathscr{T}_{0,N}} \mathbb{E}\left(B(0,\tau)\,\varphi(X_\tau)\right) = u(0,x_0),$$

where u is given by the dynamic programming algorithm (2.2)

$$\begin{cases} u(N,x) = \varphi(x) \\ u(j,x) = \max\left(\varphi(x), \tilde{P}_j u(j+1,x)\right) \quad 0 \le j \le N-1. \end{cases} \tag{2.15}$$

We will now approxime the actualized transition kernel $\tilde{P}_j B(j,j+1) P_j$ by a discrete transition kernel $\widehat{P}_j$ defined on $y^j \times y^{j+1}$. This kernel must satisfy the following requirement : $\widehat{P}_j\left(y_k^j, y_l^{j+1}\right)$ approximate the actualized probability that the process X moves from a neighborhood of y_k^j at time j to a neighborhood of y_l^{j+1} at time $j+1$. This can be done using $\widehat{P}_j$ defined by

$$\widehat{P}_j\left(y_k^j, y_l^{j+1}\right) = B(j,j+1)\frac{\mathbb{P}(X_j \in C_k(y^j), X_{j+1} \in C_l(y^{j+1}))}{\mathbb{P}(X_j \in C_k(y^j))} \tag{2.16}$$

where the set $C_i\left(y^j\right)$ is the i^{th} Voronoï cell of the set of points y^j, defined by

$$C_i\left(y^j\right) = \left\{ z \in \mathbb{R}^d, \left\| z - y_i^j \right\| = \min_{1 \le l \le n_j} \left\| z - y_l^j \right\| \right\}. \tag{2.17}$$

If we assume that, for every $x \in \mathbb{R}^d$ and l, $P_j\left(x, \partial C_l\left(y^{j+1}\right)\right) = 0$, equation (2.16) defines a probability transition kernel.

The approximation of the dynamic programming algorithm (2.15) using the matrix $\widehat{P}_j$ is given by

$$\begin{cases} \widehat{u}(N, y_k^N) = f\left(y_k^N\right) \quad 1 \le k \le n_N \\ \text{For } 0 \le j \le N-1,\, 1 \le k \le n_j, \\ \qquad \widehat{u}(j, y_k^j) \max\left(f\left(y_k^j\right), \displaystyle\sum_{1 \le l \le n_{j+1}} \widehat{P}_j\left(y_k^j, y_l^{j+1}\right) \widehat{u}(j+1, y_l^{j+1}) \right) \end{cases} \tag{2.18}$$

and the proposed price approximation is $\widehat{u}(0,x_0)$.

In order to have an algorithm we need to construct a Monte-Carlo procedure to approximate $\widehat{P}_j$. Let $(X^{(m)}, 1 \le m \le M)$ be M sample paths drawned along the law of the markov chain X and independent. The formula (2.16) suggests to consider the MonteCarlo estimator $\bar{P}_j$ defined by

$$\bar{P}_j\left(y_k^j, y_l^{j+1}\right) = B(j,j+1)\frac{\sum_{1 \le m \le M} \mathbf{1}_{\left\{X_j^{(m)} \in C_k(y^j), X_{j+1}^{(m)} C_l(y^{j+1})\right\}}}{\sum_{1 \le m \le M} \mathbf{1}_{\left\{X_j^{(m)} \in C_k(y^j)\right\}}} \tag{2.19}$$

An implementable version of (2.18) obtained by replacing $\widehat{P}_j$ by $\bar{P}_j$ in (2.18).

Remark 2.4.1. When $\widehat{u}$ has been computed at the point $(y_i^j, 1 \le i \le n_j)$, we can extend it to $\mathbb{R}^n$ by setting $\widehat{u}(j,x) = \widehat{u}(j, y_i^j)$ for $x \in C_i(y^j)$. This allows us to approximate the stopping time τ^* by

$$\widehat{\tau} = \min\left\{ j; \varphi(X_j) = \widehat{u}(j, X_j) \right\}, \tag{2.20}$$

and leads to another approximation for the price at time zero by

$$\frac{1}{M}\sum_{1\leq m\leq M} B\left(0,\widehat{\tau}^{(m)}\right) f\left(\mathscr{Q}_{\widehat{\tau}^{(m)}}, X^{(m)}_{\widehat{\tau}^{(m)}}\right),$$

where $\widehat{\tau}^{(m)}$ is the stopping time computed using 2.20 on the trajectory m.

Choice of the quantization sets There are many ways to implement the quantization method, depending on the choice of the quantization sets y^j. We present here two of them.

The first one is the Random Quantization algorithm. We assume that $n_j = n$ for all j. We use a self–quantization procedure to generate the quantization sets, that is to say we set

$$y^j_m = X^{(m)}_j \quad 1 \leq j \leq n \quad 1 \leq m \leq n.$$

where $X^{(m)}, 1 \leq m \leq M$ are independently drawn along the law of X. This choice is acceptable but is not the best among all the random choices (see [Cohort(2004)]).

A second way to choose the quantization sets is to minimize a quantity reflecting the quality of the quantization set. The distortion $\mathscr{D}$ defined for a set $y = (y_k, 1_leqk \leq n)$ and a random variable X by

$$\mathscr{D}(y,X)\,\mathbb{E}\left(\min_{1\leq k\leq n} \|X - y_k\|^2\right)$$

is a classical choice for this. Moreover, an error bound for the previously defined approximation of the price can be derived in terms of the distortions $\left(\mathscr{D}\left(y^j, X_j\right), 1 \leq j \leq N\right)$ (see [Pagès and Bally(2000)]). It can be shown that there exists an optimal set y^* satisfying

$$\mathscr{D}(y^*,X) = \min_{y\in\left(\mathbb{R}^d\right)^n} \mathscr{D}(y,X)$$

This optimal set can be computed numerically by using the Lloyd algorithm : let $y^{(0)} \in \left(\mathbb{R}^d\right)^n$ be such that all points are different and define a sequence $y^{(n)}$ setting

$$y^{(n+1)}_k = \mathbb{E}\left(Y|Y \in C_k\left(y^{(n)}\right)\right).$$

Note that a Monte-Carlo method must be used to approximate the previous expectation. Then $\mathscr{D}\left(y^{(n)}, X\right)$ is decreasing and the sequence $y^{(n)}$ converges to a point y^* which realizes a local minimum of $\mathscr{D}(.,X)$. For more details on the Lloyd algorithm and its convergence see [Sabin and Gray(1986)]. Another classical optimization procedure for $\mathscr{D}$ uses the Kohonen algorithm, see [Fort and Pagès(1995)].

2.5 The Broadie–Glassermann algorithm.

For this algorithm we use a random mesh depending on time and write a dynamical programming algorithm at points of this mesh.

Mesh generation. We assume that the transition matrices on $\mathbb{R}^n$ of the process (X_n, n_geq0) $P_j(x,dy)$ have a density

$$f_j(x,y).$$

This is obviously the case in the Black and Scholes model. We assume moreover that $X_0 = x_0 \in \mathbb{R}^n$.

Let $(Y_1^{(m)}, 1 \leq m \leq n)$ be a sample following the density $f_0(x_0,y)$. The set $(Y_1^{(m)}, 1 \leq m \leq n)$ define the mesh at time 1.

Then the mesh at time $j+1$ $(Y_{j+1}^{(m)}, 1 \leq m \leq n)$ is defined assuming that $Y_j = (Y_j^{(m)}, 1 \leq m \leq n)$ is already drawn and that the conditional law of Y^{j+1} knowing Y^j is the law of n independant random variables with (conditional) density

$$\frac{1}{n} \sum_{1 \leq l \leq n} f_j\left(Y_j^{(l)}, y\right) dy. \tag{2.21}$$

The next step consists in writing an approximation of the dynamical programming algorithm

$$\begin{cases} u(N,x) = \varphi(x), \\ u(j,x) \max\left(\varphi(x), \tilde{P}_j u(j+1,x)\right) \quad 0 \leq j \leq N-1. \end{cases} \tag{2.22}$$

using only the points $(Y_j^{(m)}, 1 \leq m \leq n)$. For this Broadie and Glassermann propose to approximate

$$P_j u(j+1, Y_j^{(k)}),$$

by a sum

$$\frac{1}{n} \sum_{1 \leq l \leq n} \hat{\beta}_j^{k,l}(Y_j^{(k)}, Y_{j+1}^{(m)}) u(j+1, Y_{j+1}^{(m)}). \tag{2.23}$$

They choose the weights $\hat{\beta}_{kl}^j$ by considering (2.23) as a Monte-Carlo sum. Indeed, one has

$$\begin{aligned} &\mathbb{E}\left(u(j+1, Y_{j+1}^{(m)}) \frac{f_j\left(Y_j^{(k)}, Y_{j+1}^{(m)}\right)}{\frac{1}{n}\sum_{1 \leq l \leq n} f_j\left(Y_j^{(l)}, Y_{j+1}^{(m)}\right)} \middle| Y_j^{(k)}, k = 1, \ldots, n \right) = \\ &\quad = \int_{\mathbb{R}^d} u(j+1,u) \frac{f_j\left(Y_j^{(k)}, u\right)}{\frac{1}{n}\sum_{1 \leq l \leq n} f_j(Y_j^{(l)}, y)} \frac{1}{n} \sum_{1 \leq l \leq n} f_j\left(Y_j^{(l)}, y\right) dy \\ &\quad = \int_{\mathbb{R}^d} u(j+1,u) f_j\left(Y_j^{(k)}, y\right) dy \\ &\quad = P_j u(j+1, Y_j^{(k)}) \end{aligned} \tag{2.24}$$

But $(Y_{j+1}^{(m)}, 1 \leq m \leq n)$ are, conditionally to $(Y_j^{(m)}, 1 \leq m \leq n)$, independant and identically distributed. So a natural estimator for (2.24) is given by

$$\frac{1}{n} \sum_{1 \leq m \leq n} \hat{\beta}_{k,m}^j u(j+1, Y_{j+1}^{(m)}) \tag{2.25}$$

where

$$\hat{\beta}_{k,m}^j = \frac{f_j\left(Y_j^{(k)}, Y_{j+1}^{(m)}\right)}{\frac{1}{n}\sum_{1 \leq l \leq n} f_j\left(Y_j^{(k)}, Y_{j+1}^{(l)}\right)}. \tag{2.26}$$

The approximation of the dynamical programming algorithm is then given by

$$\begin{cases} \widehat{u}(N, Y^N_{(k)}) = \varphi(Y^N_{(k)}), \\ \text{for } 0 \leq k \leq n,\ 1 \leq j \leq N, \\ \quad \widehat{u}(j, Y^{(k)}_j) \max\left(\varphi\left(Y^{(k)}_j\right), \sum_{1 \leq l \leq n} \hat{\beta}^j_{k,m}, \widehat{u}(j+1, Y^{(m)}_{j+1})\right), \end{cases}$$

and the price approximation by $\widehat{u}(0, x_0)$.

2.6 The Barraquand–Martineau algorithm

To avoid dimensionality issue, the authors propose to approximate the optimal stopping strategy a sub–optimal one : assume that the option holder knows at time j the payoff values $\{\varphi(X_i), i \leq n\}$ but not the stock values $\{X_i, i \leq n\}$. Then the option holder can only exercise according to a stopping time with respect to $(\mathscr{G}_n, n \geq 0)$ where $\mathscr{G}_n = \sigma(X_i, i \leq n)$. Doing this, the price is approximated (by below) by

$$\sup_{\tau \in \mathscr{G}_{0,N}} \mathbb{E}\left(B(0,\tau)\,\varphi(X_\tau)\right) \tag{2.27}$$

where $\mathscr{G}_{0,N}$ is the set of the $\mathscr{G}_n$–stopping times taking values in $\{0, \ldots, N\}$.

To compute (2.27), the authors propose a quantization method for the one-dimensional dynamic programming algorithm. Obviously, in general, the process $(\varphi(X_n), n \geq 0)$ is not Markovian with respect to $\mathscr{G}$ and a second approximation has to be done in order to write the following dynamical programming algorithm assuming that $(\varphi(X_n), n \geq 0)$ is Markovian

$$\begin{cases} u(N, x) = \varphi(x) \\ u(j, x) = \max\left(\varphi(x), \hat{P}_j u(j+1, x)\right), \quad 1 \leq j \leq N, \end{cases} \tag{2.28}$$

where $\hat{P}_j(x, dy)$ is given by

$$\mathbb{E}\left(B(n, n+1) f\left(\varphi(X_{n+1})\right) | \varphi(X_n)\right) = \hat{P}_n f\left(\varphi(X_n)\right).$$

Assume that the two previous approximations are acceptable. The algorithm (2.28) can now be handled by a one–dimensional quantization technique. For $1 \leq j \leq N$, let

$$\left\{z^j_2 < \ldots < z^j_{n_j}\right\} \subset \mathbb{R}$$

and let $z^j_1 = -\infty$, $z^j_{n_j+1} = +\infty$. Then, define the sets $y^j = \left\{y^j_1, \ldots, y^j_{n_j}\right\}$ by

$$y^j_k = \mathbb{E}\left(\varphi\left(X_j\right) | \varphi\left(X_j\right) \in \left[z^j_k, z^j_{k+1}\right]\right). \tag{2.29}$$

As in the quantization algorithm, the kernel $\hat{P}_j$ is discretized on $y^j \times y^{j+1}$ by $\hat{\hat{P}}_j$ where

$$\hat{\hat{P}}_j\left(y^j_k, y^{j+1}_l\right) = B(j, j+1) \frac{\mathbb{P}\left(\varphi(X_j) \in [z^j_k, z^j_{k+1}], \varphi(X_{j+1}) \in [z^{j+1}_l, z^{j+1}_{l+1}]\right)}{\mathbb{P}\left(\varphi(X_j) \in [z^j_k, z^j_{k+1}]\right)} \tag{2.30}$$

The next step is to approximate (2.28) using $\widehat{P}_j\left(y_k^j, y_l^{j+1}\right)$,

$$\left\{\begin{array}{lcl} \hat{u}(N, y_k^N) & = & y_k^N, \quad 1 \le k \le n_N \\ \hat{u}(j, y_k^j) & = & \max\left(y_k^j, \sum_{1 \le l \le n_{j+1}} \hat{\hat{P}}_j\left(y_k^j, y_l^{j+1}\right) \hat{u}(j+1, y_l^{j+1})\right), \\ & & \qquad \text{For } 1 \le k \le n_j, 0 \le j \le N. \end{array}\right. \tag{2.31}$$

The approximation of the price at time 0 is given by $\hat{u}(0, x_0)$.

A Monte-Carlo version of (2.31) is obtain using m samples of the process X, $(X_j^{(m)}, 1 \le j \le N)$ and approximating

$$y_k^j \text{ with } \frac{1}{\text{Card }(A_k^j)} \sum_{m \in A_k^j} \varphi(X_j^{(m)}),$$

and

$$\hat{P}_j \text{with } \frac{\text{Card }(A_k^j \cap A_l^{j+1})}{\text{Card }(A_k^j)},$$

where

$$A_k^j = \left\{1 \le m \le n, \varphi(X_j^{(m)}) \in \left[z_k^j, z_{k+1}^j\right]\right\}.$$

Remark 2.6.1. Comparison of most of the algorithms presented here can be found in [Fu et al.(2001)].

Bibliographie

[Abramovitz and Stegun(1970)] M. Abramovitz and A. Stegun. *Handbook of Mathematical Functions*. Dover, 9th edition, 1970.

[Antonov and Saleev(1980)] I.A. Antonov and V.M. Saleev. An economic method of computing lp_τ-sequences. *USSR Comput. Maths. Math. Phys*, 19 :252–256, 1980.

[Barraquand and Martineau(1995)] J. Barraquand and D. Martineau. Numerical valuation of high dimensional multivariate american securities. *J. of Finance and Quantitative Analysis*, 30 :383–405, 1995.

[Broadie and Glassermann(1997a)] M. Broadie and P. Glassermann. Pricing american-style securities using simulation. *J.of Economic Dynamics and Control*, 21 :1323–1352, 1997a.

[Broadie and Glassermann(1997b)] M. Broadie and P. Glassermann. A stochastic mesh method for pricing high-dimensional american options. *Working Paper*, Columbia University :1–37, 1997b.

[Clément et al.(2002)] Emmanuelle Clément, Damien Lamberton, and Philip Protter. An analysis of a least squares regression method for American option pricing. *Finance Stoch.*, 6 (4) :449–471, 2002.

[Cochran(1953)] William G. Cochran. *Sampling techniques*. John Wiley & Sons, Inc., New York ; Chapman & Hall, Ltd., London, 1953.

[Cohort(2001)] P. Cohort. Monte–carlo methods for pricing american options. Documentation for the premia software, INRIA and ENPC, February 2001. see `http://cermics.enpc.fr/~premia`.

[Cohort(2004)] Pierre Cohort. Limit theorems for random normalized distortion. *Ann. Appl. Probab.*, 14(1) :118–143, 2004.

[Devroye(1986)] Luc Devroye. *Nonuniform random variate generation*. Springer-Verlag, New York, 1986.

[Duflo(1990)] Marie Duflo. *Méthodes récursives aléatoires*. Techniques Stochastiques. Masson, Paris, 1990.

[Duflo(1997)] Marie Duflo. *Random iterative models*, volume 34 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1997. Translated from the 1990 French original by Stephen S. Wilson and revised by the author.

[Faure(1981)] H. Faure. Discrépance de suites associées à un système de numération (en dimension 1). *Bull. Soc. Math. France*, 109 :143–182, 1981.

[Faure(1982)] H. Faure. Discrépance de suites associées à un système de numération (en dimension *s*). *Acta Arith.*, 41 :337–351, 1982.

[Fort and Pagès(1995)] J.C. Fort and G. Pagès. About the a.s. convergence of the Kohonen algorithm with a general neighborhood function. *The Annals of Applied Probability*, 5(4), 1995.

[Fox(1988)] B.L. Fox. Algorithm 659 :implementating Sobol's quasirandom sequence generator. *ACM Transaction on Mathematical Software*, 14(1) :88–100, 1988.

[Fox et al.(1992)Fox, Bratley, and Neiderreiter] B.L. Fox, P. Bratley, and H. Neiderreiter. Implementation and test of low discrepancy sequences. *ACM Trans. Model. Comput. Simul.*, 2(3) :195–213, July 1992.

[Fu et al.(2001)] M.C. Fu, S.B. Laprise, D.B. Madan, Y. Su, and R. Wu. Pricing american options : a comparison of monte carlo simulation approaches. *Journal of Computational Finance*, 4(3), Spring 2001.

[Glasserman et al.(1999)Glasserman, Heidelberger, and Shahabuddin] P. Glasserman, P. Heidelberger, and P. Shahabuddin. Asymptotically optimal importance sampling and stratification for pricing path dependent opions. *Mathematical Finance*, 9(2) :117–152, April 1999.

[Hall and Heyde(1980)] P. Hall and C. C. Heyde. *Martingale limit theory and its application*. Academic Press, Inc. [Harcourt Brace Jovanovich, Publishers], New York-London, 1980. Probability and Mathematical Statistics.

[Hammersley and Handscomb(1979)] J. Hammersley and D. Handscomb. *Monte Carlo Methods*. Chapman and Hall, London, 1979.

[Kalos and Whitlock(2008)] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods*. Wiley-Blackwell, Weinheim, second edition, 2008.

[Kasas(2000)] Frédéric Kasas. *Méthodes de Monte-Carlo et suites à discrépance faible appliquées au calcul d'options en finance*. PhD thesis, Université d'Evry, Juillet 2000.

[Knuth(1998)] Donald E. Knuth. *The art of computer programming. Vol. 2*. Addison-Wesley, Reading, MA, 1998. Seminumerical algorithms, Third edition [of MR0286318].

[L'Ecuyer(1990)] P. L'Ecuyer. Random numbers for simulation. *Communications of the ACM*, 33(10), Octobre 1990.

[Longstaff and Schwartz(1998)] F.A. Longstaff and E.S. Schwartz. Valuing american options by simulations :a simple least-squares approach. *Working Paper Anderson Graduate School of Management University of California*, 25, 1998.

[Major(2013)] Péter Major. Central limit theorems for martingales. Technical report, Mathematical Institute of the Hungarian Academy of Sciences, 2013. `http://www.math-inst.hu/~major/probability/martingale.pdf`.

[Niederreiter(1995)] H. Niederreiter. New developments in uniform pseudorandom number and vector generation. In P.J.-S. Shiue and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 87–120, Heidelberg, New York, 1995. Springer-Verlag.

[Niederreiter(1992)] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[Pagès and Bally(2000)] G. Pagès and V. Bally. A quantization algorithm for solving multi–dimensional optimal stopping problems. *Technical report 628, université Paris 6*, 2000.

[Pagès and Xiao(1997)] G. Pagès and Y.J. Xiao. Sequences with low discrepancy and pseudo random numbers : theoretical results and numerical tests. *J. Statist. Comput. Simul*, 56 : 163–188, 1997.

[Pagès et al.(2001)] G. Pagès, V. Bally, and J. Printems. A stochastic optimisation method for nonlinear problems. *Technical report number 02, University Paris XII*, 2001.

[Press et al.(1992)] W. H. Press, Saul A. Teukolsky, W. T. Vetterling, and Brian P. Flannery. *Numerical receipes in C*. Cambridge University Press, Cambridge, second edition, 1992. The art of scientific computing.

[Radovic et al.(1996)] Igor Radovic, Ilya M. Sobol', and Robert F. Tichy. Quasi-Monte Carlo methods for numerical integration : Comparison of different low discrepancy sequences. *Monte Carlo Methods Appl.*, 2(1) :1–14, 1996.

[Ripley(2006)] Brian D. Ripley. *Stochastic simulation*. Wiley Series in Probability and Statistics. Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2006. Reprint of the 1987 original, Wiley-Interscience Paperback Series.

[Roman(1992)] S. Roman. *Coding and information theory*. Number 134 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1992.

[Roth(1954)] K. F. Roth. On irregularities of distribution. *Mathematika*, 1 :73–79, 1954.

[Rubinstein(1981)] R.Y. Rubinstein. *Simulation and the Monte Carlo Method*. Wiley Series in Probabilities and Mathematical Statistics, 1981.

[Sabin and Gray(1986)] M.J. Sabin and R.M. Gray. Global convergence and empirical consistency of the generalised lloyd algorithm. *IEEE Transactions on Information Theory*, 32 : 148–155, March 1986.

[Sobol'(1967)] I.M. Sobol'. On the distribution of points in a cube and the approximate evaluation od integral. *USSR Computational Math. and Math. Phys.*, 7 :86–112, 1967.

[Sobol'(1976)] I.M. Sobol'. Uniformly distributed sequences with an additional unifrom property. *USSR Computational Math. and Math. Phys.*, pages 236–242, 1976.

[Tsitsiklis and Roy(2000)] J.N. Tsitsiklis and B. Van Roy. Regression methods for pricing complex american-style options. *Working Paper*, MIT :1–22, 2000.

[Williams(1991)] David Williams. *Probability with martingales*. Cambridge Mathematical Textbooks. Cambridge University Press, Cambridge, 1991.