



École des Ponts

ParisTech

## ÉCOLE NATIONALE DES PONTS ET CHAUSSEES

---

2023-2024

Rapport de Projet de Département IMI

JACQUILLAT Alexis ★ IZZERMINE Faissal ★ LE BRETON Titouan ★ NACHI Brahim

### Machine Learning for Bermudan Option Pricing

Projet réalisé sous la supervision de M. Mohamed Sbai, Senior Quant Advisor à SGCIB

---



**SOCIETE  
GENERALE**

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Méthode Longstaff-Schwartz</b>	<b>2</b>
2.1	Théorie . . . . .	2
2.2	Exemple d'application . . . . .	3
<b>3</b>	<b>Méthode Rogers</b>	<b>3</b>
3.1	Théorie . . . . .	3
3.2	Implémentation à un put américain sur un sous-jacent . . . . .	4
<b>4</b>	<b>Méthode par réseau de neurones</b>	<b>4</b>
4.1	Théorie . . . . .	4
4.2	Implémentation . . . . .	5
<b>5</b>	<b>Comparaison des résultats</b>	<b>6</b>
<b>6</b>	<b>Ouverture</b>	<b>10</b>
<b>7</b>	<b>Conclusion</b>	<b>11</b>
<b>8</b>	<b>Annexe</b>	<b>12</b>
	<b>Références</b>	<b>12</b>

# 1 Introduction

Les options américaines sont des instruments financiers permettant à leur détenteur de les exercer à tout moment durant la vie du produit. Cette flexibilité offre aux investisseurs la possibilité de capitaliser sur les mouvements favorables du marché tout au long de la durée de vie de l'option, un avantage notable dans les contextes de marchés volatils. C'est pour cette raison que ces produits jouissent d'une grande popularité. En pratique, nous traiterons principalement des options bermudiennes, dont la principale différence réside dans le fait que l'ensemble des moments d'exercice est fini.

Nous nous intéresserons plus précisément au pricing de ces options c'est-à-dire à la méthode utilisée pour déterminer le juste prix de la "prime" ou prix de l'option.

D'une part, les méthodes basées sur la régression, avec en tête la méthode de Longstaff-Schwartz, permettent de calibrer une stratégie d'exercice espérée proche de l'optimale. Suite à cette calibration, une tarification classique par la méthode de Monte Carlo de l'option est réalisée, offrant ainsi une borne inférieure du prix du point de vue du détenteur de l'option. D'autre part, la formulation duale permet de déterminer une borne supérieure du prix, également du point de vue du détenteur. Bien que ces approches soient utiles, elles présentent toutes deux des limitations, comme en témoigne l'abondante littérature existante sur le sujet.

L'objectif de ce projet est d'explorer une nouvelle approche prometteuse basée sur les techniques de Machine Learning qui permet de calculer simultanément une borne inférieure et une borne supérieure du prix. Pour cela, nous étudierons successivement les approches Longstaff-Schwartz, Rodgers et de réseaux de neurones afin de confronter nos résultats avec la théorie et de voir si oui ou non, la méthode basée sur le machine learning nous offre plus de précision.

## 2 Méthode Longstaff-Schwartz

### 2.1 Théorie

La méthode de Longstaff-Schwartz [4] a pour objectif de trouver une borne inférieure dans le pricing d'une option bermudienne. Elle se base sur une formulation de programmation dynamique qui permet de revenir au temps précédent :

$$\tilde{V}_T(x) = \tilde{h}_T(x) \quad (1)$$

$$\tilde{V}_{t-1}(x) = \max \left( \tilde{h}_{t-1}(x), \mathbb{E} \left( D_{t-1,t} \tilde{V}_t(X_t) \mid X_{t-1} = x \right) \right) \quad (2)$$

où :

- $X_t$  représente la valeur du sous-jacent au temps  $t$ .
- $h_t$  est le payoff à l'instant  $t$ .
- $V_t$  est la valeur de l'option au temps  $t$  en supposant qu'elle n'a pas encore été exercée.
- $D_{t-1,t}$  est le facteur d'actualisation entre  $t$  et  $t+1$ .
- $\tilde{D}_{t-1,t}$  est le facteur d'actualisation entre  $t$  et  $t+1$ .

À l'échéance, la valeur d'une option est  $V_t(X_t) = h_t(X_t)$ . À la date d'exercice précédente, la valeur de continuation est  $C_{t-1}(x) = E[V_t(X_t) \mid X_{t-1} = x]$ . Ceci est approximé à l'aide d'un ensemble de  $R$  fonctions de base comme suit :

$$C_{t-1}(x) = \sum_{r=1}^R \beta_r \psi_r(x)$$

Les coefficients pouvant être obtenu de plusieurs manières mais la méthode générale étant la minimisation est moindre carré.

Une fois que nous avons l'approximation de la valeur de continuation, que faisons-nous ?

-Si  $C_b(X_{t-1}) < h_{t-1}(X_{t-1})$ , nous exerçons l'option et fixons  $V_{t-1} = h_{t-1}(X_{t-1})$ .

-Sinon, nous fixons  $V_{t-1} = V_t$ . (à noter : Tsitsiklis et van Roy fixait  $V_{t-1} = C_t(X_{t-1})$  avant la formulation de Longstaff et Schwartz).

Ainsi, nous obtenons la valeur  $V_0$  de l'option considérée.

## 2.2 Exemple d'application

L'exemple d'application est un put américaine portant sur un sous-jacent dont le comportement est décrit par le modèle de Black-Scholes, c'est-à-dire :

$$dS_t = rS_t dt + \sigma S_t dW_t$$

avec  $S_t$  la valeur du sous-jacent,  $\sigma$  sa volatilité,  $r$  le taux sans risque et  $W_t$  un mouvement brownien.

Les caractéristiques de l'option sont une maturité de 0,5 an et un prix d'exercice de 100. On suppose que les paramètres du marché sont les suivants : le prix initial du sous-jacent est de 80, sa volatilité est de 0.4 et le taux sans risque est de 0.06. (C'est le même exemple qui sera traité dans la suite du projet

On choisit comme base  $\psi_r$  les  $n$  premiers polynôme de Laguerre en fonction du nombre nécessaire et les valeurs de  $\beta_r$  sont obtenu par une simple régression linéaire.

On calcul via cette méthode les prix des puts pour 10000 simulations différentes et une subdivision de 50 et on en déduit le prix d'un put par une méthode de Monte Carlo. Cette méthode fournit alors une borne inf de 4.468 qui est proche de la valeur trouvée dans l'article [4].

Nous avons également tenté une autre méthode qui consiste à vectoriser le problème et utiliser l'algèbre linéaire pour calculer les coefficients de la régression linéaire qui s'est malheureusement relevée moins précise malgré une plus grande vitesse d'exécution.

## 3 Méthode Rogers

### 3.1 Théorie

La méthode Rogers [5] a pour objectif de trouver une borne supérieure de la valeur de notre put. Elle se base sur une formulation duale du problème de valorisation du put américain. On se passe sur un horizon de temps  $T > 0$ . La formule « classique » pour valoriser une option américaine est la suivante

$$V_0 = \sup_{\tau \in \mathcal{T}} \mathbb{E}_0 \left[ \frac{h_\tau}{B_\tau} \right] \quad (\text{P})$$

où :

- $V_0$  représente la valeur initiale de notre option,
- $h_\tau$  est le payoff à l'instant  $\tau$ ,
- $B_\tau$  est la valeur de l'actif non-risqué à l'instant  $\tau$ .
- $\mathcal{T}$  est l'ensemble des temps d'arrêt à valeur dans  $[0, T]$

Essayer de retrouver la valeur de l'option par ce problème se ramène donc à trouver un temps d'arrêt qui maximise notre espérance, et tout temps d'arrêt fournit une borne inférieure à notre option. Cependant, il existe aussi une formulation duale de ce problème[3] :

$$\inf_{\pi \in \mathcal{M}} \mathbb{E}_0 \left[ \max_{t \in [0, T]} \left( \frac{h_t}{B_t} - \pi_t \right) \right] + \pi_0 \quad (D)$$

$\mathcal{M}$  est l'ensemble des martingales. Toute martingale permet donc d'obtenir une borne supérieure de la valeur de l'option. Le point clé de la méthode Rogers repose sur le choix judicieux d'une martingale ou d'un ensemble de martingales afin d'obtenir un minimum sur cet ensemble qui est proche du minimum global, ce qui signifie obtenir une meilleure borne supérieure.

## 3.2 Implémentation à un put américain sur un sous-jacent

L'exemple d'application de la méthode Rogers est le même que celui de la méthode Longstaff-Schwartz (et c'est l'exemple qui sera repris dans la partie suivante sur les réseaux de neurones) : un put américain portant sur un sous-jacent dont le comportement est décrit par le modèle de Black-Scholes.

Les caractéristiques de l'option sont une maturité de 0,5 an et un prix d'exercice de 100. On suppose que les paramètres du marché sont les suivants : le prix initial du sous-jacent est de 80, sa volatilité est de 0.4 et le taux sans risque est de 0.06.

La martingale  $M$  choisie ici est la valeur actualisée d'un put européen portant sur le même sous-jacent. Calculer la valeur de ce put européen à tout instant n'est pas un souci, puisque l'on dispose d'une formule explicite dans le modèle de Black-Scholes. Concrètement, on n'utilise pas exactement  $M$  comme martingale, mais  $aM$ , avec  $a$  un réel choisi de façon à minimiser  $\mathbb{E} [\sup_{0 \leq t \leq T} (Z_t - aM_t)]$ . La valeur de  $a$  obtenue dans cet exemple est 1.0525.

Déterminer la valeur de l'espérance se ramène alors à une méthode de Monte-Carlo. D'abord, on discrétise le temps, puis on simule un nombre  $N$  de chemins. Pour chaque chemin, on retient la valeur maximale de  $\left( \frac{h_t}{B_t} - \pi_t \right) + \pi_0$ , puis on fait la moyenne sur les  $N$  valeurs afin d'obtenir une simulation de la borne supérieure.

Notre simulation a porté sur 5000 chemins et une subdivision du temps en 50. L'option ayant une valeur de 20.6893, nous obtenons une borne supérieure de 21.6688. De façon assez surprenante, c'est un meilleur résultat que celui de l'article d'origine, qui avait 21.6953 comme borne supérieure. Cependant, cet article date notamment d'il y a plus de 20 ans, et à l'époque, trouver le  $a$  optimal était beaucoup plus coûteux. Ainsi, l'amélioration des résultats est probablement due à la possibilité d'obtenir une valeur de  $a$  plus optimale.

## 4 Méthode par réseau de neurones

### 4.1 Théorie

L'objectif de cette méthode présenté dans [2] est de déterminer à la fois une borne supérieure et inférieure pour la valeur d'un put américain. L'idée est de faire une régression par réseau de neurones (inspirée de [1] ) pour pouvoir estimer deux paramètres,  $\phi$  et  $\psi$ . Cette régression se fait à chaque étape de notre subdivision de temps, que l'on parcourt de façon décroissante.

Les paramètres  $\phi$  et  $\psi$  qu'on veut estimer découlent de l'équation :

$$\frac{V_{t_{i+1}}}{\beta_{t_{i+1}}} = \mathbb{E} \left[ \frac{V_{t_{i+1}}}{\beta_{t_{i+1}}} \mid \mathcal{F}_{t_i} \right] + \int_{t_i}^{t_{i+1}} H_u dW_u$$

avec :

- $V_{t_{i+1}}$  la valeur de notre option à l'instant  $t_{i+1}$
- $B_{t_{i+1}}$  la valeur de l'actif sans risque à l'instant  $t_{i+1}$
- $H_u$  un processus prévisible qui découle de la décomposition de Doob-Meyer de  $V_t$

On cherche à estimer  $\mathbb{E} \left[ \frac{V_{t_{i+1}}}{\beta_{t_{i+1}}} \mid \mathcal{F}_{t_i} \right]$  (appelé terme de continuation) par  $\phi(S_{t_i})$  (où  $S_{t_i}$  est la valeur du sous-jacent) et  $\int_{t_i}^{t_{i+1}} H_u dW_u$  (appelé incrément martingale) par  $\psi(S_{t_i})\Delta W_{t_i}$ . Le réseau de neurones qu'on implémente à chaque instant  $t_i$  à donc pour objectif de minimiser :

$$\min_{\Phi, \Psi} \left( \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Phi(S_{t_i}) - \Psi(S_{t_i})\Delta W_{t_i} \right)^2$$

où  $Y_{t_{i+1}}$  représente la borne inférieure de notre option à l'instant  $t_{i+1}$ .

## 4.2 Implémentation

L'algorithme qu'on a implémenté est le suivant :

---

**Résultat :** Fonctions  $\Phi_{t_i}, \Psi_{t_i}$  pour  $i \in \{0, 1, \dots, n-1\}$

Simuler  $N$  trajectoires d'actions

Initialiser  $Y_{t_n} = X_{t_n} = f(S_{t_n})$

**for**  $i = n-1$  **jusqu'à** 0 **do**

Régression de  $\beta_{\Delta t}^{-1} Y_{t_{i+1}}$  sur  $S_{t_i}$  :  $\min_{\Phi, \Psi} \left( \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Phi(S_{t_i}) - \Psi(S_{t_i})\Delta W_{t_i} \right)^2$

$Y_{t_i} = \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Psi_{t_i}(S_{t_i})\Delta W_{t_i}$

$X_{t_i} = \beta_{\Delta t}^{-1} X_{t_{i+1}} - \Psi_{t_i}(S_{t_i})\Delta W_{t_i}$

**if**  $f(S_{t_i}) > \Phi_{t_i}(S_{t_i})$  **then**

$Y_{t_i} = f(S_{t_i})$

**end if**

**if**  $f(S_{t_i}) > X_{t_i}$  **then**

$X_{t_i} = f(S_{t_i})$

**end if**

**end for**

Régression de  $\beta_{\Delta t}^{-1} Y_{t_1}$  sur  $S_{t_0}$  :  $\min_{\Phi, \Psi} \left( \beta_{\Delta t}^{-1} Y_{t_1} - \Phi(S_{t_0}) - \Psi(S_{t_0})\Delta W_{t_0} \right)^2$

$Y_{t_0} = \beta_{\Delta t}^{-1} Y_{t_1} - \Psi_{t_0}(S_{t_0})\Delta W_{t_0}$

$X_{t_0} = \beta_{\Delta t}^{-1} X_{t_1} - \Psi_{t_0}(S_{t_0})\Delta W_{t_0}$

---

On l'applique pour un put de strike price 40 et de maturité 1 an. On suppose que la volatilité est de 0.2 sur le marché, avec un taux sans risque de 0.06, et que la valeur initiale du sous-jacent est de 36. Le temps est discrétisé en 50 instants.

La question clé est quelle structure donner à notre réseau de neurones qui va faire notre algorithme. On commence par un réseau qui consiste de 2 MLP connectés : chaque MLP a un input (le prix du sous-jacent) et un output ( $\Phi$  ou  $\Psi$ ) ; le forward step se fait de façon indépendante, mais pas la backpropagation. La première structure choisie est une à deux couches pour le réseau  $\Psi$  et trois couches pour le réseau  $\Phi$ . Elle permet d'avoir comme bornes [4.4376, 4.5657], soit un écart d'environ 0.13. Un souci majeur de cette première version de code est son temps d'exécution : environ 2h.

Afin d'améliorer nos résultats, on commence par ajouter un terme  $\Psi_2$  à l'approximation du terme de martingale  $\int_{t_i}^{t_{i+1}} H_u, dW_u$  :

Nous nous appuyons sur la méthode de Milstein présentée dans l'article de référence [6]. L'intuition derrière cette méthode est que, si on écrit  $X(t) = \int_0^t H_u dW_u$ , l'équation différentielle stochastique vérifiée par  $X$  est donc, en posant  $b = H_t$  :

$$dX(t) = b dW_t$$

$\Delta w_{i+1}$  approche  $\int_{t_i}^{t_{i+1}} H_u dW_u$ , par :

$$\Delta w_{i+1} = b\Delta W_i + \frac{1}{2}bb_x(\Delta W_i^2 - \Delta t_i)$$

où  $\Delta W_i$  est une simulation d'une  $\sqrt{\Delta t_i}N(0, 1)$ , et les variables indicées caractérisent des dérivées partielles.

On approxime maintenant l'intégrale par  $\psi(S_{t_i})\Delta W_{t_i} + \Psi_2(S_{t_i})(\Delta W_{t_i}^2 - dt)$ . On cherche donc à minimiser :

$$\min_{\Phi, \Psi, \Psi_2} \left( \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Phi(S_{t_i}) - \Psi(S_{t_i})\Delta W_{t_i} - \Psi_2(S_{t_i})(\Delta W_{t_i}^2 - dt) \right)^2$$

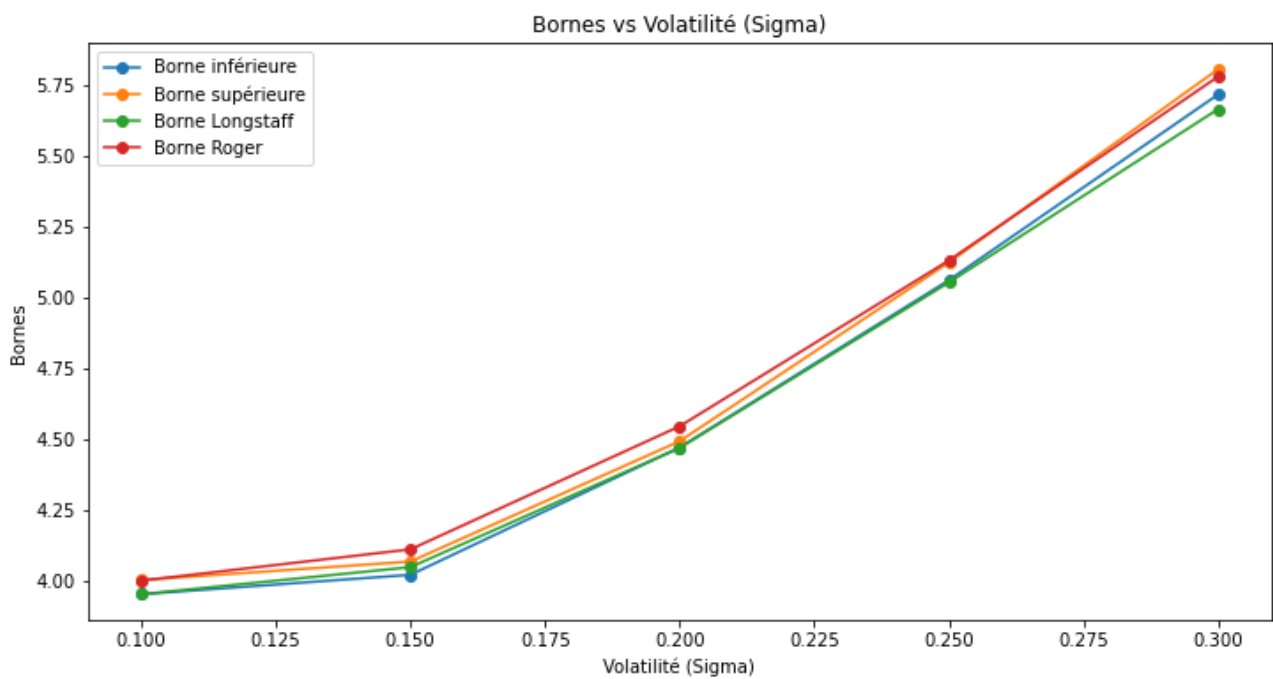
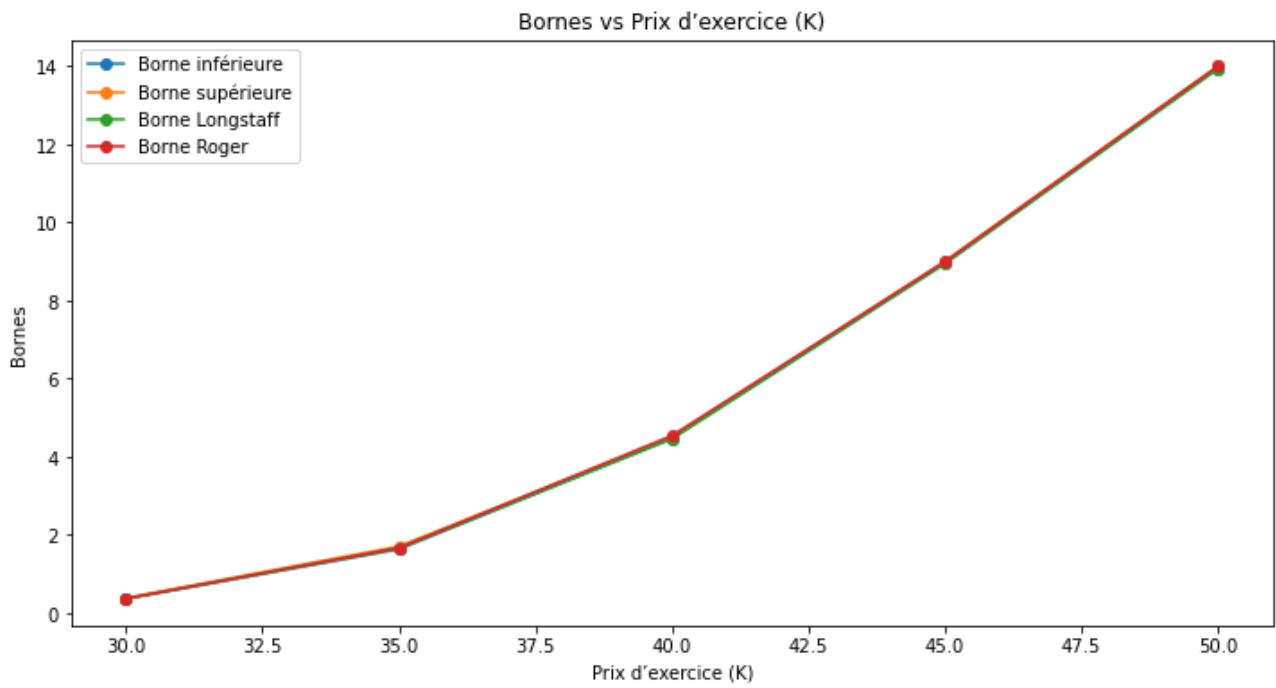
L'ajout de ce troisième terme signifie que notre réseau doit avoir maintenant trois sorties.

De plus, on donne plus de paramètres à notre réseau. Chaque MLP a maintenant 5 couches, chacune avec 100 neurones. Ceci permet de passer de 5000 paramètres à 120 000. Finalement, afin de pouvoir exécuter notre code dans un temps raisonnable, on l'adapte pour tourner sur GPU Nvidia (RTX 4060) en utilisant TensorFlow.

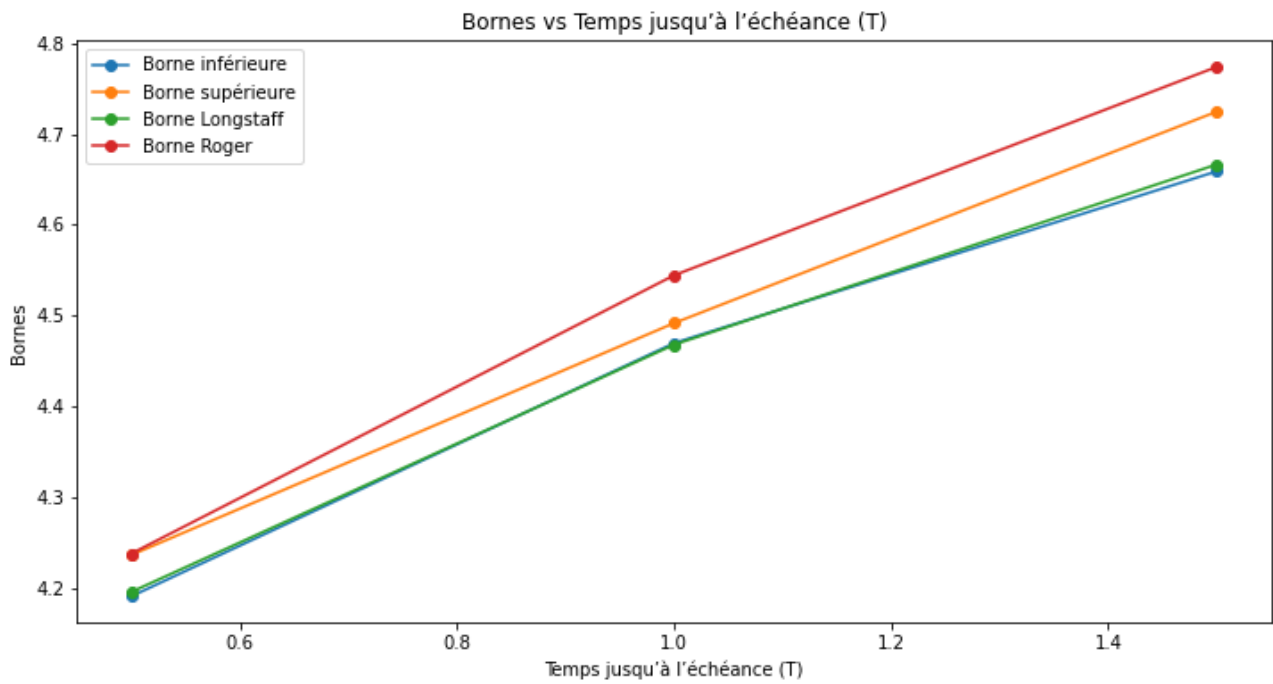
On obtient ainsi un code qui tourne en 20 minutes et qui a comme bornes [4.4697, 4.4917], soit un écart de 0.02, une réduction de 6.5 par rapport à l'écart précédent.

## 5 Comparaison des résultats

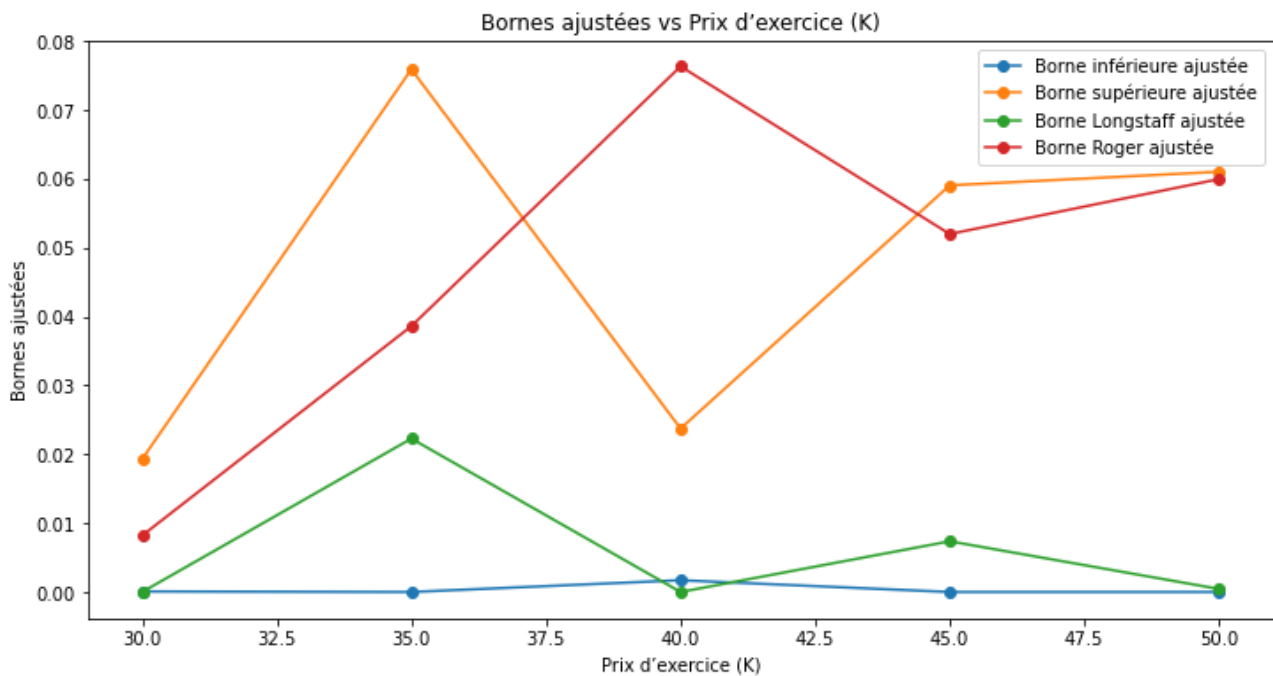
À continuation, nous allons comparer l'output des différentes méthodes sur un ensemble de cas. Le cas de base est celui de la partie précédente : strike à 40 et maturité d'un an, sous-jacent à valeur initiale de 36 et volatilité du marché à 0.2 avec un taux sans risque de 0.06 par an. Nous examinerons comment les bornes évoluent pour chacune des trois méthodes lorsque, à partir de cette configuration de base, nous faisons varier le strike price, la volatilité et la maturité :

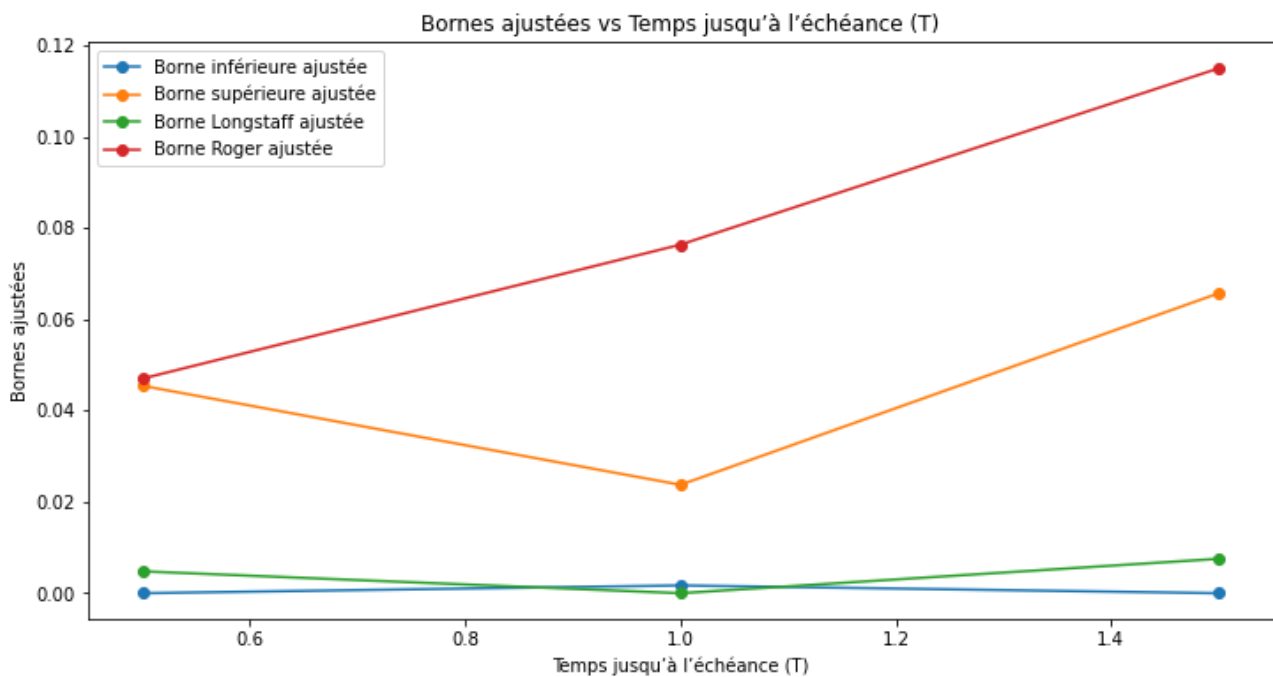
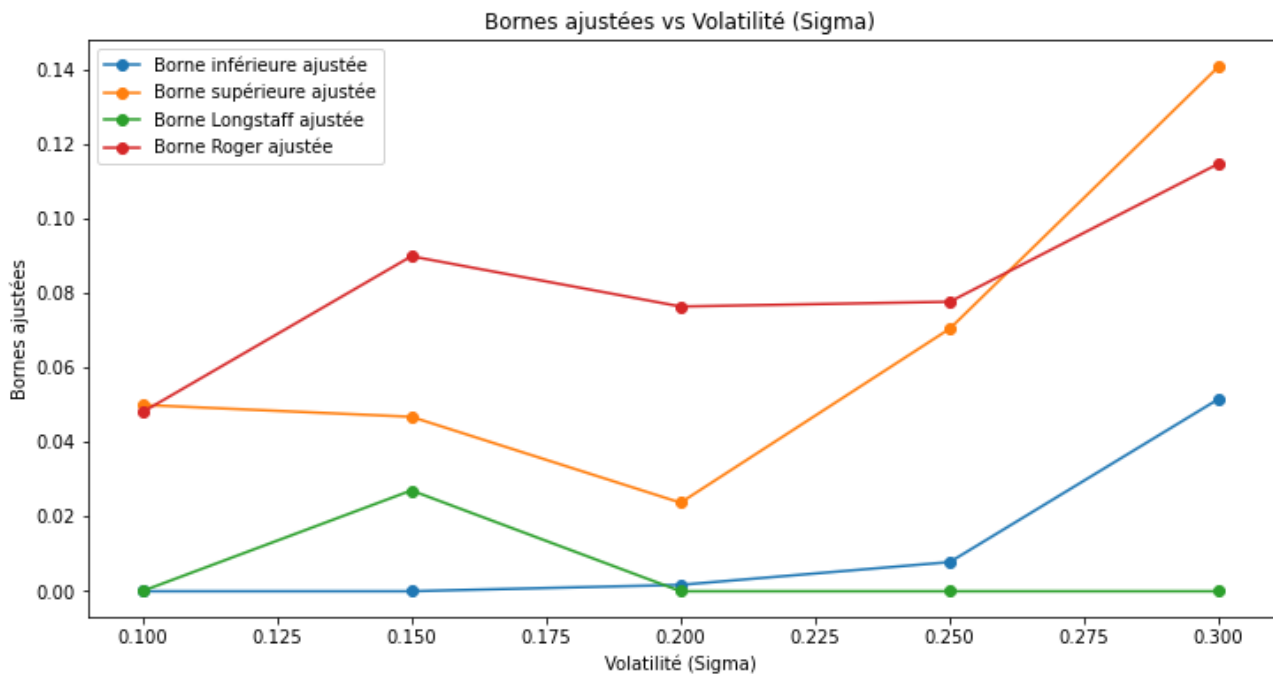






On constate que nos résultats sont satisfaisants dans la mesure où toutes les bornes ont environ la même valeur : il n'y a pas d'anomalie particulière. Cependant, il est difficile de voir la différence entre les différentes méthodes, notamment dans le cas de la variation du strike price. On fait donc des graphes ajustés, où l'on soustrait à chaque fois la valeur de la plus petite borne. :





On observe qu'aucune méthode ne performe mieux que les autres de façon consistante. Sur la borne supérieure, la méthode par réseau de neurones fonctionne mieux que la méthode Longstaff-Schwartz pour la borne supérieure sur la volatilité et la maturité, mais pas sur le strike. Quant à la borne inférieure, le meilleur modèle varie selon les cas. Dans l'ensemble, on constate que l'écart entre nos bornes est de l'ordre de 0.10 sur une option ayant une valeur de l'ordre de 10, ce qui est assez faible. Cependant, il y a des cas, notamment le cas de base, où choisir une borne ou une autre (supérieure dans le cas de base) peut doubler l'écart entre les bornes.

## 6 Ouverture

Dans cette section, nous explorerons rapidement d'autres approches. Nous nous appuierons également sur l'article de référence [6] pour remplacer la méthode de Milstein précédemment utilisée pour résoudre les équations différentielles stochastiques (SDEs) par la méthode de Taylor faible d'ordre 2. Cette dernière est considérée parmi les méthodes de résolution faible les plus efficaces pour les SDEs. L'idée fondamentale de cette méthode est que, si on écrit  $X(t) = \int_0^t H_u dW_u$ , l'équation différentielle stochastique vérifiée par  $X$  est donc, en posant  $b = H_t$  :

$$dX(t) = b dW_t$$

$\int_{t_i}^{t_{i+1}} H_u dW_u$  est approchée par :

$$\Delta w_{i+1} = b\Delta W_i + \frac{1}{2}bb_x(\Delta W_i^2 - \Delta t_i) + \left(\frac{1}{2}b_{xx}b^2\right)(\Delta W_i\Delta t_i - \Delta Z_i)$$

où  $\Delta W_i$  est une simulation d'une  $\sqrt{\Delta t_i}N(0,1)$  et  $\Delta Z_i$  est une variable aléatoire normale avec une moyenne de 0 et une variance  $E(\Delta Z_i^2) = \frac{1}{3}\Delta t_i^3$  et qui est corrélée avec  $\Delta W_i$  par la covariance  $E(\Delta Z_i\Delta W_i) = \frac{1}{2}\Delta t_i^2$ .

Notons que  $\Delta Z_i$  peut être généré par :  $\Delta Z_i = \frac{1}{2}\Delta t_i \left(\Delta W_i + \frac{\Delta V_i}{\sqrt{3}}\right)$  où  $\Delta V_i$  est indépendante et suit une distribution  $\sqrt{\Delta t_i}N(0,1)$ .

Techniquement, cela implique de simuler une nouvelle variable aléatoire  $L_{t_i}$  à chaque pas de temps, distribuée selon une loi gaussienne de moyenne 0 et de variance  $0.5 \times \Delta t$ . Le choix du coefficient (ici 0.5) est arbitraire, car il peut être absorbé dans la fonction correspondant à ce nouveau terme. Ensuite, un autre terme,  $\gamma_1$ , est ajouté à l'approximation du terme de martingale  $\int_{t_i}^{t_{i+1}} H_u dW_u$  : on l'approxime maintenant par  $\psi(S_{t_i})\Delta W_{t_i} + \Psi_2(S_{t_i})(\Delta W_{t_i}^2 - dt) + \gamma_1(S_{t_i})L_{t_i}$ . On cherche donc à minimiser :

$$\min_{\Phi, \Psi, \Psi_2, \gamma_1} \left( \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Phi(S_{t_i}) - \Psi(S_{t_i})\Delta W_{t_i} - \Psi_2(S_{t_i})(\Delta W_{t_i}^2 - dt) - \gamma_1(S_{t_i})L_{t_i} \right)^2$$

L'ajout de ce quatrième terme signifie que notre réseau doit désormais avoir quatre sorties.

Pour rappel, dans notre comparaison, conformément aux sections précédentes, nous considérons les paramètres suivants : un put avec un prix d'exercice de 40 et une maturité de 1 an. Nous supposons que la volatilité sur le marché est de 0.2, avec un taux sans risque de 0.06, et que la valeur initiale du sous-jacent est de 36. Le temps est discrétisé en 50 intervalles.

Avec un réseau de neurones, cette fois-ci composé de 172904 paramètres, on obtient ainsi un code qui s'exécute en 35 minutes et qui a pour bornes [4.5105214, 4.4642377], soit un écart de 0.046283722.

Nous pouvons aller plus loin en ajoutant un autre terme, en utilisant la méthode forte de Taylor d'ordre 1,5, également exposée dans [6]. Cette méthode peut être décrite par la formule suivante, en utilisant la même variable simulée précédemment, et en conservant les mêmes notations que pour la méthode de Taylor faible :

$$\begin{aligned} \Delta w_{i+1} = & b\Delta W_i + \frac{1}{2}bb_x(\Delta W_i^2 - \Delta t_i) + \left(\frac{1}{2}b_{xx}b^2\right)(\Delta W_i\Delta t_i - \Delta Z_i) \\ & + \frac{1}{2}b(bb_{xx} + b_x^2)\left(\frac{1}{3}\Delta W_i^2 - \Delta t_i\right)\Delta W_i \end{aligned}$$

Pour notre implémentation, nous répétons la même approche que précédemment, mais notre réseau disposera d'une sortie supplémentaire, en utilisant les mêmes notations que la méthode précédente. Voici le problème d'optimisation que nous aurons :

$$\min_{\Phi, \Psi, \Psi_2, \gamma_1, \gamma_2} \left( \beta_{\Delta t}^{-1} Y_{t_{i+1}} - \Phi(S_{t_i}) - \Psi(S_{t_i}) \Delta W_{t_i} - \Psi_2(S_{t_i}) (\Delta W_{t_i}^2 - \Delta t) \right. \\ \left. - \gamma_1(S_{t_i}) L_{t_i} - \gamma_2(S_{t_i}) \left( \frac{1}{3} \Delta W_{t_i}^2 - \Delta W_{t_i} \right) \right)^2$$

Sachant que le réseau cette fois-ci comporte 203505 paramètres, le code s'exécute en 40 minutes et donne des bornes de [4.5206633, 4.4622817], soit un écart de 0.058381557 .

Déjà cette méthode prend beaucoup de temps à s'exécuter donc il est dur de boucler sur les hyperparamètres afin de déterminer lesquels sont les meilleurs. On obtient donc des résultats peu cohérents avec la partie sur les réseaux neurones. En effet, l'écart entre les bornes, dans notre codes, est une fonction croissante du nombre de paramètre. Il est toutefois à nuancer que l'écart de 0.02 a été trouvé après beaucoup d'itération et un choix plus optimal d'hyperparamètre. Ainsi, on peut penser que ce problème de résultats est due à un mauvais choix d'hyperparametre.

## 7 Conclusion

Dans notre étude sur le pricing des options bermudiennes, nous avons examiné et comparé les performances de plusieurs méthodes classiques et innovantes, avec un accent particulier sur l'incorporation de techniques de machine learning pour améliorer la précision du pricing. Les méthodes de Longstaff-Schwartz et Rogers, bien que robustes, montrent leurs limites lorsqu'elles sont confrontées à des problèmes de haute dimension et des marchés financiers en constante évolution. Ces méthodes traditionnelles, centrées respectivement sur les approches par régression et formulation duale, fournissent des estimations utiles mais souvent imprécises en présence de conditions de marché complexes.

Pour surmonter ces limitations, notre projet a implémenté une méthode basée sur des réseaux de neurones pour estimer simultanément les bornes inférieure et supérieure des prix des options. Cette approche moderne a permis de traiter de manière plus efficace la complexité et la multidimensionalité des données de marché. Grâce à cette innovation, nous avons non seulement atteint une précision améliorée dans le pricing des options mais avons également réduit le temps de calcul nécessaire, exploitant la capacité des réseaux de neurones à apprendre de vastes ensembles de données et à identifier des patterns complexes que les méthodes traditionnelles ne peuvent pas facilement capturer.

Notre application pratique a démontré que les réseaux de neurones peuvent efficacement remplacer, dans de nombreux cas, les approximations linéaires par des modèles non linéaires et dynamiques, offrant ainsi des estimations de prix plus adaptatives et réactives aux fluctuations du marché. Les résultats obtenus suggèrent que notre approche peut fournir des estimations plus proches des vrais prix de marché, avec des marges d'erreur réduites par rapport aux méthodes conventionnelles. Toutefois en tentant d'augmenter le nombre de parametre du réseau, les résultats ne sont pas concluants, ce qui rappelle l'importance du choix des hyperparametre et appelle à la prudence sur l'utilisation du Machine Learning.

En conclusion, ce projet a non seulement validé l'efficacité de l'intégration des techniques de machine learning dans le domaine du pricing des options, mais a également ouvert des perspectives pour de futures recherches. Il encourage l'exploration continue de méthodes de deep learning plus avancées et leur application dans d'autres types d'instruments financiers. Ces travaux pourraient éventuellement mener à un changement de paradigme dans les pratiques de pricing au sein de l'industrie financière, favorisant une prise de décision plus informée et stratégique basée sur des analyses prédictives précises et robustes.

## 8 Annexe

Vous trouverez sur ce lien Github, l'ensemble des codes réalisés pour le projet. Pour faire très simple, il y a trois codes principaux (correspondant aux trois méthodes présentées) autour duquel ont été fait des ajouts et des changements mineurs.

<https://github.com/POupi2305/Code-Projet-IMI/tree/main>

## Références

- [1] Jérôme Lelong Bernard Lapeyre. *Neural network regression for Bermudan option pricing*. 2020.
- [2] Ivan Guo, Nicolas Langren'e, and Jiahao Wu. *Simultaneous upper and lower bounds of American option prices with hedging via neural networks*. 2023.
- [3] Martin B. Haugh and Leonid Kogan. *Pricing American Options : A Duality Approach*. 2001.
- [4] Francis A Longstaff and Eduardo S Schwartz. *Valuing American Options by Simulation : A Simple Least-Squares Approach*. 2001.
- [5] L. C. Rogers. *Monte Carlo valuation of American Options*. 2002.
- [6] Timothy Sauer. *Numerical Solution of Stochastic Differential Equations with Jumps in Finance*. 2012.