

PARIS-SACLAY UNIVERSITY
MASTER 2 DATA SCIENCE



Project 2

Pricing American Options

Realisé par :

Zoueïn Tia

Domenzain Maria Jose

Zaari Abdelouahab

SUPERVISÉ PAR :

DR.KEBAIER AHMED

Anée : 2022 - 2023

Contents

1	Abstract	2
2	Introduction	3
3	Les options américaines	3
3.1	Prix d'une option américaine	4
4	Les options Bermudiennes	5
5	Mouvement brownien géométrique	5
6	Algorithme de Longstaff-Schwartz	6
6.1	Description de la méthode Monte Carlo	6
6.2	Description de l'algorithme	7
7	Les réseaux de neurones	8
7.1	Description des modèles	9
7.1.1	Fonction Bermudan-NN-simple:	9
7.1.2	Fonction Bermudean-Put-MC-NN	11
7.2	Les résultats analytiques	12
7.2.1	Résultat de Bermudan-NN-simple	12
7.2.2	Résultat de Bermudean-Put-MC-NN	13
8	Conclusion	18

List of Figures

1	Extrait du code de la première fonction	9
2	Extrait du code de la deuxième fonction	11
3	la fonction Loss	12
4	la fonction Loss avec $N = 10000$	13
5	l'évolution du prix de l'option bermudienne	14
6	Extrait du résultat	14
7	la fonction Loss avec $N = 100000$	15
8	l'évolution du prix de l'option bermudienne	16
9	Extrait du résultat	16

Remerciements :

Nous tenons à exprimer notre gratitude envers Dr. Kebaier Ahmed, notre professeur, pour son soutien et ses conseils tout au long de ce projet. Sa passion pour le domaine de la finance et son engagement envers l'excellence académique ont été une source d'inspiration pour nous. Ses commentaires constructifs et ses encouragements ont grandement contribué à la réussite de ce projet. Nous sommes reconnaissants de pouvoir compter sur son expertise et son mentorat pour nous guider dans notre parcours académique et professionnel. Merci, pour votre dévouement envers l'éducation et votre impact positif sur nos vies.

Nous tenons également à remercier notre professeur pour les cours passionnants et informatifs qu'il nous a donnés tout au long du semestre. Grâce à ses connaissances approfondies et à son expérience pratique dans le domaine de la finance, nous avons acquis une compréhension plus approfondie des concepts clés et des techniques avancées utilisées dans l'évaluation des produits financiers. Nous avons particulièrement apprécié sa capacité à rendre les sujets complexes plus accessibles en utilisant des exemples concrets et en encourageant la participation en classe. Ses cours ont été une source d'inspiration pour nous et nous ont permis de développer nos compétences et notre confiance dans le domaine de la finance. Nous sommes reconnaissants de pouvoir compter sur son expertise et son mentorat pour nous guider dans notre parcours académique et professionnel.

1 Abstract

The determination of the fair price of an American option, which gives the holder the right to exercise the option at any time before its expiration date, is a challenging problem in the field of financial mathematics. In this project, our objective is to compare and apply pricing methods for American options.

We first provide a detailed description of the key elements of a classic option, including the type of option, the underlying asset, the quantity of the underlying asset, the expiration date, and the strike price. We then discuss the pricing and hedging problems associated with selling options and present the theoretical framework for pricing American options.

We implement the above-mentioned pricing methods for a range of American options with varying strike prices, underlying assets, and expiration dates. We compare the accuracy and efficiency of these methods.

Key words: *American option, pricing methods, classic option, underlying asset, pricing problem, hedging problem*

2 Introduction

Dans ce projet, notre objectif est de déterminer le prix de l'option américaine en utilisant des méthodes de pricing appropriées.

La description précise d'une option classique se fait à partir des éléments suivants:

- La nature de l'option : on parle, suivant la terminologie anglo-saxonne, de *call* pour une option d'achat et de *put* pour une option de vente.
- L'actif sous-jacent (*underlying*), sur lequel porte l'option : dans la pratique, il peut s'agir d'une action, d'une obligation, d'une devise, d'un taux, etc.
- Le montant, c'est-à-dire la quantité d'actif sous-jacent 'à acheter ou 'à vendre.
- L'échéance ou date d'expiration, qui limite la durée de vie de l'option ; si l'option peut être exercée à n'importe quel instant précédent l'échéance, on parle d'option américaine, si l'option ne peut être exercée qu'à l'échéance, on parle d'option européenne.
- Le prix d'exercice (*strike*), qui est le prix (fixé d'avance) auquel se fait la transaction en cas d'exercice de l'option.

Au moment de la vente de l'option, qu'on prendra pour origine des temps, le cours S_T est inconnu et deux questions se posent.

1. Combien faire payer à l'acheteur de l'option, autrement dit, comment évaluer à l'instant $t = 0$, une richesse $(S_T - K)_+$ disponible à la date T ? C'est le problème du pricing.
2. Comment le vendeur, qui touche la prime à l'instant 0, parviendra-t-il à produire la richesse $(S_T - K)_+$ à la date T ? C'est le problème de la couverture.

3 Les options américaines

Les options américaines sont un type d'option financière qui permettent à son détenteur d'exercer l'option à tout moment avant la date d'expiration de l'option, contrairement aux options européennes qui ne peuvent être exercées qu'à la date d'expiration. En d'autres termes, l'acheteur d'une option américaine peut décider d'exercer son option à tout moment avant la date d'expiration, ce qui lui confère une plus grande flexibilité.

Les options américaines sont souvent utilisées pour gérer le risque de marché, car elles permettent aux investisseurs de s'adapter rapidement aux fluctuations des prix des actifs sous-jacents. Les options américaines sont également plus complexes que les options européennes car elles nécessitent une évaluation plus fréquente du moment opportun pour exercer l'option.

3.1 Prix d'une option américaine

Le prix d'une option américaine dépend de plusieurs facteurs, notamment le prix de l'actif sous-jacent, le taux d'intérêt sans risque, la volatilité du marché et le temps restant jusqu'à l'expiration. Les méthodes d'évaluation des options américaines sont plus complexes que celles des options européennes, car il est nécessaire de prendre en compte la possibilité d'exercice anticipé de l'option à tout moment avant l'expiration.

Donc, comme dit avant, une option américaine peut être exercée à n'importe quel instant entre 0 et N . Elle (son profit) sera définie comme un processus (Z_n) positif et adapté à la filtration (F_n) . Z_n représente le profit que permet l'exercice de l'option au temps n .

Dans le cas d'un call américain sur une unité d'actif 1 au prix d'exercice K , on aura $Z_n = (S_n^1 - K)^+$; dans le cas d'un put américain on a $Z_n = (K - S_n^1)^+$.

Pour définir la valeur de l'option américaine associée au processus $(Z_n)_{\{0 \leq n \leq N\}}$, il faut raisonner par récurrence rétrograde en partant de l'échéance N .

A l'instant N , il est clair que la valeur est $U_N = Z_N$.

A quel prix vendre l'option à l'instant $N-1$? Si l'acheteur exerce immédiatement, il fera le profit Z_{N-1} ; sinon il exercera (éventuellement) à la date N et le vendeur doit être prêt à payer la richesse Z_N à l'instant N . Le vendeur doit donc disposer à l'instant $N-1$, d'une somme:

- au moins égale à Z_{N-1} ,
- lui permettant de fournir la richesse Z_N à l'instant N .

La somme qui, disponible à l'instant $N-1$, permet d'obtenir la richesse Z_N à l'instant N , c'est la valeur à l'instant $N-1$, d'une stratégie admissible de valeur finale Z_N , c'est-à-dire $S_{N-1}^0 E^*(\tilde{Z}_N | F_{N-1})$ avec $\tilde{Z}_N = \frac{Z_N}{S_N^0}$.

Il est donc naturel de prendre pour valeur de l'option américaine à l'instant $N-1$ la quantité

$$U_{N-1} = \max \left\{ Z_{N-1}, S_0^{N-1} E^* \left(\frac{S_0^N}{U_N} \middle| F_{N-1} \right) \right\} \quad (1)$$

De proche en proche on définit la valeur de l'option américaine à l'instant n par la relation de récurrence suivante valable pour $n = 1, \dots, N$:

$$U_{n-1} = \max \left\{ Z_{n-1}, S_0^{n-1} E^* \left(\frac{S_0^n}{U_n} \middle| F_{n-1} \right) \right\} \quad (2)$$

Lorsque l'actif sans risque évolue avec un taux d'intérêt constant et égal à r sur chaque période, i.e. $S_0^n = (1+r)^n$, la relation précédente devient:

$$U_{n-1} = \max \left\{ Z_{n-1}, \frac{1}{1+r} E^* \left(U_n \middle| F_{n-1} \right) \right\} \quad (3)$$

Soit $\tilde{U}_n = U_n / S_0^n$, la valeur actualisée de l'option américaine. La relation précédente devient:

$$\tilde{U}_{n-1} = \max(\tilde{Z}_{n-1}, \mathbb{E}^*[\tilde{U}_n | \mathcal{F}_{n-1}]) \quad (4)$$

4 Les options Bermudiennes

Une variante des options de style américain, qui permettent aux détenteurs d'exercer tôt à tout moment, les options bermudiennes permettent aux investisseurs d'acheter ou de vendre un titre ou un actif sous-jacent à un prix prédéfini à des dates spécifiques ainsi qu'à la date d'expiration de l'option.

Contrairement aux options américaines, les options bermudiennes ne peuvent être exercées qu'à des dates spécifiques prédéfinies, ce qui en fait un compromis entre les options américaines et européennes.

- Dans ce contexte, considérons une subdivision discrète de $[0, T]$ telle que $0 = t_0 < t_1 < \dots < t_n = T$.
- Une option bermudienne donne le droit à l'acheteur d'exercer à n'importe quelle date t_0, \dots, t_n et de payer $f(S_{t_k})$ au temps t_k .
- Nous notons $(\tilde{V}_t)_{0 \leq t \leq T}$ le portefeuille de couverture associé.
- À la date d'expiration $T = t_n$ nous avons $V_{t_n} = f(S_{t_n})$
- À la date $T = t_{n-1}$ nous avons

$$V_{t_{n-1}} = \max(f(S_{t_{n-1}}); e^{-r(t_n - t_{n-1})} \mathbb{E}^*[f(S_{t_n}) | \mathcal{F}_{t_{n-1}}]) \quad (5)$$

$$= \max(f(S_{t_{n-1}}); e^{-r(t_n - t_{n-1})} \mathbb{E}^*[V_{t_n} | \mathcal{F}_{t_{n-1}}]) \quad (6)$$

- Ainsi, on obtient $\forall k \in \{0, \dots, n-1\}$

$$\begin{cases} V_T = f(S_T) \\ V_{t_k} = \max(f(S_{t_k}); e^{-r(t_{k+1} - t_k)} \mathbb{E}^*[V_{t_{k+1}} | \mathcal{F}_{t_k}]) \end{cases}$$

5 Mouvement brownien géométrique

L'une des hypothèses essentielles du modèle de Black-Scholes est que l'actif sous-jacent, le plus souvent une action, est modélisé comme un mouvement brownien géométrique. Tout d'abord, nous devons définir un mouvement brownien standard.

Un processus aléatoire $\{W(t)\}$ est dit être un mouvement brownien standard sur $[0, T]$ s'il satisfait les conditions suivantes :

- $W(0) = 0$.
- $\{W(t)\}$ a des incréments indépendants et stationnaires.
- $W(t) - W(s) \sim N(0, t - s)$ pour tout $0 \leq s \leq t \leq T$.
- $\{W(t)\}$ a des trajectoires presque sûrement continues.

Il découle de cette définition que $W(t) \sim N(0, t)$ pour $0 \leq t \leq T$ ce qui est important lors de la simulation du mouvement brownien.

6 Algorithme de Longstaff-Schwartz

L'algorithme de Longstaff-Schwartz est une méthode de simulation de Monte Carlo qui permet de tarifier des options américaines et bermudiennes. Cette méthode repose sur la simulation d'un grand nombre de trajectoires du prix de l'actif sous-jacent et l'utilisation d'une régression polynomiale pour estimer la valeur de l'option à chaque date d'exercice anticipée. L'algorithme de Longstaff-Schwartz est particulièrement utile pour les options qui ne peuvent pas être tarifées de manière analytique.

6.1 Description de la méthode Monte Carlo

l'algorithme Longstaff-Schwartz utilise Monte Carlo pour estimer la valeur de l'option. L'idée générale des méthodes MC est de simuler un échantillon et en prenant la moyenne pour trouver la "vraie" valeur. Mathématiquement, MC est souvent décrit comme une méthode pour estimer la valeur d'un intégral:

$$\mu = \int f(x)dx \quad (7)$$

factorisons la fonction $f(x)$:

$$f(x) = h(x)p(x) \quad (8)$$

où $p(x)$ est une certaine densité, l'intégrale peut être interprétée comme l'espérance valeur

$$\mu = E[h(x)] = \int h(x)p(x)dx \quad (9)$$

En générant un i.i.d. échantillon x_1, x_2, \dots, x_n de $p(x)$, la valeur attendue peut être estimée comme

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n h(x_i) \quad (10)$$

Sachant que $h(x)$ est intégrable, la loi des grands nombres dit que $\hat{\mu} \xrightarrow{p} \mu$ a.s quand $n \rightarrow \infty$ et si $h(x)$ est aussi de carré intégrable et on définit

$$\sigma^2 = \text{Var}[h(x)] = \int (h(x) - \mu)^2 p(x)dx$$

il dit aussi que

$$\frac{\sqrt{n}(\hat{\mu} - \mu)}{\sigma} \rightarrow \mathcal{N}(0, 1)$$

Cela signifie que l'erreur $\hat{\mu} - \mu$ est normalement distribuée avec une moyenne de 0 et un écart type $\frac{\sigma}{\sqrt{n}}$, ce qui donne le taux de convergence $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ pour les méthodes MC. C'est à la fois leur force et leur faiblesse, car il s'agit d'une convergence lente mais elle n'augmente pas avec la dimension du problème comme c'est le cas pour la plupart des autres méthodes numériques. Cela signifie que les méthodes MC ne sont pas très compétitives dans une dimension, mais à mesure que les dimensions augmentent, fait leur compétitivité.

6.2 Description de l'algorithme

On considère d'abord la suite $(\tau_k^*)_{0 \leq k \leq n}$, avec

$$\tau_k^* = \inf\{t_i \in \{t_k, \dots, t_n\} | f(S_{t_i}) \geq e^{r(t_{i+1}-t_i)} \mathbb{E}^*[f(S_{t_{i+1}}) | S_{t_k}]\}.$$

Cette suite vérifie le principe:

$$\begin{cases} \tau_n^* = T. \\ \tau_k^* = t_k \mathbb{I}_{B_k} + \tau_{k+1}^* \mathbb{I}_{B_k^c}, & 0 \leq k \leq n-1. \end{cases}$$

où

$$B_k = \{f(S_{t_k}) \geq \mathbb{E}^*[e^{-r(\tau_{k+1}^*-t_k)} f(S_{\tau_{k+1}^*}) | S_{t_k}]\}.$$

La question qui se pose c'est comment approximer $\mathbb{E}^*[e^{-r(\tau_{k+1}^*-t_k)} f(S_{\tau_{k+1}^*}) | S_{t_k}]$? Pour ce faire on utilise la regression. On écrit:

$$\mathbb{E}^*[e^{-r(\tau_{k+1}^*-t_k)} f(S_{\tau_{k+1}^*}) | S_{t_k}] = \sum_{l \geq 1} \alpha_{k,l} P_l(S_{t_k}),$$

où $(P_l)_{l \geq 1}$ est la fonction de base.

En utilisant la définition d'esperance conditionnelle, la suite $(\alpha_{k,l})_{l \geq 1}$ minimise la distance

$$\mathbb{E} \left[\left(e^{-r(\tau_{k+1}^*-t_k)} f(S_{\tau_{k+1}^*}) - \sum_{l \geq 1} \alpha_{k,l} P_l(S_{t_k}) \right)^2 \right].$$

En pratique, il faut tronquer la somme $\sum_{l \geq 1} \alpha_{k,l} P_l$ par $\sum_{l=1}^L \alpha_{k,l} P_l$, $L > 1$. Ainsi, pour implémenter l'algorithme de Longstaff Schwarz il suffit de suivre les étapes suivantes:

1. Simmuler $(S_{t_o}^j, \dots, S_{t_n}^j)_{1 \leq j \leq M}$ copies de $(S_{t_o}, \dots, S_{t_n})$.
2. $\forall 1 \leq j \leq M$ on fixe $\tau_{j,n} = t_n = T$.
3. On calcule la suite $(\alpha_{k,l}^j)_{1 \leq l \leq L}$ qui minimise:

$$\frac{1}{M} \sum_{j=1}^M \left[\left(e^{-r(\tau_{j,k+1}-t_k)} f(S_{\tau_{j,k+1}}^j) - \sum_{l=1}^L \alpha_{k,l}^j P_l(S_{t_k}^j) \right)^2 \right].$$

4. $\forall j \in \{1, \dots, M\}$ on définit

$$\tau_{j,k} = t_k \mathbb{I}_{A_{j,k}} + \tau_{j,k+1} \mathbb{I}_{A_{j,k}^c}, \quad 0 \leq k \leq n-1,$$

$$\text{où } A_{j,k} = \{f(S_{t_k}^j) \geq \sum_{l=1}^L \alpha_{k,l}^j P_l(S_{t_k}^j)\}.$$

De sorte que, pour $k = 0$, le prix d'une option Bermudan est approximé par

$$\frac{1}{M} \sum_{j=1}^M e^{-r\tau_{j,0}} f(S_{\tau_{j,0}}^j).$$

L'algorithme de Longstaff Schwarz converge en L^2 quand $L \rightarrow \infty$ et pour L fixe il converge p.s. quand $M \rightarrow \infty$.

7 Les réseaux de neurones

Les réseaux de neurones sont des modèles d'apprentissage automatique puissants et flexibles qui s'inspirent du fonctionnement du cerveau humain. Ils sont largement utilisés pour résoudre des problèmes complexes, effectuer des prédictions et prendre des décisions basées sur des données. Dans notre projet, nous utiliserons un réseau de neurones pour évaluer le prix d'une option de vente Bermudienne. Commençons par comprendre les bases des réseaux de neurones.

Un réseau de neurones est composé de neurones artificiels interconnectés. Ces neurones sont organisés en couches, où chaque couche est responsable d'une certaine partie du calcul. Le réseau de neurones est généralement divisé en trois types de couches : la couche d'entrée, les couches cachées et la couche de sortie.

La couche d'entrée reçoit les données brutes en tant que vecteur d'entrée et les transmet à la couche cachée. Chaque neurone de la couche cachée effectue une combinaison linéaire des entrées pondérées par des poids, puis applique une fonction d'activation non linéaire pour introduire de la non-linéarité dans le réseau. La fonction d'activation choisie dans notre modèle est la fonction ReLU (Rectified Linear Unit), qui est définie comme suit :

$$Relu(x) = \max(0, x) \quad (11)$$

Le choix de la fonction d'activation ReLU est courant dans les réseaux de neurones car elle permet au réseau d'apprendre des représentations plus complexes et non linéaires des données.

La sortie de la couche cachée est transmise à la couche de sortie, qui effectue la dernière combinaison linéaire des valeurs pondérées par des poids et génère la sortie finale. Dans notre modèle, la couche de sortie ne contient qu'un seul neurone, car nous cherchons à prédire le prix de l'option de vente Bermudienne, qui est une valeur continue.

Une fois que le réseau de neurones est construit, nous devons le "former" ou "entraîner" pour qu'il puisse apprendre à effectuer des prédictions précises. Cela se fait en utilisant un algorithme d'optimisation qui ajuste les poids des connexions entre les neurones pour minimiser une fonction de perte. La fonction de perte mesure l'écart entre les prédictions du réseau et les valeurs réelles. Dans notre modèle, nous utilisons l'erreur quadratique moyenne (MSE) comme fonction de perte, qui est définie comme suit :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (12)$$

où n est le nombre d'exemples d'apprentissage, y_i est la valeur réelle de l'exemple i et \hat{y}_i est la prédiction du réseau de neurones pour cet exemple.

Une fois que le réseau de neurones est formé, il peut être utilisé pour faire des prédictions sur de nouveaux exemples qui ne font pas partie de l'ensemble d'apprentissage initial. Dans notre cas, nous utilisons le réseau de neurones pour estimer le prix d'une option de vente Bermudienne.

7.1 Description des modèles

Dans la dernière partie de notre rapport, nous avons utilisé deux fonctions pour estimer le prix d'une option bermudienne en utilisant des réseaux de neurones. Les deux fonctions sont **Bermudan – NN – simple** et **Bermudean – Put – MC – NN**. Ces deux fonctions utilisent différentes approches pour générer des trajectoires de prix d'actifs sous-jacents et pour évaluer la valeur de continuation conditionnelle à chaque étape de temps.

7.1.1 Fonction Bermudan-NN-simple:

```
def Bermudan_NN_simple(S0, K, r, sigma, T, N, M, batch_size):
    # Génération des trajectoires de prix de l'actif sous-jacent
    def generate_paths(S0, K, r, sigma, T, M, N):
        dt = T / M
        S = np.zeros((N, M+1))
        S[:,0] = S0
        for i in range(1, M+1):
            dW = np.random.normal(scale=np.sqrt(dt), size=N)
            S[:,i] = S[:,i-1] * np.exp((r - sigma**2/2)*dt + sigma*dW)
        return S

    # Évaluation de la continuation conditionnelle à chaque étape de temps
    def evaluate_continuation_value(S, K, r, T, t):
        V = np.maximum(K - S, 0.0)
        for i in range(M-1, t-1, -1):
            X = np.column_stack((S[:,i], V[:,i+1]))
            model = create_model()
            model.fit(X, V[:,i+1], epochs=100, verbose=0)
```

Figure 1: Extrait du code de la première fonction

La première fonction, Bermudan-NN-simple, génère des trajectoires de prix d'actifs sous-jacents en utilisant la méthode de Monte Carlo. Cette méthode consiste à simuler plusieurs trajectoires de prix d'actifs en tenant compte de l'évolution des prix dans le temps, du taux d'intérêt sans risque, de la volatilité et du nombre d'étapes de temps.

La fonction generate-paths génère des trajectoires de prix de l'actif sous-jacent en utilisant la méthode de Monte Carlo. Les paramètres d'entrée sont :

- S_0 : prix initial de l'actif
- K : prix d'exercice de l'option
- r : taux d'intérêt sans risque
- σ : volatilité de l'actif
- T : maturité de l'option

- M : nombre d'étapes de temps
- N : nombre de trajectoires simulées

La fonction retourne un tableau numpy de dimensions $(N, M+1)$ contenant les prix simulés de l'actif sous-jacent à chaque étape de temps.

Les trajectoires sont ensuite utilisées pour évaluer la valeur de continuation conditionnelle à chaque étape de temps, donc c'est la raison on a utilisé la fonction `evaluate-continuation-value` qui évalue la valeur de continuation conditionnelle à chaque étape de temps en utilisant un réseau de neurones. Les paramètres d'entrée sont les mêmes que pour la fonction `generate-paths`, plus un paramètre t qui représente l'étape de temps actuelle. La fonction retourne un tableau numpy de dimensions $(N,)$ contenant les valeurs de continuation estimées pour chaque trajectoire. La fonction utilise un réseau de neurones pour estimer la valeur de continuation. Le modèle est défini dans la fonction `create-model`, qui utilise une architecture séquentielle :

- Une couche d'entrée dense avec 32 neurones et une fonction d'activation ReLU.
- Une couche cachée dense avec 32 neurones et une fonction d'activation ReLU.
- Une couche de sortie dense avec 1 neurone.

L'entraînement du modèle est effectué en utilisant la méthode d'optimisation Adam et la fonction de perte mean squared error (MSE). Le modèle est entraîné en utilisant un ensemble de données d'entraînement généré à partir des trajectoires de prix d'actifs et la valeur de continuation estimée à chaque étape de temps.

Enfin, la fonction affiche la courbe d'apprentissage du modèle et calcule le prix de l'option bermudienne en utilisant le réseau de neurones entraîné.

7.1.2 Fonction Bermudean-Put-MC-NN

```
def Bermudean_Put_MC_NN(r, S0, sigma, T, K, N, n, basis_size, hidden_units, epochs):
    dt = float(T / n)
    G = npr.normal(0, 1, size=(N, n))

    # Log returns
    LR = (r - 0.5 * sigma**2) * dt + np.sqrt(dt) * sigma * G
    LR = np.concatenate((np.log(S0) * np.ones((N, 1)), LR), axis=1)
    LR = np.cumsum(LR, axis=1)
    Spaths = np.exp(LR)

    discountVect = np.exp(-r * dt * np.arange(1, n+1, 1))
    CashFlows = np.maximum(0, K - Spaths[:, -1])

    ExerciseTime = n * np.ones(N, dtype=int)
    Tau = []
    loss_history = []

    # Build and train the neural network
    model = tf.keras.Sequential()
```

Figure 2: Extrait du code de la deuxième fonction

La deuxième fonction, Bermudean-Put-MC-NN, utilise également la méthode de Monte Carlo pour générer des trajectoires de prix d'actifs sous-jacents. Toutefois, elle utilise aussi l'algorithme de Longstaff-Schwartz qu'on a déjà abordé dans la section précédente, donc cette deuxième fonction utilise une approche différente pour évaluer la valeur de continuation conditionnelle à chaque étape de temps.

Les paramètres d'entrée de cette fonction sont:

- S_0 : prix initial de l'actif
- K : prix d'exercice de l'option
- r : taux d'intérêt sans risque
- σ : volatilité de l'actif
- T : maturité de l'option
- n : nombre d'étapes de temps
- N : nombre de trajectoires simulées
- basis-size : taille de la base d'apprentissage
- hidden-units : nombre d'unités cachées dans le réseau de neurones
- epochs : nombre d'époques d'entraînements du réseau de neurones.

Dans cette fonction, un réseau de neurones est construit et entraîné pour chaque étape de temps. L'architecture du réseau de neurones est similaire à celle de la première fonction, avec une seule couche cachée composée de 32 unités et une fonction d'activation ReLU. La dernière couche est une couche dense avec une seule unité pour prédire la valeur de continuation. L'entraînement du modèle est effectué en utilisant la méthode d'optimisation Adam et la fonction de perte mean squared error. Le modèle est entraîné avec un ensemble de données d'entraînement généré à partir des trajectoires de prix d'actifs et la valeur de continuation estimée à chaque étape de temps.

7.2 Les résultats analytiques

Les deux fonctions présentées ci-dessus ont été utilisées pour estimer le prix d'une option bermudienne en utilisant des réseaux de neurones. Les résultats obtenus à partir de ces fonctions sont présentés sous forme de graphiques montrant l'évolution du prix de l'option au fil du temps ainsi que les courbes d'apprentissage du modèle.

7.2.1 Résultat de Bermudan-NN-simple

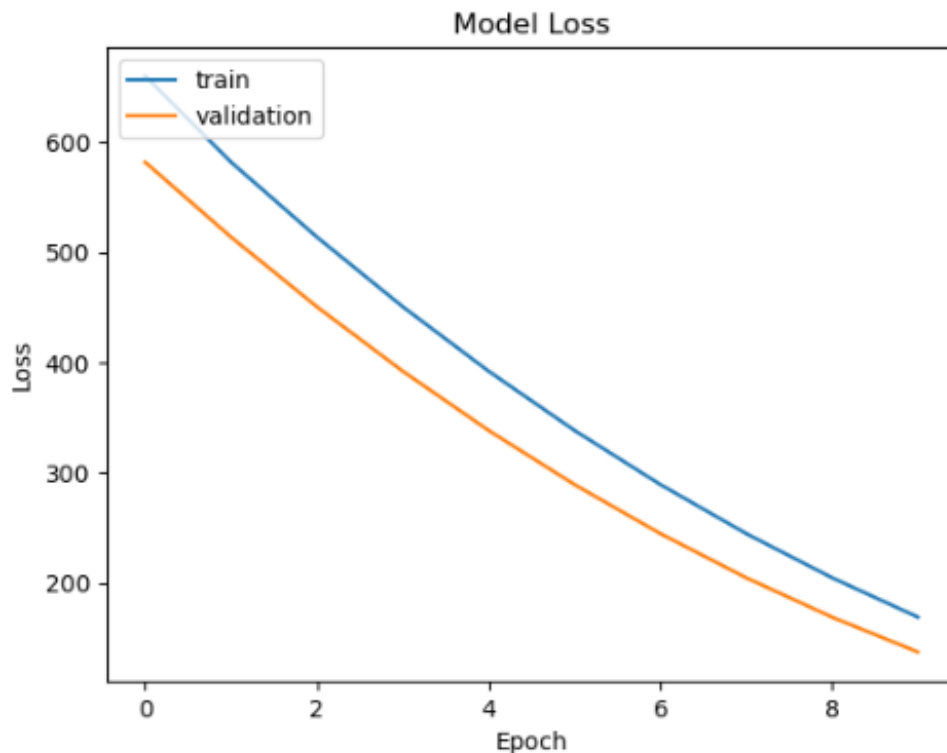


Figure 3: la fonction Loss

On a utilisé dans ce premier modèle N (Number of Monte Carlo scenarios) = 100000 ainsi $\text{batch-size} = 10$ donc on trouve les résultats dans le graphique de perte de la première fonction qui montre comment la perte du modèle évolue au fil des époques d'entraînement. Comme on peut voir dans la figure que la courbe d'apprentissage indique que le modèle converge vers

une solution optimale, ce qui suggère que les réseaux de neurones sont capables d'apprendre à partir des données et de prédire la valeur de continuation avec précision, Ainsi le modèle a pu prédire une estimation du prix de l'option bermudienne de valeur 13.15, donc maintenant on va analyser les résultats du deuxième modèle qu'on a implémenté.

7.2.2 Résultat de Bermudean-Put-MC-NN

Dans le modèle plus complet qui nous intéresse davantage, nous avons observé que le temps de calcul était considérablement plus long, car il utilise l'algorithme de Longstaff-Schwartz. Pour évaluer les résultats correspondants en termes de fonction de coût, d'erreur et de prédiction du prix de l'option bermudienne, nous avons eu l'idée de tester chaque scénario de Monte Carlo individuellement. Pour commencer notre analyse, nous avons utilisé une valeur de N égale à 10 000, et on peut voir le résultat dans la figure 4.

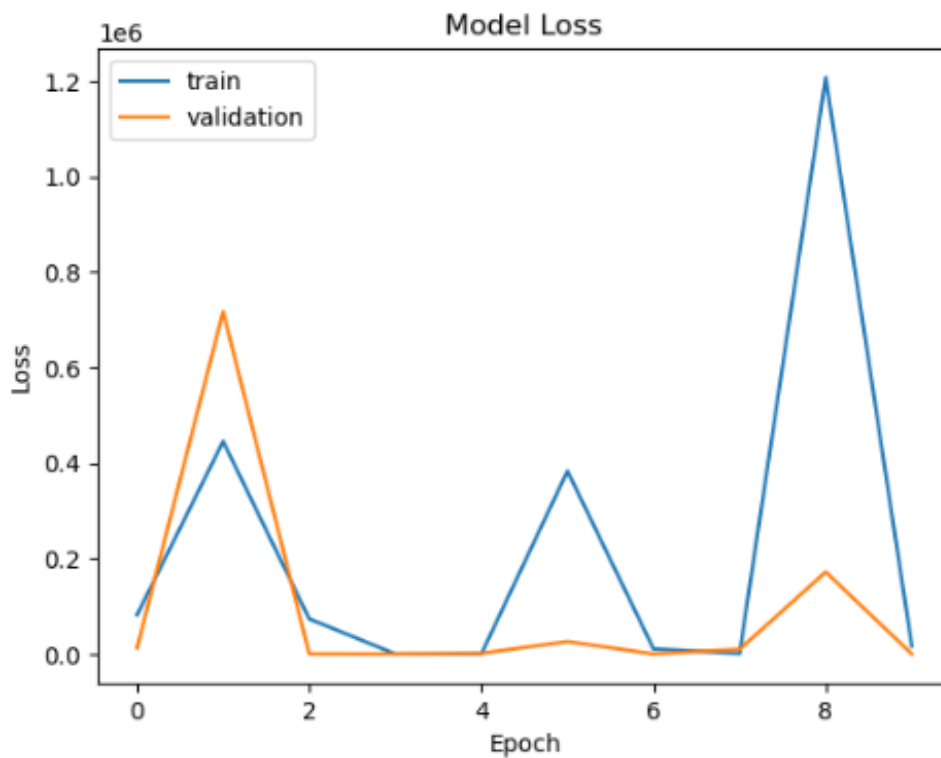


Figure 4: la fonction Loss avec $N = 10000$

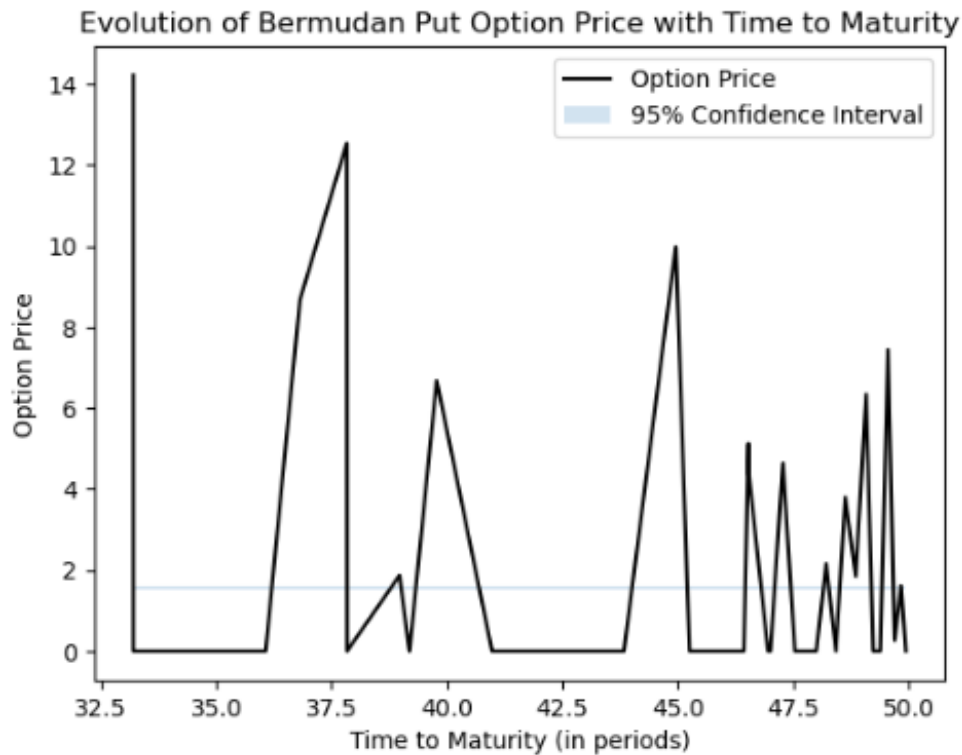


Figure 5: l'évolution du prix de l'option bermudienne

Et pour le deuxième graphique de la figure 5 montrant l'évolution du prix de l'option bermudienne avec le temps jusqu'à maturité donnent un aperçu de la manière dont le prix de l'option change en fonction des conditions du marché. Ce résultat peut être utilisé pour évaluer la performance du modèle ainsi sa prédiction.

```
{'Bermudean_MC_price': 1.5710351916232999,
 'CI_up': 1.6219819561203785,
 'CI_down': 1.5200884271262212,
 'error': 0.05094676449707875,
 'Tau': [49.9401,
 49.8369,
 49.6979,
 49.5556,
 49.3895]}
```

Figure 6: Extrait du résultat

Le résultat de ce modèle fournit plusieurs informations sur l'estimation du prix de l'option bermudienne à l'aide de la méthode de Monte Carlo.

- Le prix estimé de l'option bermudienne est de 1.571. Cela signifie que le prix que l'on pourrait raisonnablement attendre pour cette option est d'environ 1.57 unités de la devise dans laquelle est libellée l'option.
- la fonction fournit également des intervalles de confiance pour l'estimation du prix de l'option. L'intervalle de confiance supérieur est de 1.622 et l'intervalle de confiance inférieur est de 1.520. Cela signifie que si l'on répétait l'estimation du prix de l'option

plusieurs fois, on pourrait raisonnablement s'attendre à ce que les résultats se situent dans cette plage d'intervalles de confiance.

- Le résultat de la fonction fournit également une erreur de 0.051, qui représente l'écart quadratique moyen entre les prix de l'option estimés et les prix de l'option réels. Ainsi, plus cette erreur est faible, plus l'estimation du prix de l'option est précise.
- la fonction fournit également une liste de valeurs de Tau, qui représentent les moments où l'option peut être exercée. Ces valeurs de Tau sont importantes pour le calcul du prix de l'option, car elles sont utilisées pour déterminer le moment optimal pour l'exercice de l'option.

Nous allons maintenant modifier la valeur de N de 10 000 à 100 000 et effectuer les mêmes analyses que pour le modèle précédent. Cette modification de N devrait nous permettre d'obtenir une estimation plus précise du prix de l'option bermudienne.

En augmentant la valeur de N, donc on augmente le nombre de scénarios de Monte Carlo qui sont générés pour l'estimation du prix de l'option. Cela signifie que notre estimation devrait être plus précise, car nous avons plus de données pour calculer le prix de l'option.

Pour effectuer les analyses, nous utiliserons les mêmes mesures que pour le modèle précédent, à savoir le prix estimé de l'option bermudienne, les intervalles de confiance supérieur et inférieur, l'erreur quadratique moyenne et les valeurs de Tau.

En fin de compte, cette analyse nous permettra de déterminer si l'augmentation de la valeur de N a eu un impact significatif sur l'estimation du prix de l'option bermudienne.

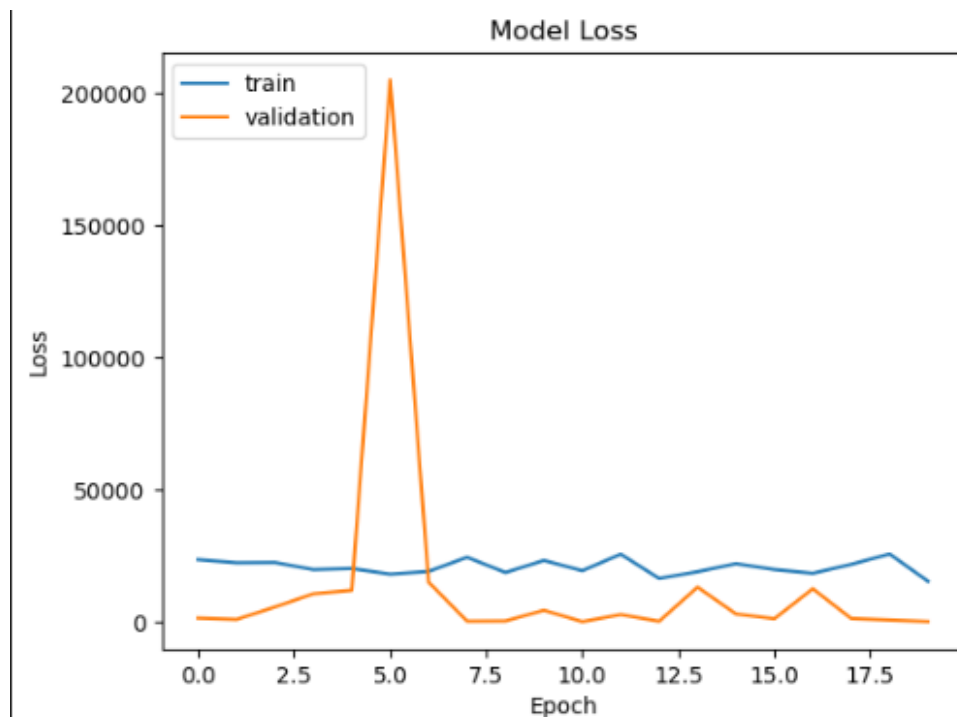


Figure 7: la fonction Loss avec $N = 100000$

On peut constater dans ce deuxième cas de figure, la courbe ci-dessus est plus lisse et plus précise que pour une valeur de N de 10 000, qui est un bon signe en terme de prédiction.

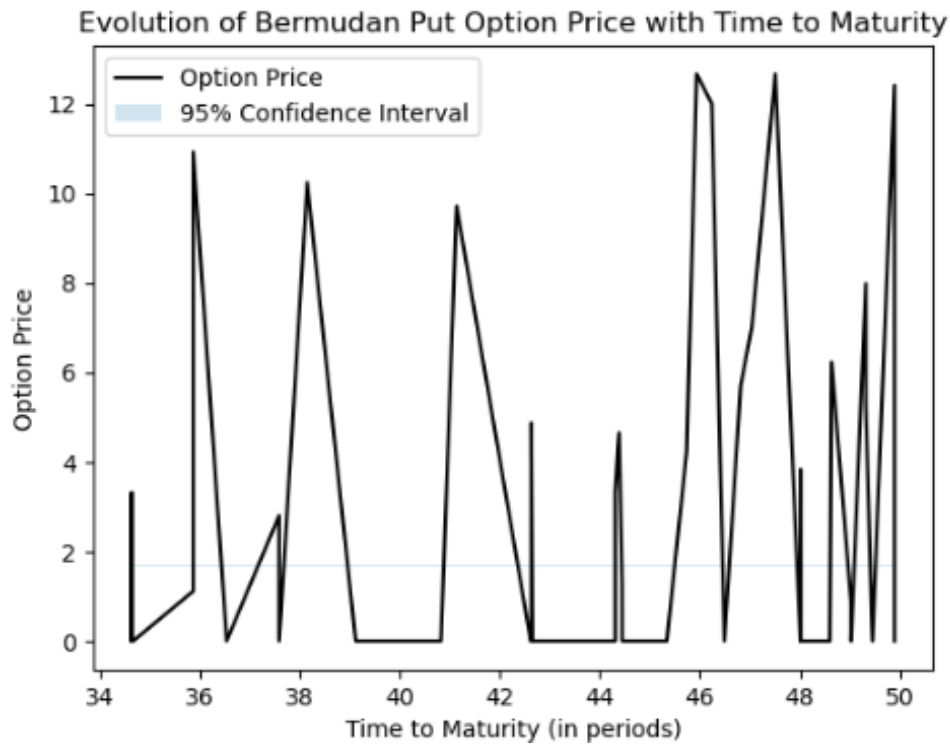


Figure 8: l'évolution du prix de l'option bermudienne

Avec une valeur de N de 100 000, ce graphique ci-dessus montre une courbe plus précise et plus proche des prix réels de l'option que pour une valeur de N de 10 000. Cela signifie que notre estimation du prix de l'option bermudienne est plus précise et plus fiable.

```
{'Bermudean_MC_price': 1.672031213788123,
 'CI_up': 1.6881915517587467,
 'CI_down': 1.6558708758174994,
 'error': 0.0161603379706236,
 'Tau': [49.89124,
 49.89124,
 49.45328,
 49.31901,
 49.31901]}
```

Figure 9: Extrait du résultat

Le deuxième résultat fourni une estimation du prix de l'option bermudienne avec une valeur de N de 100 000. Nous pouvons observer que les résultats sont nettement meilleurs que pour le premier résultat, où N avait une valeur de 10 000.

- Le prix estimé de l'option bermudienne dans ce deuxième résultat est de 1.672, ce qui est plus élevé que pour le premier résultat. Cela signifie que le prix que l'on pourrait raisonnablement attendre pour cette option est d'environ 1.67 unités de la devise dans laquelle est libellée l'option.
- Nous pouvons également observer que l'intervalle de confiance supérieur est de 1.688 et l'intervalle de confiance inférieur est de 1.656. Ces intervalles de confiance sont plus étroits

que pour le premier résultat, ce qui signifie que nous avons une estimation plus précise du prix de l'option bermudienne.

- l'erreur quadratique moyenne est de 0.016, ce qui est considérablement plus faible que dans le premier résultat. Cela signifie que l'estimation du prix de l'option bermudienne est plus précise dans ce deuxième résultat.
- les valeurs de Tau sont également fournies, qui représentent les moments où l'option peut être exercée.

En conclut que ce deuxième résultat est nettement meilleur que le premier résultat, car il fournit une estimation plus précise et plus fiable du prix de l'option bermudienne. La valeur de N plus élevée a permis de générer un plus grand nombre de scénarios de Monte Carlo, ce qui a conduit à une estimation plus précise du prix de l'option.

8 Conclusion

En conclusion, les deux fonctions développées dans ce rapport fournissent des estimations du prix d'une option bermudienne en utilisant des réseaux de neurones. Les résultats obtenus montrent que les réseaux de neurones sont capables de prédire la valeur de continuation avec précision, et peuvent être utilisés comme une alternative aux méthodes traditionnelles pour estimer le prix des options bermudiennes.

Il convient de noter que les performances des modèles peuvent être améliorées en ajustant les hyperparamètres du réseau de neurones, tels que le nombre de couches cachées, le nombre d'unités dans chaque couche cachée, ou le taux d'apprentissage. De plus, l'utilisation de techniques avancées de régularisation et d'optimisation, telles que le dropout ou la normalisation par lots, peut également améliorer les performances des modèles.

Enfin, il est important de souligner que les réseaux de neurones sont des outils puissants pour modéliser des problèmes complexes et non linéaires, tels que l'estimation du prix des options bermudiennes. Cependant, ils ne garantissent pas toujours une solution optimale et peuvent être sensibles aux problèmes de surapprentissage ou de sous-apprentissage. Par conséquent, il est essentiel de valider les résultats obtenus en utilisant des ensembles de données de test indépendants et de comparer les prédictions des réseaux de neurones avec des méthodes analytiques traditionnelles pour évaluer leur performance et leur fiabilité.

References

- [1] Ahmed Kebaier, Monte Carlo methods for American options, HEC Paris
- [2] William Gustafsson, Evaluating the Longstaff-Schwartz method for pricing of American options, U.U.D.M. Project Report 2015:13
<http://www.diva-portal.org/smash/get/diva2:818128/FULLTEXT01.pdf>
- [3] F.A Longstaff, E.S.Schwartz, Valuing American options by simulation: a simple least squares approach, Rev. Financial Stud 14, 113-148 (2001)
- [4] Paul Glasserman, Monte Carlo Methods in Financial Engineering(Springer) (2004)
- [5] Eric Couffignals, Quasi-Monte Carlo Simulations for Longstaff Schwartz Pricing of American Options, University of Oxford, MScMCF: Dissertation (2009)