CentraleSupélec | université PARIS-SACLAY

Pricing of American Options

ST7 - Financial Markets

David Abehsera, Samuel Moyal, Benjamin Tibi

Mars 2024

Outline

Introduction

This project embarks on a comprehensive exploration of valuation techniques for American options, which are financial derivatives that grant the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price before or at the expiration date. Unlike their European counterparts, American options offer the flexibility of early exercise, making their valuation a complex and intriguing challenge. This project aims to dissect and enhance existing methodologies, leveraging advanced computational techniques such as Monte Carlo simulations, least squares regression, and quantization.

Thus, calculating the prices of American options presents a formidable challenge, particularly when it involves multiple underlying assets. The core of this challenge lies in solving an optimal stopping problem, which, in the realm of classical diffusion models, aligns with a variational inequality. However, traditional partial differential equation (PDE) methods fall short in tackling this issue in more complex, multi-dimensional settings.
In response, a range of scholars have proposed numerical strategies that utilize Monte Carlo simulations, as highlighted in various studies [1,2]. These approaches begin by simplifying the continuous exercise timeframe of American options to a finite set of dates, effectively transforming them into Bermudan options. This simplification makes it relatively straightforward to manage the error introduced by limiting the exercise dates to discrete times, a point elaborated upon in certain analyses [3]. Our discussion will primarily focus on addressing the problem within these discrete time frames.
Addressing the discrete optimal stopping problem essentially boils down to efficiently applying the dynamic programming principle. The significant hurdle in utilizing Monte Carlo methods arises from the need to accurately estimate conditional expectations during the dynamic programming iterations. A promising solution to this challenge involves employing least squares regression on a select group of functions to approximate conditional expectations. This approach, initially mentioned in [4], has been a critical component in the advancements made in this field, notably in the influential works of Longstaff and Schwartz [1], and Tsitsiklis and Van Roy [2].

Another approach we will cover is the Quantization Tree Method for American Option pricing. This method, emerging from the intersection of probabilistic analysis and numerical optimization, offers a novel approach to tackling the high-dimensional and complex nature of American option pricing, especially when compared to traditional methods such as finite difference, binomial trees, and Monte Carlo simulations.
At its core, the quantization tree method leverages the concept of vector quantization, a process that approximates the continuous state space of underlying asset prices with a finite but optimally selected set of points. By doing so, it simplifies the computation of conditional expectations, a crucial step in the valuation of American options while preserving the accuracy of the results. This approach not only facilitates a more efficient computational process by reducing the dimensionality of the problem but also enhances the method's applicability to options with multiple underlying assets, where the curse of dimensionality renders many conventional methods impractical.
The introduction of the quantization tree method to the field of financial engineering is

attributed to several pioneering works. Among them, the studies by Bally and Pagès [5] stand out for their fundamental contributions to the theoretical underpinnings and practical implementations of the method. Their research demonstrated how the quantization of the asset price process could be effectively combined with the dynamic programming principle to yield accurate option pricing under complex financial models.
Furthermore, subsequent developments have refined and extended the quantization tree method, addressing various challenges such as the efficient generation of quantization grids and the implementation of the method in high-dimensional settings. These advancements have significantly broadened the scope and applicability of the method, as evidenced by its adoption in both academic research and industry practice for the pricing of American and exotic options under a variety of market conditions.

In this paper, we will then explore two models for pricing and elaborate on their mechanisms. Subsequently, we will conduct a comparative analysis of the outcomes obtained under certain parameters against the established benchmark in American options pricing : the Binomial Tree Model, originally developed by Cox, Ross, and Rubinstein [6].

1 First Model : Least Squares Regression Method for Pricing

1.1 Theoretical results

In the realm of probabilistic approximations, our initial strategy substitutes the continuous-time issue with a discrete-time optimal stopping challenge. This approach underpins the subsequent algorithm, which we will explore in a discrete temporal context.

Contemplate a probability space $(\Omega, \mathcal{F}, P)$, structured with a discrete filtration $\{\mathcal{F}_j\}_{j=0,..,L}$, where L symbolizes the finite horizon of discrete events. We examine a sequence of adapted stochastic processes $(Z_j)_{j=0,...L}$ where each Z_i for $i = 0, \ldots, L$ is a square-integrable random variable. The objective is to evaluate the supremum of the expected value of stopping times within this framework, which is :

$$\sup_{\tau \in \mathcal{T}_{0,L}} \mathbb{E}[Z_\tau],$$

where $\tau_{j,L}$ symbolize the collection of permissible stopping times, constrained to the set $\{j, \ldots, L\}$. The Snell envelope, $(U_j)_{j=0,...,L}$, of the adapted process $(Z_j)_{j=0,...,L}$ is encapsulated as follows :

$$U_j = \operatorname*{ess\,sup}_{\tau \in \tau_{j,L}} \mathbb{E}(Z_\tau | \mathcal{F}_j), \quad j = 0, \ldots, L.$$

Employing the dynamic programming paradigm, we articulate :

$$\begin{cases} U_L = Z_L \\ U_j = \max(Z_j, \mathbb{E}(U_{j+1} | \mathcal{F}_j)), \quad 0 \leq j < L. \end{cases}$$

Furthermore, $U_j = \mathbb{E}(Z_{\tau_j} | \mathcal{F}_j)$, and we identify the critical stopping time τ_j as the minimum index k such that $U_k = Z_k$: $\tau_j = \min\{k \geq j \,|\, U_k = Z_k\}$.
Our goal is thus to find U_0. To solve the problem, we chose to reformulate the problem in term of optimal stopping time and we go for a dynamical programming approach.
Firstly, by considering the underlying model as a Markov chain, we can assume that there is an $(\mathcal{F}_j)$-Markov chain $(X_j)_{j=0,...,L}$ with state space $(E, \mathcal{E})$ such that, for $j = 0, \ldots, L$,

$$Z_j = f(j, X_j),$$

for some Borel function $f(j, \cdot)$. We then have $U_j = V(j, X_j)$ for some function $V(j, \cdot)$; and $\mathbb{E}[Z_{\tau_{j+1}} | \mathcal{F}_j] = \mathbb{E}[Z_{\tau_{j+1}} | X_j]$. We will also assume that the initial state $X_0 = x$ is deterministic, so that U_0 is also deterministic.

Then, we can consider a dynamic programming approach by rewriting τ_j as :

$$\begin{cases} \tau_L = L \\ \tau_j = j\mathbf{1}_{\{Z_j \geq \mathbb{E}(Z_{\tau_{j+1}}|X_j)\}} + \tau_{j+1}\mathbf{1}_{\{Z_j < \mathbb{E}(Z_{\tau_{j+1}}|X_j)\}}, \quad 0 \leq j \leq L-1 \end{cases}$$

This approach will enable us to compute τ by back-propagation.
However, this recursive definition uses a conditional expectancy that we don't know the expression. The first main idea of our paper is to approximate it by using an orthogonal projection of $\mathbb{E}(Z_{\tau_{j+1}}|X_j)$ on a finite dimension space generated by functions of X_j. We therefore consider a a sequence $(e_k(x))_{k\geq 1}$ of measurable real-valued functions defined on E and satisfying the following conditions :

- **1** : For $j = 1$ to $L-1$, the sequence $(e_k(X_j))_{k\geq 1}$ is total in $L^2(\sigma(X_j))$.
- **2** : For $j = 1$ to $L-1$ and $m \geq 1$, if $\sum_{k=1}^m \lambda_k e_k(X_j) = 0$ a.s., then $\lambda_k = 0$ for $k = 1$ to m.

We can fix the dimension of our projection space to m and for $j = 1$ to $L-1$ and denote P_j^m the orthogonal projection from $L^2(\Omega)$ to $\mathbf{Span}\,(e_1(X_j), \ldots, e_m(X_j))$. We will note e^m the vector valued function (e_1,...,e_m).
Moreover, as P_j^m is an orthogonal projection on a subspace of the space of $\mathcal{F}_j$-measurable random variables we obtain this main result :

$$P_j^m\left(Z_{\tau_{j+1}}\right) = P_j^m\left(\mathbb{E}\left(Z_{\tau_{j+1}} \mid \mathcal{F}_j\right)\right),$$

Thanks to this equation,we just have to project $Z_{\tau_{j+1}^{[m]}}$ instead of the conditional expectancy. The advantage of this result is that we know Z_j so we will be able to find its orthogonal projection contrary to the conditional expectancy.
By noting $\tau_j^{[m]}$ the approximation of τ_j by replacing the conditional expectancy by its projection in the definition of τ, the approximation of the value function can be defined as :

$$U_0^m = \max(\, Z_0,\, \mathbb{E}[Z_{\tau_1^{[m]}}]\,).$$

The last part of this approach is to evaluate numerically $\mathbb{E}[Z_{\tau_1^{[m]}}]$ by using a Monte-Carlo procedure : We simulate N independent processes $(X_j^{(1)}), \ldots, (X_j^{(N)})$ of the underlying and their respective payoffs $(Z_j^{(1)}), \ldots, (Z_j^{(N)})$.
For each path and each step, we define a stopping time $\tau_j^{n,m,N}$ that we estimate by using the least square estimator :

$$\alpha_j^{(m,N)} = \arg\min_{a\in\mathbb{R}^m} \sum_{n=1}^{N} \left(Z^{(n)}_{\tau_{j+1}^{n,m,N}} - a \cdot e^m\left(X_j^{(n)}\right)\right)^2 \tag{1}$$

Then, $\alpha_j^{(m,N)} \cdot e^m\left(X_j^{(n)}\right)$ is an approximation of $P_j^m\left(Z_{\tau_{j+1}^{n,m,N}}\right)$ and the stopping times of each path are therefore defined recursively by :

$$\begin{cases} \tau_L^{n,m,N} = L \\ \tau_j^{n,m,N} = j\mathbf{1}_{\left\{Z_j^{(n)} \geq \alpha_j^{(m,N)} \cdot e^m\left(X_j^{(n)}\right)\right\}} \\ \qquad\qquad + \tau_{j+1}^{n,m,N}\mathbf{1}_{\left\{Z_j^{(n)} < \alpha_j^{(m,N)} \cdot e^m\left(X_j^{(n)}\right)\right\}}, \end{cases} \quad , 1 \leq j \leq L-1$$

The price of our American option is finally given by :

$$U_0^{m,N} = \max\left(Z_0, \frac{1}{N}\sum_{n=1}^{N} Z^{(n)}_{\tau_1^{n,m,N}}\right) \tag{2}$$

1.2 Implementation

Thanks to these results we are finally able to compute the price of our an American option. In our simulation, we have made a few assumptions :

- **Time horizon** : $L = 10$
- **Number of Monte-Carlo simulations** : $N = 2500$
- **Dimension of the projection space** : $m = 30$
- **Total basis** : $\forall k \in \{0, \dots, m-1\}, e_k(x) = x^k$ (we use a polynomial approximation which is relevant in our model)
- **Initial price of the underlying** : $X_0 = 100$
- **Modeling of the underlying** : $\forall j \in \{1, \dots, J\}, X_j = X \exp(\sigma W_j - \frac{\sigma^2}{2} j)$, with W a Brownian motion.
- **Interest rate and dividends** : $r = 0, d = 0$

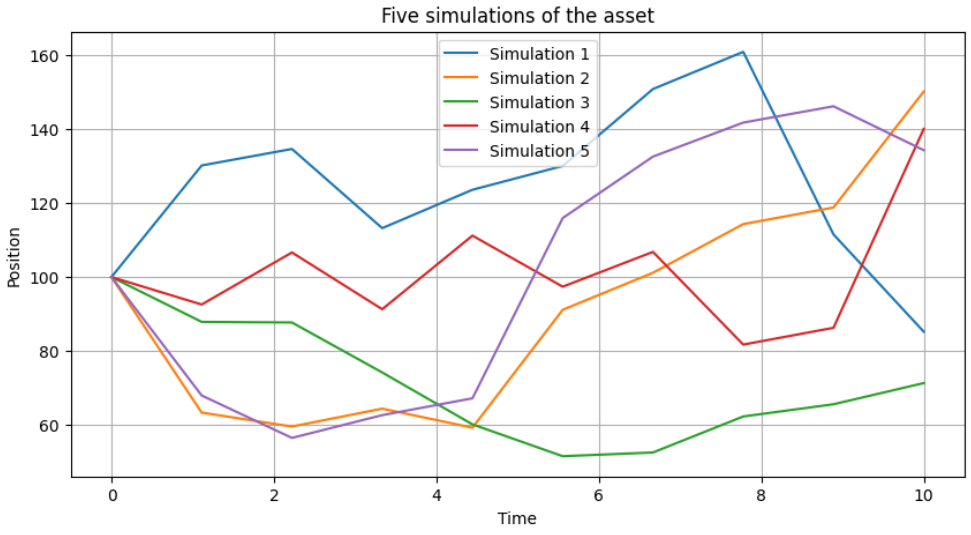

FIGURE 1 – Five simulations of X for $\sigma = 0.2$

For instance, for $\sigma = 0.2$, we obtain this kind of simualtion for our underlying X. With these parameters, our algorithm requires between 2 and 4 minutes to price an American option. We will evaluate the accuracy and the relevance of our model in the next subsections.

1.2.1 Call options

In order to evaluate our model, we have priced call options of different values of strike K and different volatilities σ.
In the framework of a call option, the payoff function is given by :

$$\forall j \in \{0, \dots, J\}, Z_j = (X_j - K)^+$$

By fixing $\sigma = 0.2$ and for 10 values of K evenly spread between 75 and 95, we find these results :

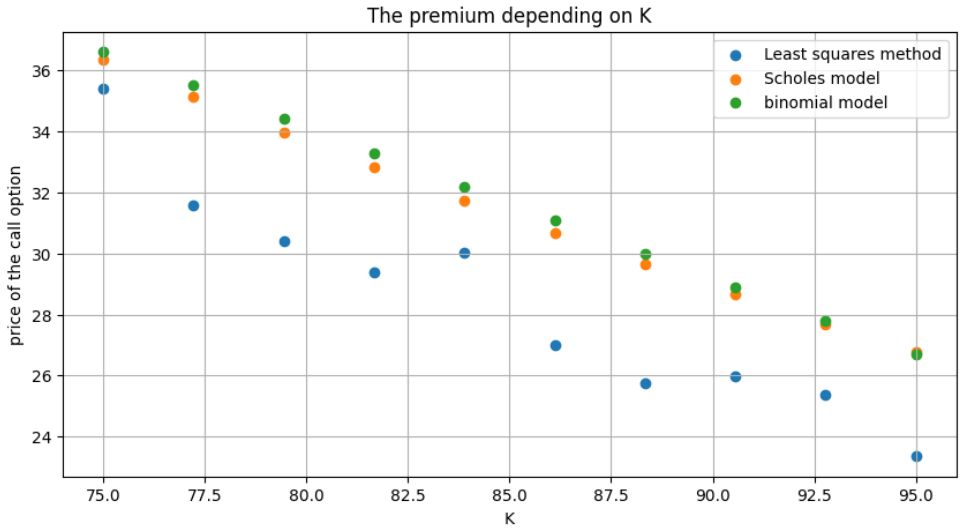

FIGURE 2 – Call price for $\sigma = 0.2$ and 10 values of K between 75 and 95

And by fixing K to 85 and for 10 values of σ evenly spread between 0.05 and 0.25, we obtain these results :

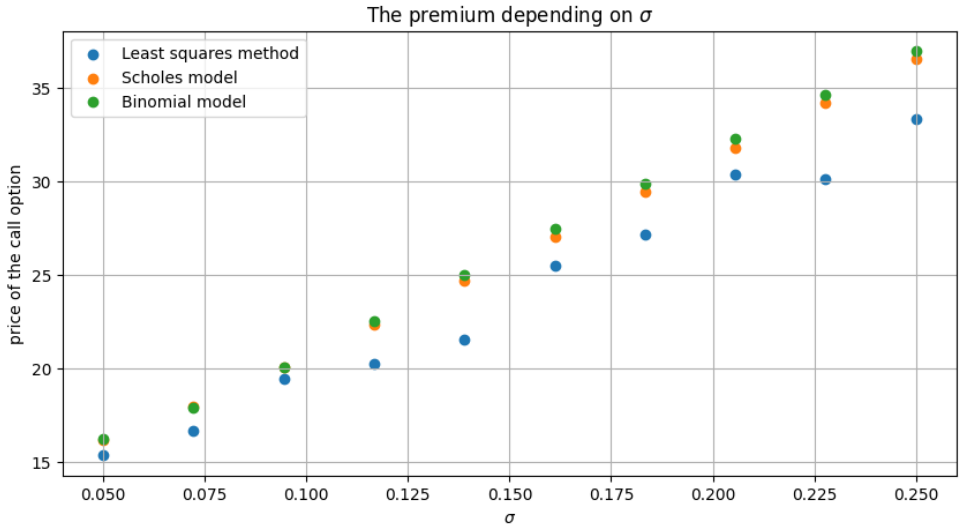

FIGURE 3 – Call price for $K = 85$ and 10 values of σ between 0.05 and 0.25

As we can see, the binomial and the Black-Scholes models provide almost the same price. It was predictible because, as we saw in our course, in the case of a call option with 0 dividends, the two call options have the same value.
However, our model gives a lower value than the two other models. The first reason is that we have approximated our American option by a Bermudan option. Since we can no longer exercise the call at each time but only at certain times, its value will be lower than its real value. The second reason and ,as we shall see next, will be discussed later in this paper.

1.3 Put options

Then, we have done the same work for put options (of payoff $Z_j = (K - X_j)^+$). The goal of this study was to find out if the only reason of our underpricing was the Bermudan option approximation or if there was another reason. If this approximation was the only reason, we should find a model that prices the American put option between the binomial model and the Black-Scholes model. Indeed, even with the approximation mentioned above, the Bermudan put option should be more expensive than the European one. Indeed, an American put option is more expensive than an European. However its value should (or could without being a mistake) be lower than the value given by the Binomial model because of the bermudan option approximation. Then, we should logically have :

$$U_0^{B-S(European)} \leq U_0^{m,N} \leq U_0^{Binomial}$$

By fixing $\sigma = 0.2$ and for 10 values of K evenly spread between 85 and 120, we find these results :

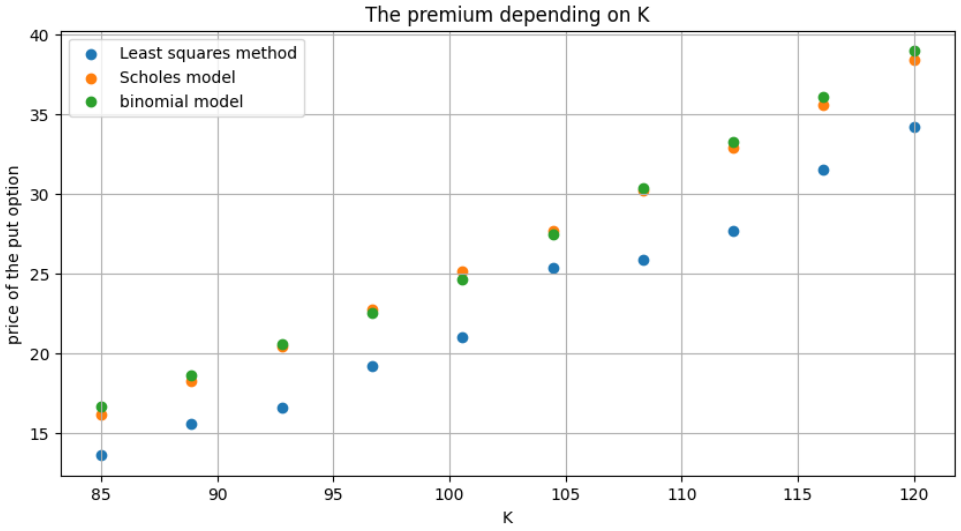

FIGURE 4 – Put price for $\sigma = 0.2$ and 10 values of K between 85 and 120

And by fixing K to 85 and for 10 values of σ evenly spread between 0.05 and 0.25, we obtain these results :

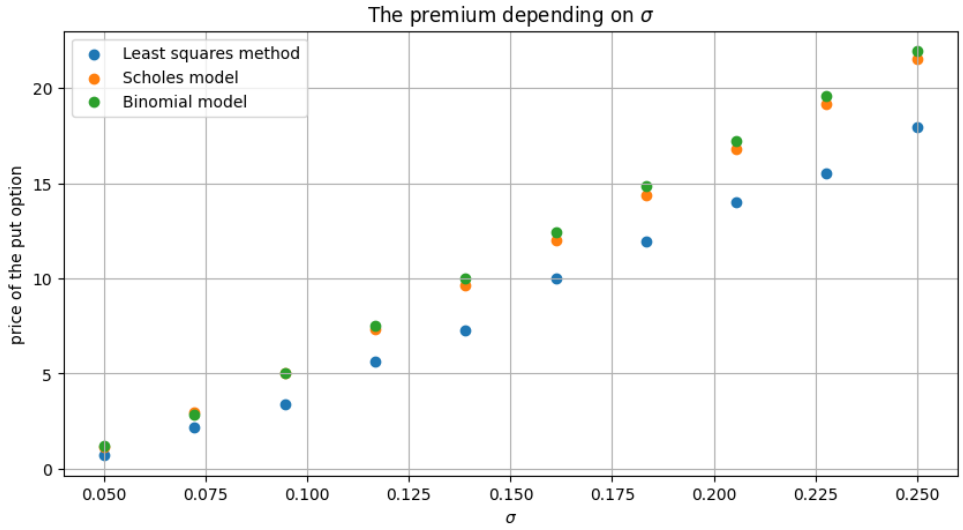

FIGURE 5 – Put price for $K = 85$ and 10 values of σ between 0.05 and 0.25

As we can see, the results are not the ones that we expected. The main reason in reality is that the speed of convergence of our algorithm with a polynomial basis is too low. We could not increase the dimension of our projection space because x^j increases too fast with j so Python was displaying an overflow error with a higher dimension. Thus, we were not able to increase the dimension of our projection space to converge to the right value. The convergence error could not come from the number of Monte-Carlo simulations because we could do as much simulations that as we wanted (the only constraint for the number of simulations was the time).
The solutions to this problem could be to consider an other family of functions that provide a faster convergence or that takes longer to overflow.

2 Second Model : Quantization Tree Method for Pricing

2.1 Theory

From now on, we present another approach based on the discretization of continuous random variables. Recall the system of equations we are interested about :

$$\begin{cases} U_L = Z(t_L, X_L), \\ U_k = \max(Z(t_k, X_k), \mathbb{E}[U_{k+1}|X_k]), \quad 0 \leq k \leq L-1. \end{cases} \tag{3}$$

where h is a known measurable function, S represents the prices of the underlying, V the option's price, and L the time horizon. The difficulty here is the same as before, to estimate the conditional expected value, to reach our goal which is to determine U_0. We will do it using the discretization of continuous random variables.

The starting point of the method is to discretize the random variables X_k by some $\sigma(X_k)$-measurable random variables $\hat{X}_k$ taking finitely many values in $\mathbb{R}^d$. Such a random vector $\hat{X}_k$ is called a *quantization* of X_k. Equivalently, one may define a quantization of X_k by setting $\hat{X}_k = q_k(X_k)$ where $q_k : \mathbb{R}^d \to \mathbb{R}^d$ is a Borel "quantizing" function such that $|q_k(\mathbb{R}^d)| = N_k < +\infty$. The elements of the set $X_k(\Omega)$ are called *elementary quantizers*. Let $N = N_0 + N_1 + \ldots + N_L$, denote the total number of elementary quantizers.

We assume in this section that for every $k \in \{0, 1, \ldots, L\}$ we have access to a sequence of quantizations $\hat{X}_k = q_k(X_k)$, $k = 0, \ldots, L$, of the Markov chain $(X_k)_k$. We denote by $\{x_1^k, \ldots, x_{N_k}^k\} = q_k(\mathbb{R}^d)$ the grid of N_k points used to quantize X_k and by $x^* = (x^1, \ldots, x^L)$ the induced N-tuple. Since we discretized X, we do the same for Z.

The quantized dynamic programming formula below is devised by analogy with the original one :

$$\begin{cases} \hat{U}_n = Z(t_n, \hat{X}_n), \\ \hat{U}_k = \max(Z(t_k, \hat{X}_k), \mathbb{E}[\hat{U}_{k+1}|\hat{X}_k]), \quad 0 \leq k \leq L-1. \end{cases} \tag{4}$$

Now, it is possible to compute the conditional expectation. In order to do so, we will define :

- $x^k = (x_1^k, \ldots, x_{N_k}^k)$ for all $0 \leq k \leq L$.
- For every i in 1,...,N_k : $p_k^i := P(\hat{X}_k = s_k^i) = P(X_k \in C_i(x^k))$, where $C_i(x^k)$ is some interval around x^k.
- $\pi_{ij}^k := P(\hat{X}_{k+1} = x_j^{k+1} | \hat{X}_k = x_i^k) = P(X_{k+1} \in C_j(x^{k+1}) | X_k \in C_i(x^k))$, the transition matrix.

Thus, we can rewrite the system and we get the following one :

$$\begin{cases} \hat{u}_L(x_i^L) := v_L(x_i^L), \quad i \in \{1, \ldots, N_L\}, \\ \hat{u}_k(x_i^k) := \max\left(v_k(t_k, x_i^k), \sum_{j=1}^{N_{k+1}} \pi_{ij}^k \hat{u}_{k+1}(x_j^{k+1})\right), \quad 1 \leq i \leq N_k, 0 \leq k \leq L-1. \end{cases} \tag{5}$$

In what follows, we aim to calculate the solution of this system, which is when $k = 0$.

2.2 Implementation

2.2.1 Quantization

We noticed that X_t follows a log-normal law :

$$f_{X_t}(x) = \frac{1}{x\sigma\sqrt{2\pi t}} \exp\left(-\frac{(\ln(x)-(ln(X_0)-\frac{\sigma^2 t}{2})^2}{2\sigma^2 t}\right), \quad x > 0$$

The discretization points chosen are the quantiles of the law. This choice is motivated by the fact that such points are easily computable thanks to their relation with the quantiles of the normal law, available in python libraries.

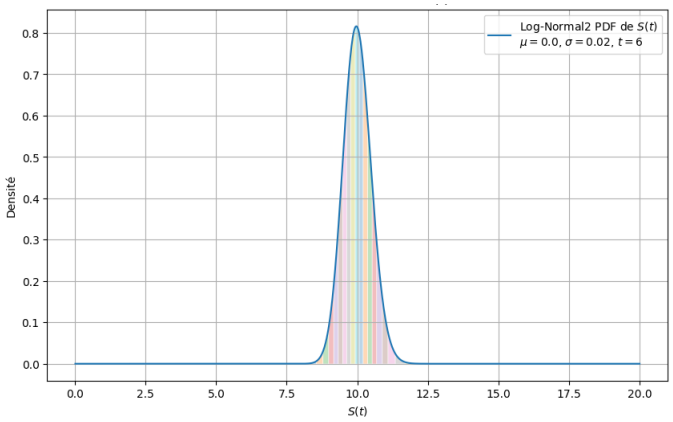

FIGURE 6 – discretization of X_t for $t = 6$

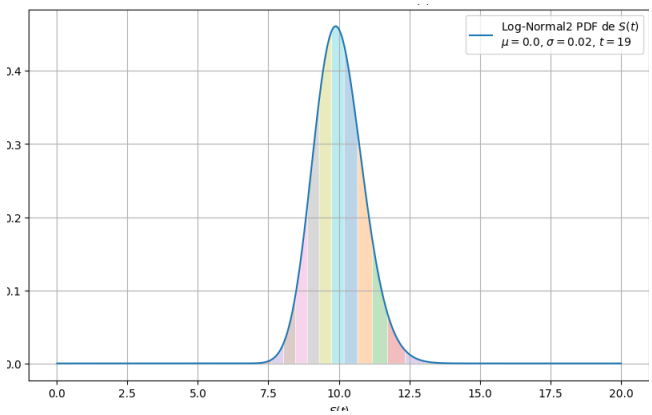

FIGURE 7 – discretization of X_t for $t = 19$

We see that the density function of X_t spreads around the initial value X_0. The variance increases as t increases.

The discretization points are chosen so that the probability to be associated to a certain point is roughly uniform.

2.2.2 Transformation function

The transformation function giving the connection between the law of the discrete random variable and the continuous one is given by :

$$P(X_t^* = x_i^t) = \int_{\frac{x_i+x_{i-1}}{2}}^{\frac{x_i+x_{i+1}}{2}} f_{X_t}(x)\,dx$$

2.2.3 Transformation matrix and change of variable

We need to compute the coefficients of the transition matrix :

$$\pi_{ij} = P(X_{t+1}^* = x_j^* | X_t^* = x_i^*) = \frac{p_{ij}}{p_i}$$

With $p_{ij} = \int_{a_{t+1}}^{b_{t+1}} \int_{a_t}^{b_t} f_{X_t,X_{t+1}}(x_t, x_{t+1})\,dx_t\,dx_{t+1}$and $p_i = \int_{a_t}^{b_t} f_{X_t}(x_t)\,dx_t$

As stressed before X_t follows a log-normal law. Then p_i is computable thanks to the libraries of python or by using Monte Carlo to compute integrales whose density function is known. p_{ij} seems more complex because X_t and X_{t+1} are correlated . But we notice that$\frac{X_{t+1}}{X_t}$ and X_t are not. Indeed :

$$\frac{X_{t+1}}{X_t} = e^{-\frac{1}{2}\sigma^2+\sigma(W_{t+1}-W_t)}$$

with $W_{t+1} - W_t$ following a normal law. Thus $\frac{X_{t+1}}{X_t}$ follows a log-normal law :

$$f_{\frac{X_{t+1}}{X_t}}(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln(x)+\frac{\sigma^2}{2})^2}{2\sigma^2}\right), \quad x > 0$$

By doing the change of variable : $(u, v) = (\frac{x_{t+1}}{x_t}, x_t)$ we have :

$$p_{ij} = \int_{a_t}^{b_t} \int_{\frac{a_{t+1}}{v}}^{\frac{b_{t+1}}{v}} v f_{\frac{X_{t+1}}{X_t}}(u) f_{X_t}(v)\,du\,dv \quad \text{which is computable on python}$$

Unfortunately, the results were unexpected and inconsistent with the logic of market pricing. It must be due to the fact that our quantization points are indeed easy to compute but are not optimal. A more precise choice of the quantization points (minimisation of the L_p norm of the error on the transformation at quantization points fixed ; then minimisation of the error at transformation fixed to find the optimal quantization points)

Conclusion

In conclusion, this paper has addressed the complex problem of pricing American options, a challenge that necessitates a careful consideration of optimal stopping theory and the Snell envelope. The Snell envelope plays a crucial role in determining the option's stopping policy, which directly influences the option's value. We have shown that the key to this valuation lies in the accurate approximation of the conditional expectations of the payoff process.

Our study proposed two distinct approximation methods : linear regression and quantization. Through linear regression, we employed a straightforward approach to project the payoff onto a set of basis functions, thus capturing the essence of the conditional expectations within a regression framework. This method proved not only intuitive but also computationally efficient.

On the other hand, the quantization method offered a novel perspective. By discretizing the state space, we transformed the problem into one that could be handled by dynamic programming techniques. Quantization provided a more granular approach, which, while potentially more computationally demanding, offered a higher degree of precision and robustness against model assumptions.

Both methods have their merits and limitations, and the choice between them may depend on the specific requirements of the problem at hand. For scenarios where computational resources are limited, linear regression may be the preferable choice. Conversely, for situations where accuracy is paramount, and computational complexity is less of a concern, quantization demonstrates clear advantages.

Future research may focus on combining these methods to exploit the efficiency of linear regression and the accuracy of quantization. Additionally, exploring other basis functions for the regression or refining the quantization grid could further enhance the precision of American option pricing models.

In the opening to future explorations, we could consider the impact of dividends and interest rates on our model, which has so far assumed a constant volatility framework. Dividends and varying interest rates play a significant role in option pricing, influencing the underlying asset's expected returns and, consequently, the pricing of derivatives. Their inclusion could offer a more holistic view of option valuation and risk assessment.

Ultimately, the assumption of constant volatility is often a simplification that overlooks the complexities of equity markets. Real-world equity assets exhibit volatility skew—a phenomenon where the implied volatility varies for different strike prices and maturities. By incorporating volatility skew into our model, we can capture the intricacies of market sentiment and the subtleties of option pricing more accurately. Adjusting for these factors would potentially refine the robustness of our pricing approach, aligning it closer to market realities and providing a more nuanced tool for traders and analysts in the dynamic landscape of financial markets.

References

1. Longstaff, F.A., Schwartz, E.S. : Valuing American options by simulation : a simple least-squares approach. Rev. Financial Stud. 14, 113–148 (2001)

2. Tsitsiklis, J.N., Van Roy, B. : Regression methods for pricing complex American-style options. Preprint, August 2000. IEEE Transactions on Neural Networks (forthcoming)

3. Lamberton, D., Brownian optimal stopping and random walks. Appl. Math. Optimization (forthcoming)

4. Carriere, J. : Valuation of Early-Exercice Price of options Using Simulations and Nonparametric ‘ Regression. Insurance : Math. Econ. 19, 19–30 (1996)

5.BALLY,V.,G.PAGE‘S,andJ.PRINTEMS(2003) :FirstOrderSchemesintheNumerical-Quantization Method, Math. Finance 13(1), 1–16.

6. Cox, John C., Stephen A. Ross, and Mark Rubinstein. 1979. "Option Pricing : A Simplified Approach." Journal of Financial Economics 7 : 229-263.