*06/11/2025*

# TRINOMIAL TREE PRICER

Lan Phuong VU & Romain BOREAN

Paris-Dauphine PSL

Professeur : Emmanuel Fruchard

Master 272

Quant Finance

# Mémoire : Option Pricer avec Arbre Trinominal

## Table des matières

# I. Introduction

A travers ce mémoire, nous allons présenter le fonctionnement de notre pricer et ses performances. Nous allons également étudier son comportement sous différentes contraintes et transmettre les difficultés rencontrées lors de l'élaboration de notre projet.

Pour la suite et sauf mention du contraire, les paramètres utilisés seront les suivants :

- **Spot ($S_0$)** : 100,00 €
- **Prix d'exercice (K)** : 102,00 €
- **Date de pricing** : 01/09/2025
- **Maturité** : 01/09/2026
- **Nombre de pas $N$** : 1 000
- **Pas de temps $\Delta t$** : 0,001
- **Volatilité ($\sigma$)** : 30,00 %
- **Taux sans risque ($r$)** : 5,00 %
- **Dividende (discret)** : 0 €
- **Rendement du dividende ($q = \rho = \frac{dividende}{spot}$)** : 3,00 %
- **Date de dividende** : 21/08/2026
- **Alpha** ($\alpha = e^{\sigma \cdot \sqrt{3 \cdot \Delta t}}$) : 1,016567419

# II. Fonctionnement du pricer

## A. Utilisation du Pricer

Pour lancer un pricing, il faut tout d'abord renseigner les différents paramètres dans la feuille « Param», puis sélectionner la méthode et le langage de programmation souhaités.

Le bouton « RUN PRICER » permet de :

- Calculer et afficher le prix de l'option ainsi que la durée de calcul,
- Mesurer l'erreur entre le prix issu de l'arbre et celui de Black-Scholes,
- Choisir le mode d'optimisation : avec pruning ou sans pruning,
- Visualiser les arbres : évolution du prix du sous-jacent, de l'option, ainsi que les probabilités $p\_reach$, $p\_up$, $p\_down$, et $p\_mid$.

Remarque : le prix Black-Scholes n'est pas calculé si l'option est américaine ou si des dividendes sont présents.

Spécification pour les dividendes :

- En cas de dividende discret, il suffit d'indiquer le montant du dividende. Le taux proportionnel (ρ) est alors calculé automatiquement selon la formule Div/Spot
- Si aucun λ (lambda) n'est considéré dans le modèle, il doit être fixé à 0.

## B. Test de vitesse

Le bouton “TEST VITESSE” intègre une fonctionnalité de test de vitesse permettant d’évaluer la performance d’exécution selon plusieurs configurations. Ce test compare :

- Les cas avec et sans pruning,
- Les méthodes de pricing : Backward et Récursive,
- Les langages de programmation : Python et VBA.

Il permet d’analyser la rapidité et la scalabilité du modèle en fonction du nombre d’étapes (N) et du niveau d’optimisation choisi.

## C. Test de convergence

Pour étudier la convergence du modèle trinominal vers le prix théorique de Black-Scholes, le bouton “TEST CONVERGENCE” intègre une fonctionnalité de test de vitesse permettant mesurer la différence entre les deux prix en fonction du nombre de pas (N). Les graphiques et résultats ne sont cependant pas affichés en présence de dividendes ou pour une option américaine, ces cas n’ayant pas de solution analytique directe sous Black-Scholes.

## D. Test de pruning

Afin d’étudier l’impact du seuil de pruning sur le modèle, ce test consiste à faire varier le seuil de pruning, à recalculer le prix de l’option et à mesurer le temps d’exécution pour chaque valeur. Les résultats obtenus sont ensuite comparés à ceux du modèle sans pruning, afin d’évaluer le compromis entre précision et rapidité de calcul.

## E. Test des Sensibilités pour le prix d'option

Ce test permet d’analyser la sensibilité du prix de l’option à la variation des paramètres de marché :

- Volatilité (σ)
- Taux sans risque (r)
- Maturité (T)
- Prix d’exercice (K)

Pour chaque paramètre, le pricer calcule et affiche la variation du prix d'option du modèle et Black-Scholes ainsi que leur différence, ce qui permet d’interpréter le comportement du modèle face aux changements de conditions de marché.

## F. Calcul des Grecques

Le module de **calcul des Grecques** (delta, gamma, vega, theta, rho, vanna, vomma) repose sur une approche **numérique par différences finies**. Le pricer recalcule le prix de l’option pour de petites perturbations des paramètres afin d’obtenir une estimation précise des sensibilités.

## G. Test de Sensibilité pour les Grecques

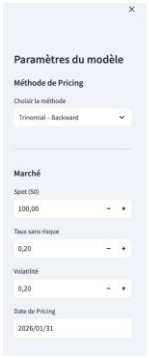
Ce test permet ensuite de visualiser l’évolution de ces indicateurs en fonction des 3 variables de marché (la volatilité, le taux sans risque, le strike) vis-à-vis du modèle Black-Scholes. Cela offre une compréhension complète de la réactivité du modèle et de la robustesse du pricing face aux variations des paramètres.

## III. Fonctionnement de l'application Streamlit

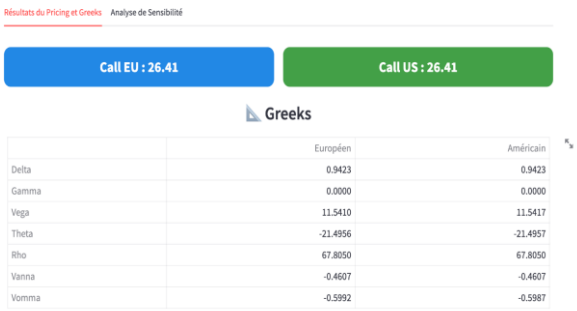
Cette application permet de comparer les options américaines et européennes et d'analyser la sensibilité des prix aux différents paramètres de marché.

Interface de l'application :

- Barre latérale gauche : Permet de renseigner les différents paramètres du modèle (spot, volatilité, taux, maturité, etc.).
- Premier onglet – "Résultats du Pricing et Greeks" : Affiche les prix des options américaines et européennes, ainsi que les Greeks correspondants.
- Deuxième onglet – "Analyse de Sensibilité" : Présente les résultats des tests de sensibilité du prix en fonction du paramètre sélectionné et de l'intervalle défini par l'utilisateur.








# IV. VBA vs Python

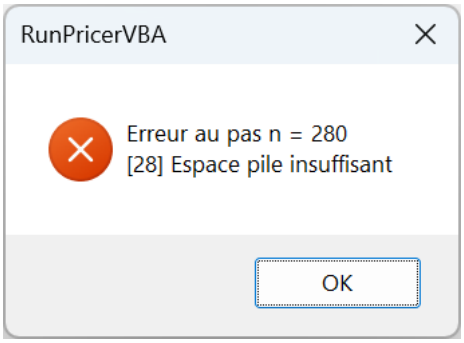
Nous avons tout d'abord commencé à coder notre projet en VBA car cela était plus simple, Excel nous servait directement d'interface utilisateur sans paramétrage nécessaire.

Mais rapidement nous avons été confrontés à ses limites :

## 1) Lenteur d'exécution

VBA est plus lourd et moins efficace que Python, pourtant tous deux des langages interprétés. Mais, l'atout cité précédemment en fait aussi sa faiblesse. Cette « interface utilisateur intégrée » (Excel) le rend plus lent car les objets Excel (Range, Worksheet, etc.) sont lourds à manipuler. De plus, Excel utilise le moteur d'Excel pour ses boucles (For / While) tandis que Python vectorise ces boucles via NumPy.

Nous avons tenté de d'optimiser au mieux ce code, via par exemple la mémoïsation qui consiste à stocker des résultats de calculs pour éviter de les refaire plus tard. Mais cela impose une utilisation plus élevée de la mémoire et on se heurte aux limites RAM de la machine ou bien tomber sur l'erreur « Espace pile insuffisant [28] », atteint au pas $N = 280$ dans notre cas avec la méthode récursive.




## 2) La solution : Python

Python n'est pas une solution miracle car il limite aussi le nombre de récursion, mais il dispose en revanche de nombreux autres outils.

Par exemple, NumPy permet de vectoriser les opérations et de les exécuter en langage C, tandis que Numba permet de compiler les boucles Python en C/C++, améliorant ainsi considérablement la performance.

Sur un MacBook M1, la première version du modèle atteignait un temps d'exécution d'environ 2,8 secondes pour N = 500 avec pruning. Dans la version optimisée, ce temps a été réduit à environ 1,48 seconde pour le même nombre de pas (N = 500), ce qui représente une amélioration notable des performances.

## 3) Autres solutions d'optimisation

En dehors des techniques d'optimisation propres aux différents langages, il y a le pruning, qui consiste à « négliger » des nœuds si la probabilité qu'il soit atteint est inférieure à un certain seuil. Cela permet d'économiser en calcul mais « fausse » donc le prix final. Nous allons étudier l'impact de ce pruning sur le prix final. Mais avant cela, il convient d'abord de vérifier si notre pricer rempli correctement son objectif.

# V. Vérifications

Pour vérifier que notre pricer fonctionne correctement, on peut procéder par étapes, en vérifiant si chacune des hypothèses théoriques sont vérifiées :

## 1) Prix des Call Européens et Américains sans dividendes = Prix Black & Scholes

On a bien $Call\ EU\ \&\ US\ =\ Call\ BS\ \pm\ Tree\ Error$ :

| MARKET | |
|---|---|
| Spot | 100,00 |
| Volatilité | 30,00% |
| Taux | 5,00% |
| Dividende (discret) | 0 |
| Div Yield (rho = dividende /spot) | 0,00% |
| Taux croissance de div (lambda) | 0,00% |
| Date Dividende | 21/08/2026 |

| OPTION | |
|---|---|
| Prix d'Exercice | 102,00 |
| Maturité | 01/09/2026 |
| Call/Put | Call |
| Exercice | EU |

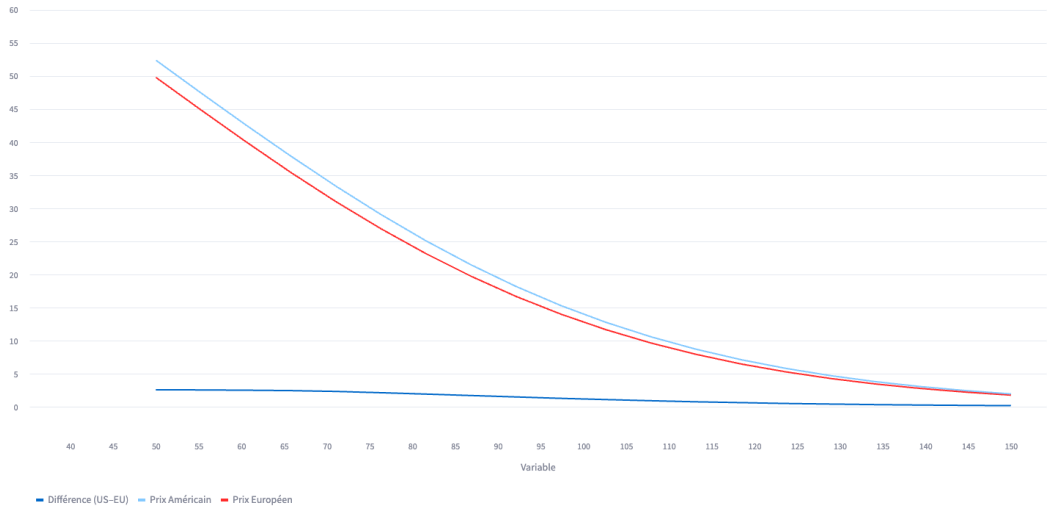
| ARBRE | |
|---|---|
| Date de pricing | 01/09/2025 |
| Nombre de pas | 950 |
| Δt | 0,001052632 |
| Alpha | 1,017001452 |

| METHODE DE PRICING | |
|---|---|
| Méthode | Backward |
| Langage | Python |

| RESULTAT | | |
|---|---|---|
| | Arbre | Black - Scholes |
| Prix d'option | 13,2925 | 13,2925 |
| Temps du calcul | 1.8761 s | 0.0003 s |
| Erreur Tree < | 0,0046 EUR | |

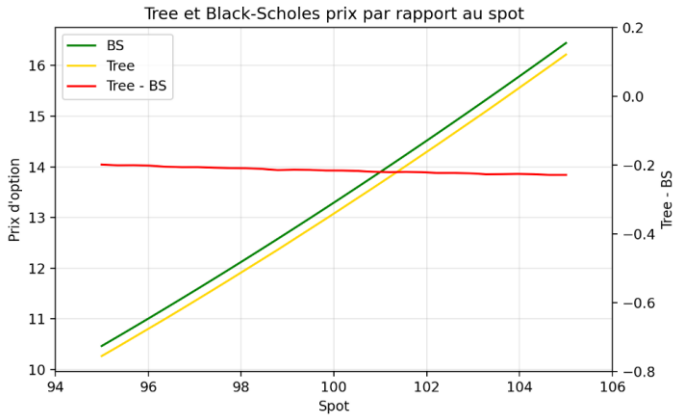
## 2) Call avec dividende : Américain ≥ Européen

Dans le graphique, on essaie de faire varier le strike.

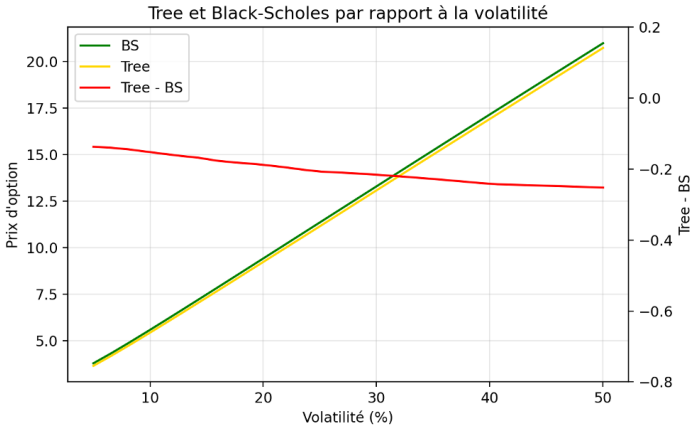



## 3) Le prix augmente quand :

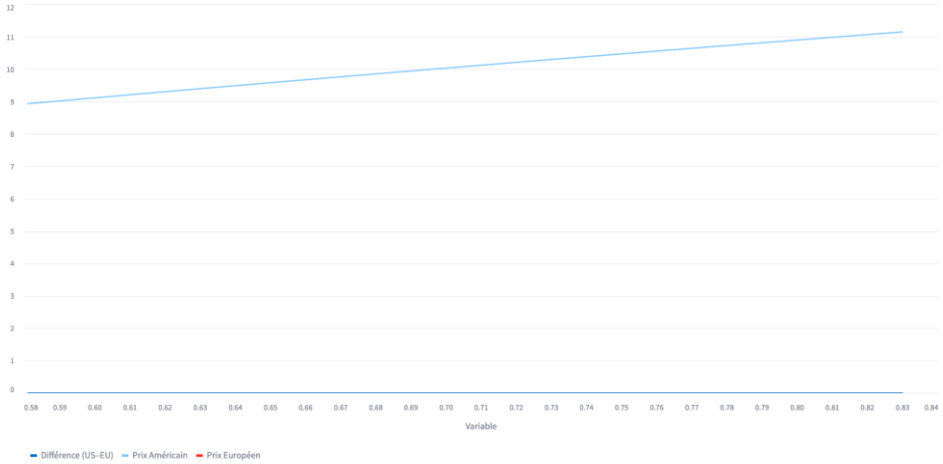
Le spot $S_0$ augmente :




La volatilité $\sigma$ augmente :

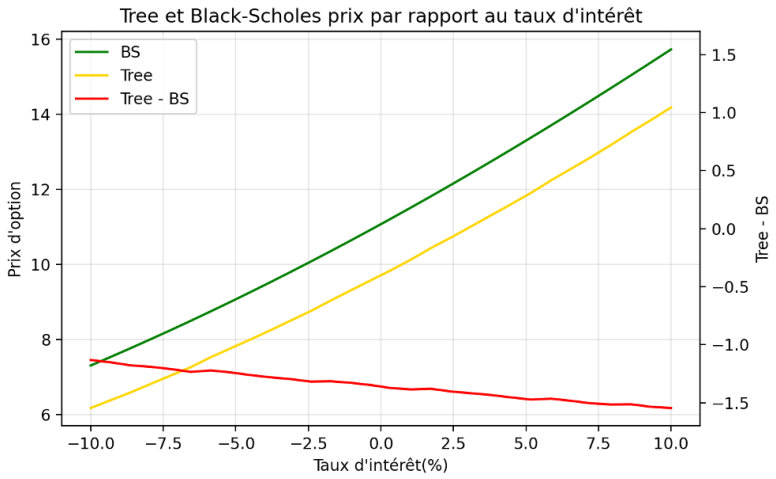



La date de maturité $T$ s'éloigne :



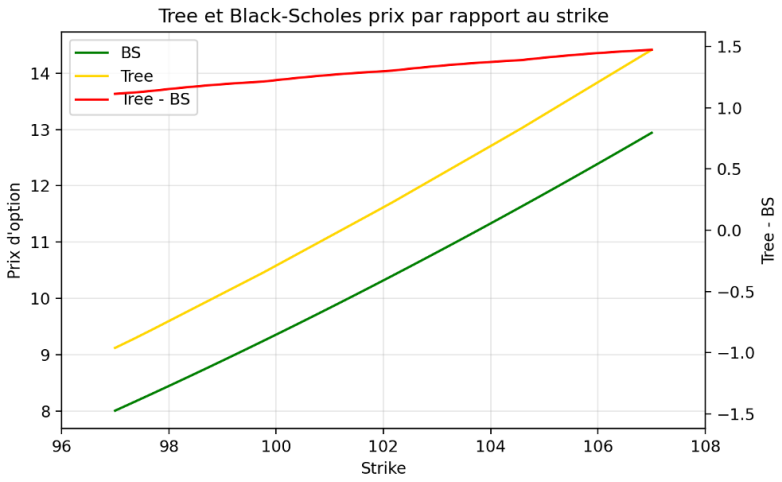

## 4) Le prix des Call :

Augmente quand le taux d'intérêt $r$ augmente :

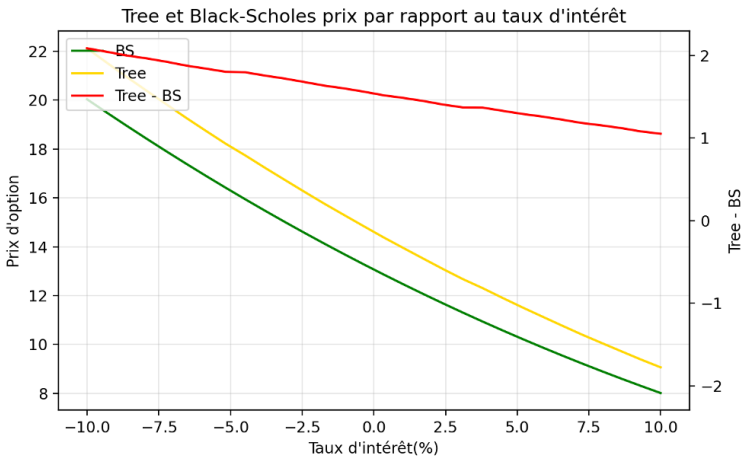


## 5) Le prix des Put :

Augmente avec le prix d'exercice $K$ :




Diminue quand le taux d'intérêt $r$ augmente :




Ainsi, nos prix vérifiant ces tests, on peut donc affirmer que notre pricer fonctionne correctement.

Nous pouvons donc passer à l'étude du comportement de ce dernier quand les différents paramètres varient.

# VI. Etudes

Nous allons maintenant étudier le comportement de notre pricer et de ses fonctionnalités selon différentes configurations.

## A. Le pruning

**Gain en vitesse d'exécution :**

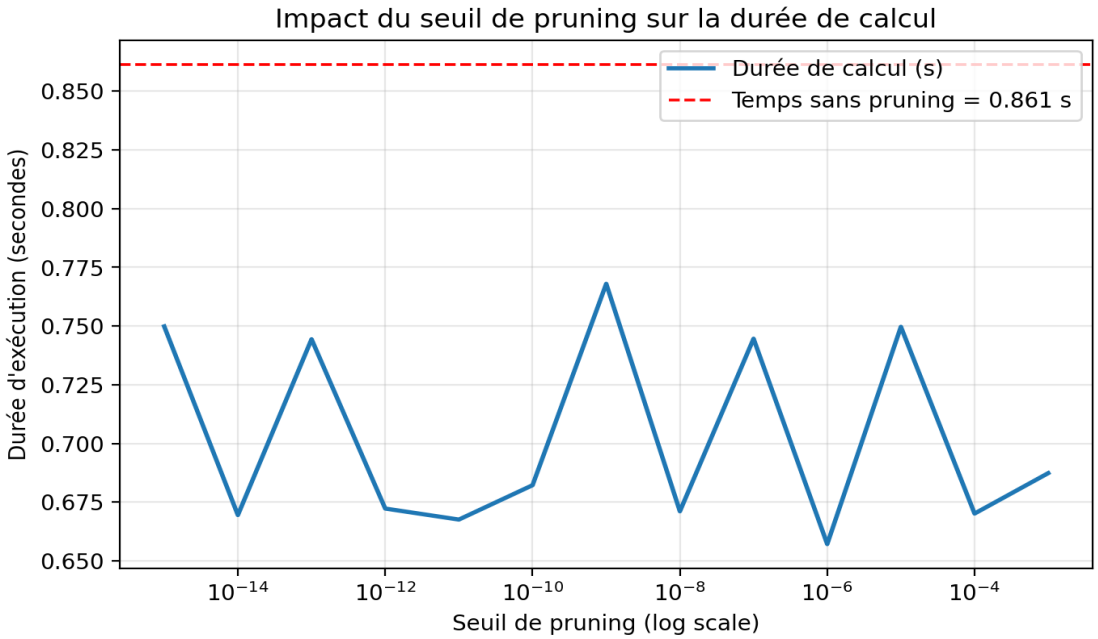
Sans pruning : Le pricer met 5,92 secondes pour calculer le prix avec N = 1000

| RESULTAT | |
|---|---|
| | Arbre |
| Prix d'option | 13,2920 |
| Temps du calcul | 5,921938 |

Avec pruning (seuil $= 1 \cdot E^{-7}$) : Le temps d'exécution avec les mêmes paramètres tombe à 2,05 secondes.

| RESULTAT | |
|---|---|
| | Arbre |
| Prix d'option | 13,29197267 |
| Temps du calcul | 2.0563 s |

Le pruning permet donc de diviser par 2,9 notre temps d'exécution. Mais quel est l'impact sur le prix final ?




*Pour N = 500*

Quand on fait le test de pruning sur plusieurs seuils, on obtient le graphique ci-dessus. Le principe du pruning est de supprimer les nœuds dont la probabilité d'atteinte est très faible, afin de réduire le nombre de calculs nécessaires et ainsi accélérer considérablement le temps d'exécution du modèle.

En théorie, lorsque le seuil de pruning augmente, le temps d'exécution devrait diminuer de manière monotone. Cependant, le graphique ci-dessus montre un comportement non strictement décroissant : certaines valeurs de seuil produisent des durées légèrement plus longues et d'autres plus faibles.

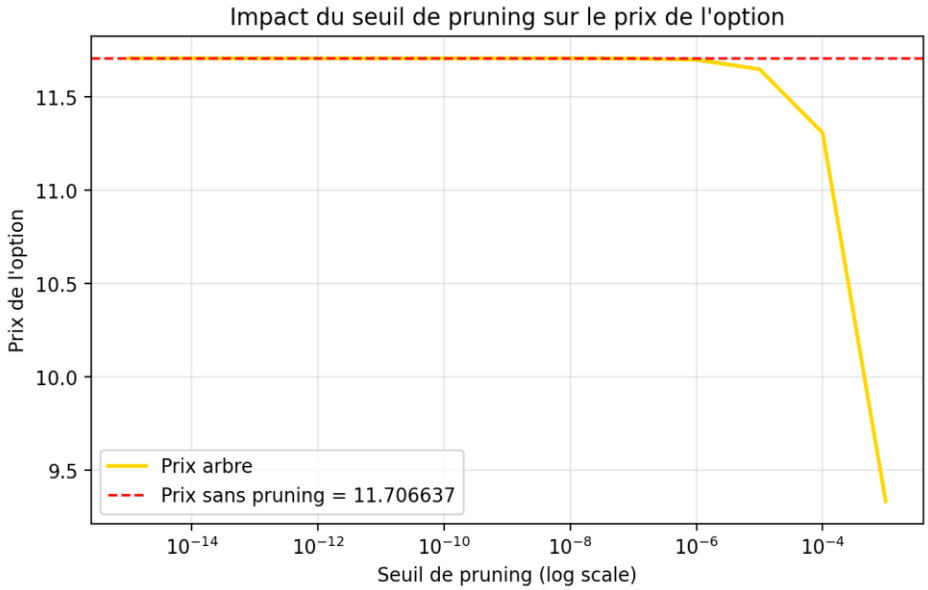
**Perte en précision :**

Malheureusement, plus le seuil de pruning est élevé, plus l'impact sur le prix est grand.

Voilà l'impact sur le prix selon différents seuils de pruning :

| Seuil | Prix Tree | Durée (s) |
|---|---|---|
| 1,00E-03 | 9,333072 | 2,355 |
| 1,00E-04 | 11,307277 | 2,503 |
| 1,00E-05 | 11,648276 | 2,155 |
| 1,00E-06 | 11,698815 | 2,269 |
| 1,00E-07 | 11,705639 | 2,431 |
| 1,00E-08 | 11,706515 | 2,455 |
| 1,00E-09 | 11,706622 | 2,362 |
| 1,00E-10 | 11,706635 | 2,339 |
| 1,00E-11 | 11,706637 | 2,544 |
| 1,00E-12 | 11,706637 | 2,355 |
| 1,00E-13 | 11,706637 | 2,270 |
| 1,00E-14 | 11,706637 | 2,461 |
| 1,00E-15 | 11,706637 | 2,163 |

*Tableau montrant l'effet du pruning sur la précision et le temps d'exécution, pour N = 1000*



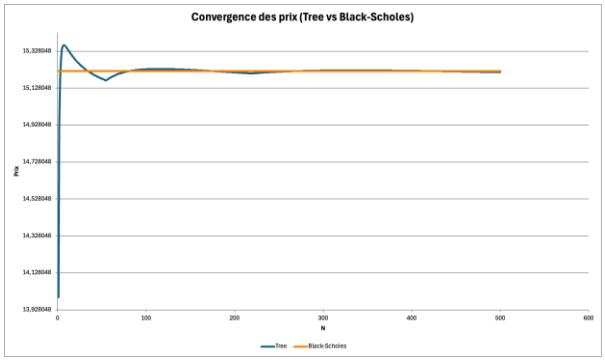

*Graphique montrant l'évolution du prix en fonction du seuil de pruning (N = 1000)*

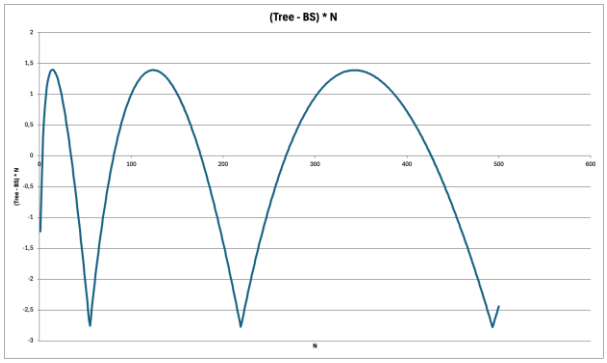
Au vu de ces résultats, on peut donc choisir un seuil de pruning de 1,00E-07 (seuil en rouge : précision au centime près) qui permet à la fois d'obtenir un gain de temps non négligeable, tout en gardant une certaine précision sur les prix. Mais si la précision demandée est importante, alors les prix restent inchangés à $10^{-6}$ pour tout seuil de pruning inférieur à 1,00E-11 (seuil en jaune).

## B. Convergence vers les prix Black Scholes

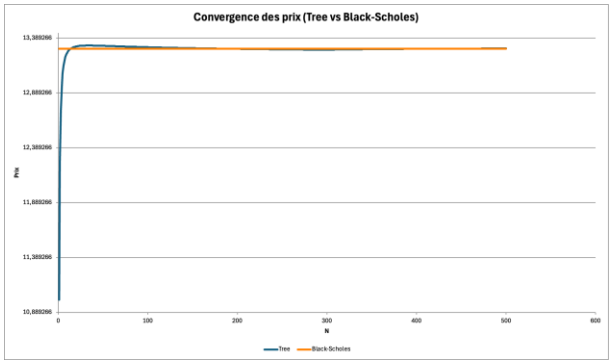
Ici, on teste avec N = 500 et sans dividende :

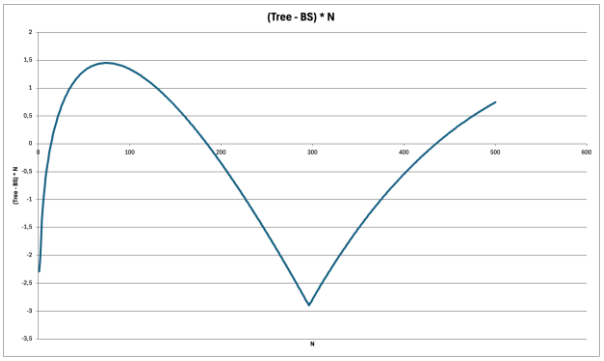
- Dans le cas **K < S (in the money)**




- Dans le cas **K > S (out of the money)**



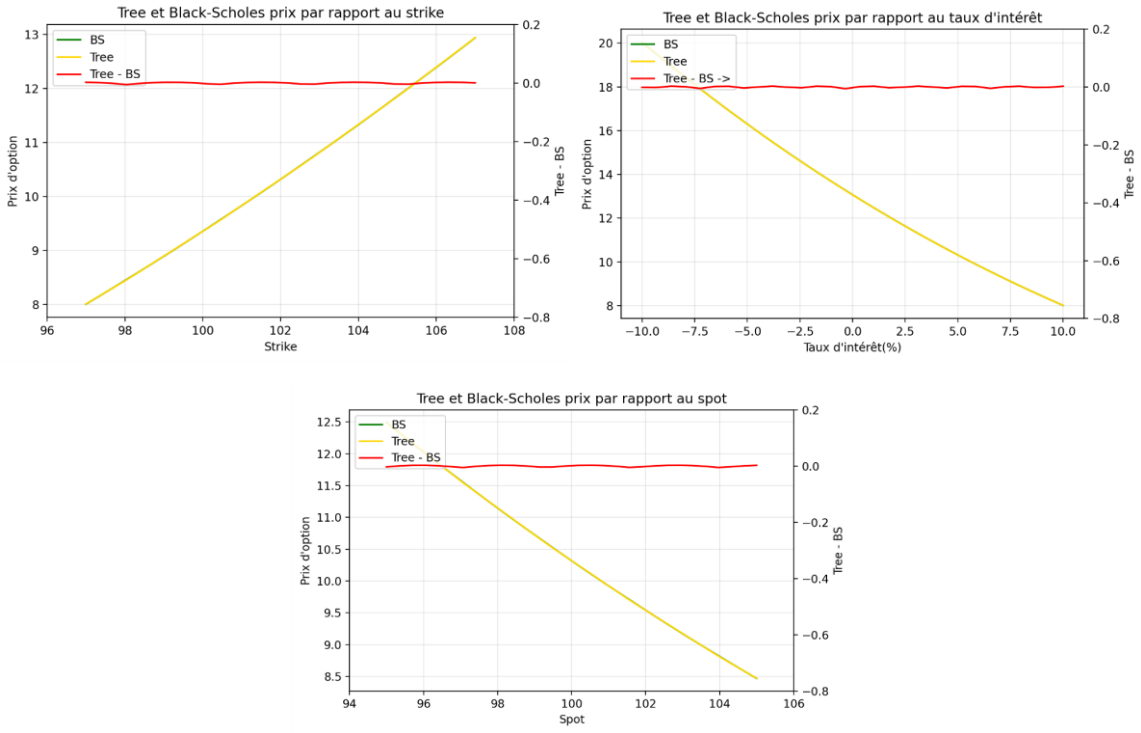

**Comparaison du cas d'une option dans la monnaie (ITM) et hors de la monnaie (OTM) :**

- ITM : Les écarts $(Tree - BS) \times N$ oscillent autour de zéro, formant une **onde quasi-périodique.**
- OTM : L'écart $(Tree - BS) \times N$ reste oscillant, mais avec une amplitude légèrement inférieure et une fréquence différente

On constate que dans le cas ITM, le prix est bien plus volatil autour de Black-Scholes pour $N < 100$, tandis que dans le cas OTM, le prix se stabilise beaucoup plus vite, dès $N > 50$. Dans les deux cas, le prix converge avec une précision suffisante quand $N > 300$.

## C. Tests de Sensibilité de Prix sur les paramètres

**On considère le cas sans dividendes :**




On observe que la variation du strike, du spot ou du taux d'intérêt fait apparaître une **différence quasi-cyclique** entre le prix issu de l'arbre trinominal et celui du modèle de Black-Scholes. Dans les trois cas, cette différence **oscille faiblement autour de zéro**, traduisant une bonne précision numérique, bien que **la fréquence des oscillations diffère** selon le paramètre considéré.

## D. Comparaisons des Grecques Black Scholes vs. Arbre Trinomial

| | Black - Scholes | | | Arbre Trinominal | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Gap BS: VBA vs Python | | | | | Gap Recursive : VBA vs Python | | | Gap Backward : VBA vs Python |
| | Python | VBA | | Python Rec | VBA Rec | | Python Back | VBA Back | |
| Delta | 0,5989607 | 0,54333 | 0,055630358 | 0,551627738 | 0,562077 | -0,010449242 | 0,551627738 | 0,562077 | -0,010449242 |
| Gamma | 0,0128868 | 0,012759 | 0,000127387 | 0,061973137 | 0,038156 | 0,023816673 | 0,061973137 | 0,038156 | 0,023816673 |
| Vega | 38,660444 | 38,27828 | 0,382159565 | 39,30087508 | 38,26439 | 1,036482967 | 39,30087508 | 38,26439 | 1,036482967 |
| Theta | -8,129245 | -6,24942 | -1,879820558 | -7,689728534 | -7,940649 | 0,250920686 | -7,689728534 | -7,94065 | 0,250920686 |
| Rho | 46,603565 | 42,75345 | 3,850112139 | 46,05671358 | 46,44236 | -0,385648114 | 46,05671358 | 46,44236 | -0,385648114 |
| Vanna | 0,0635862 | 0,190552 | -0,12696573 | 0,201477436 | 0,196833 | 0,004643954 | 0,201477436 | 0,196833 | 0,004643954 |
| Vomma | -1,593839 | -2,87082 | 1,276976755 | -3,061042881 | -2,865368 | -0,195674601 | -3,06104288 | -2,86537 | -0,1956746 |

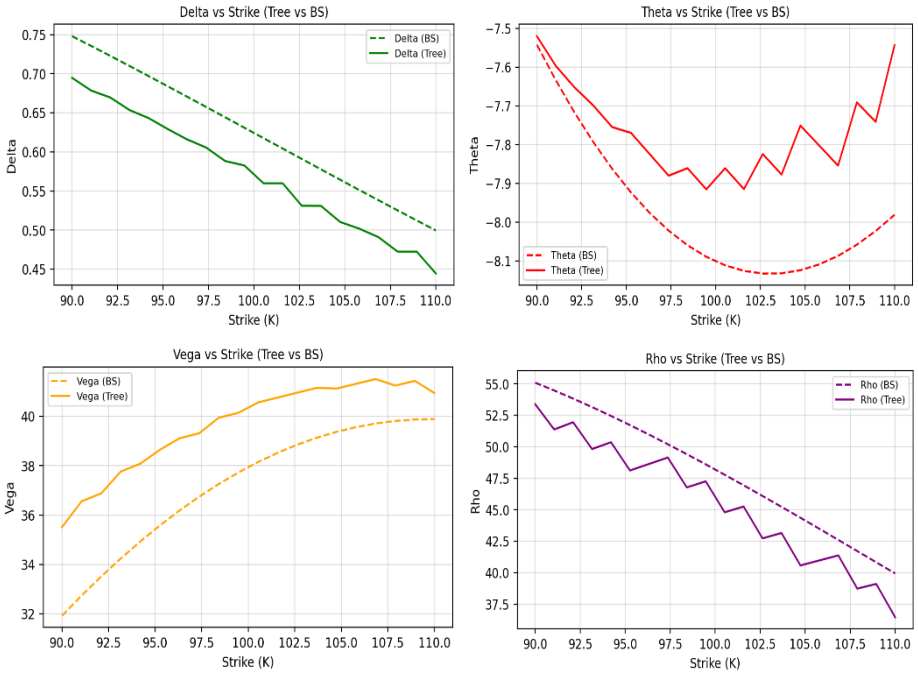
| BS Vs Rec | | BS Vs. Backward | |
|---|---|---|---|
| Python | VBA | Python | VBA |
| 8% | 3% | 8% | 3% |
| 381% | 199% | 381% | 199% |
| 2% | 0% | 2% | 0% |
| 5% | 27% | 5% | 27% |
| 1% | 9% | 1% | 9% |
| 217% | 3% | 217% | 3% |
| 92% | 0% | 92% | 0% |

*Ecart relatif par rapport à Black Scholes*

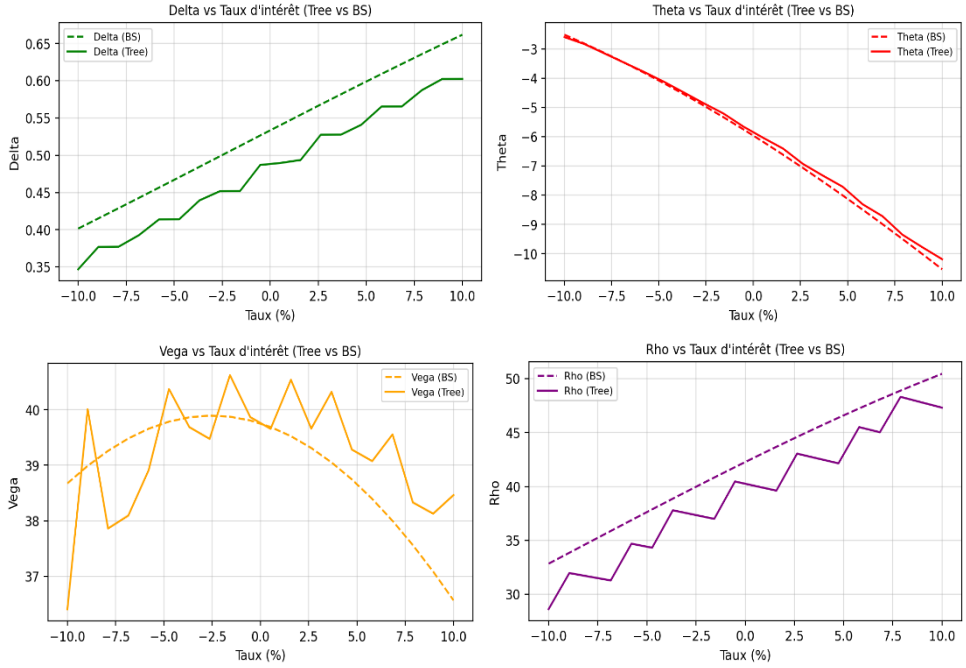
Nos grecques calculées par différences finies sont assez proches en valeur relative, sauf pour les Grecques du second ordre, notamment Gamma et Vanna, mais cela s'explique car nos paramètres sont ceux d'une option proche de la monnaie (2 € d'écart) et l'évolution est très rapide pour les Grecques du second ordre.

## E. Tests de Sensibilité de Grecques sur les paramètres

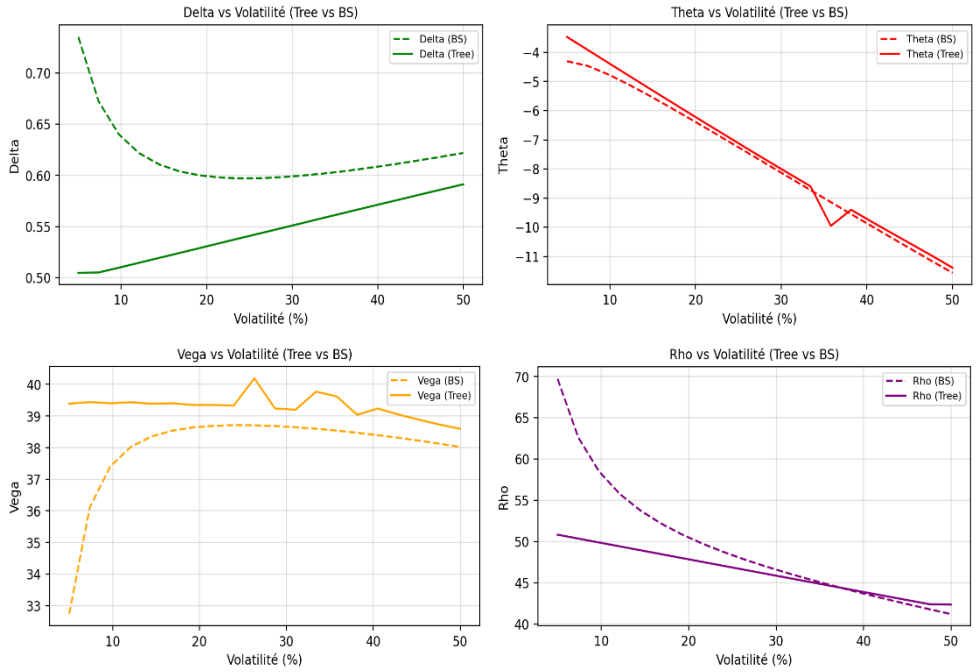
Comportement des Grecques lors d'une variation sur le prix d'exercice :




Comportement des Grecques lors d'une variation sur le taux :



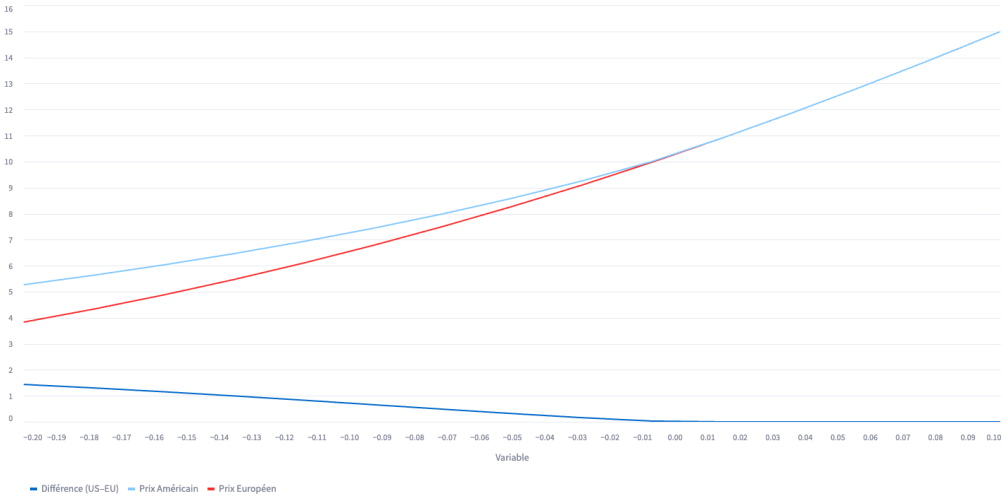
Comportement des Grecques lors d'une variation sur la volatilité :



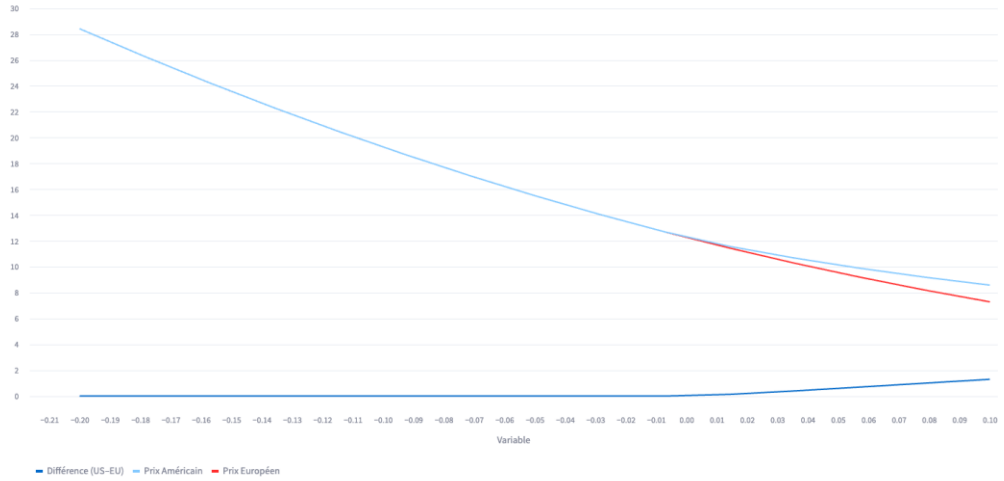

Le comportement des Grecques du premier ordre colle avec le comportement des Grecques calculées par Black & Scholes. Toutefois, on constate que le comportement de Delta, Theta, Vega et Gamma n'est pas aussi calé sur Black & Scholes que les deux autres tests.

## F. Comparaison des option Américaines et Européennes

On considère le cas sans dividende avec la variation de taux d'intérêt :




- Lorsque le taux d'intérêt est **négatif**, le prix de l'option **américaine** est systématiquement **supérieur** à celui de l'option **européenne**, car la possibilité d'exercer anticipativement a davantage de valeur dans un environnement de taux bas.
- À mesure que le taux **augmente**, les deux prix augmentent également, mais **l'écart se réduit** : le prix américain **tend vers** le prix européen, et les deux deviennent **égaux lorsque le taux est positif**, l'avantage de l'exercice anticipé disparait alors.




- Lorsque le taux d'intérêt est **négatif**, le prix de l'option **américaine** reste **proche** du prix **européen**, voire **légèrement inférieur**.
- En revanche, lorsque le taux **augmente**, les deux prix **diminuent**, mais **le prix américain devient supérieur** au prix européen. Cet écart croissant s'explique par la **valeur supplémentaire** liée à la possibilité d'exercer l'option **avant l'échéance**, particulièrement avantageuse lorsque les taux sont positifs.

# VII. Difficultés Rencontrées

Au cours du développement du pricer, plusieurs difficultés techniques ont été rencontrées.

## A. Inclusion de dividende :

L'intégration des dividendes dans le modèle trinominal a constitué une difficulté majeure. En effet, le modèle de base repose sur l'hypothèse d'un sous-jacent suivant une dynamique continue, sans saut de prix. Or, le versement d'un dividende introduit une **discontinuité** : à la date ex-dividende, le prix du sous-jacent chute instantanément du montant distribué.
Cette rupture de continuité perturbe la construction du tronc central de l'arbre et biaisait le calcul des probabilités locales, fondées sur la correspondance entre l'espérance et la variance conditionnelles du processus. Pour gérer le problème, deux ajustements principaux ont été mis en œuvre:

- **Recentrage du nœud médian** : à chaque pas de temps, la position du nœud central du niveau suivant est ajustée de manière à rester alignée avec la valeur espérée du sous-jacent après prise en compte du dividende. Cela garantit la recombinaison correcte de l'arbre et la stabilité des probabilités locales.

```
# Recentrage de la position centrale si l'espérance sort des bornes
S_up = S_mid * a
S_down = S_mid / a
lower = 0.5 * (S_mid + S_down)
upper = 0.5 * (S_mid + S_up)
shifts = 0

while (E > upper or E < lower) and shifts < 10:
    if E > upper:      You, la semaine dernière • separate …
        kprime += 1
        S_mid *= a
    else:
        kprime -= 1
        S_mid /= a
    S_up = S_mid * a
    S_down = S_mid / a
    lower = 0.5 * (S_mid + S_down)
    upper = 0.5 * (S_mid + S_up)
    shifts += 1
```

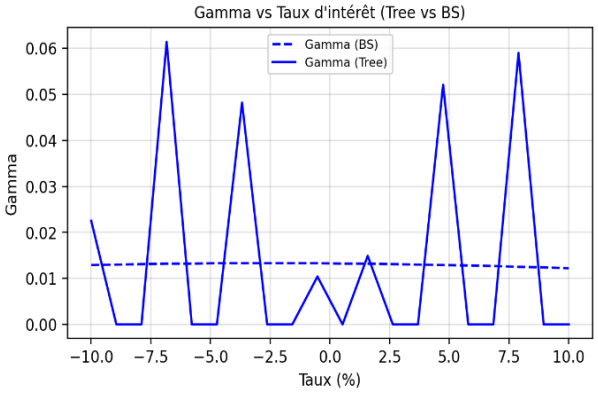
- **Contrôle des variances aberrantes** : dans certaines configurations (forte volatilité, pas de temps trop court ou dividende élevé), la variance calculée localement pouvait devenir numériquement instable, entraînant des probabilités négatives ou supérieures à 1. Pour éviter ces dérives, des bornes minimales ont été imposées sur la variance et les probabilités ont été normalisées afin d'assurer leur cohérence et leur somme unitaire

```
if V < 1e-18:
    base_next = trunk_next
    if base_next < MIN_P:
        base_next = max(MIN_P, S_i_k * exp_r_dt)
    denom = loga if abs(loga) > EPS else EPS
    kprime = int(round(math.log(max(E, MIN_P) / base_next) / denom))
    S_mid = base_next * (a ** kprime)
```

## B. Les Grecques de second ordre :

Lors du codage des Grecques, nous avons rencontrés plusieurs problèmes, notamment avec celles du second ordre.

Tandis que la plupart de nos grecques convergent vers Black Scholes lorsqu'on fait varier les paramètres, les Grecques de second ordre et tout particulièrement Gamma oscillaient autour de la courbe Black & Scholes.




*Exemple avec Gamma quand r varie*

On a donc essayé différentes configurations : avec ou sans pruning, avec un $N$ plus ou moins grands, rien ne changeait.

On pourrait corriger cette courbe par interpolation des points autour d'une valeur moyenne entre deux points voisins, mais nous avons préféré garder un vrai reflet de nos grecques du second ordre.

# VIII. Conclusion

A travers ce projet, nous avons pu apprendre à utiliser un arbre trinomial et comment le coder tout en tenant compte de la mémoire et de l'impact des dividendes sur le comportement des probabilités.

Le pricing par arbre trinomial présente certains avantages, notamment celui de pouvoir modéliser l'exercice anticipé dans le cas d'option américaine, mais ce modèle présente également des limites.

- **Calcul et de gestion de la mémoire :**

La puissance de calcul augmente fortement avec le nombre de pas (rapport de $N^2$), cela devient vite lent voire-même impossible du fait de la saturation de la mémoire. Or, la convergence vers Black Scholes s'obtient justement quand $N$ est grand ($N > 1000$ pour une erreur de $\pm\ 0.01\%$).

- **Valeurs extrêmes sur les paramètres :**

Aussi, des probabilités négatives peuvent survenir lors d'utilisation de dividendes élevés ou de volatilité quasi faible, mais ces cas étant rares sur les marchés, cela pose moins un problème.

- **Dividendes discrets :**

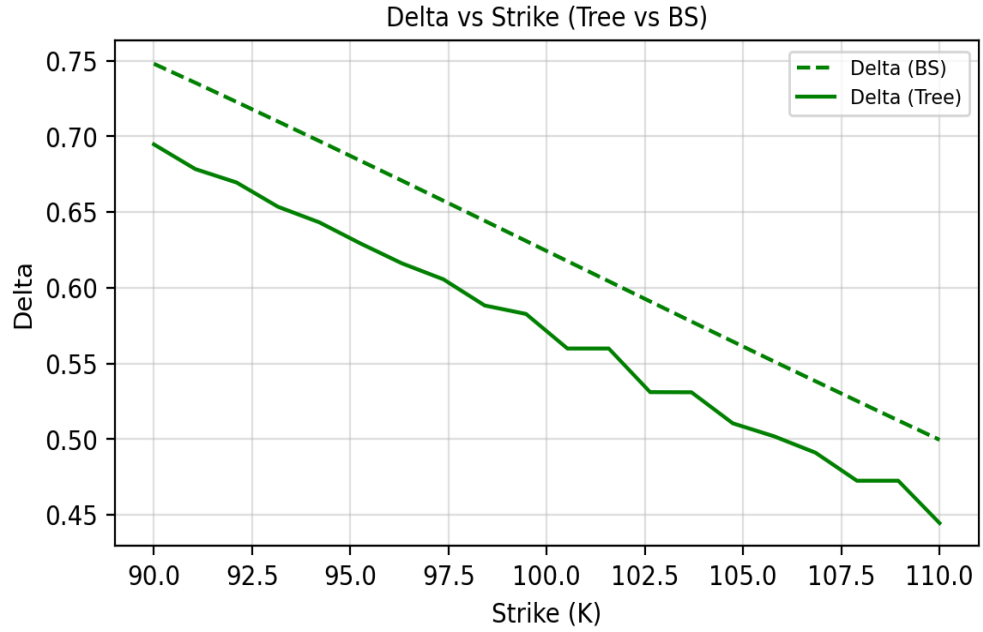
Enfin, une certaine tolérance de date ex-div est demandée pour éviter une erreur si elle ne coïncide pas avec les pas de l'arbre.

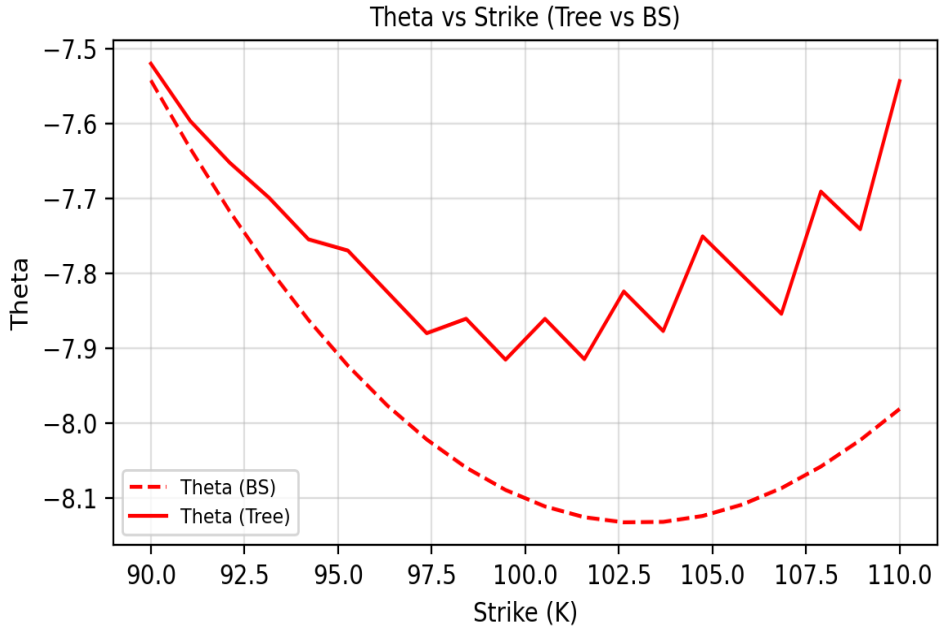
Pour améliorer les performances de ce pricer, son temps d'exécution, on pourrait utiliser la librairie Python « Cudda » qui permet d'allouer la charge de calcul à la carte graphique (et ainsi mettre à profit ma RTX 4070).

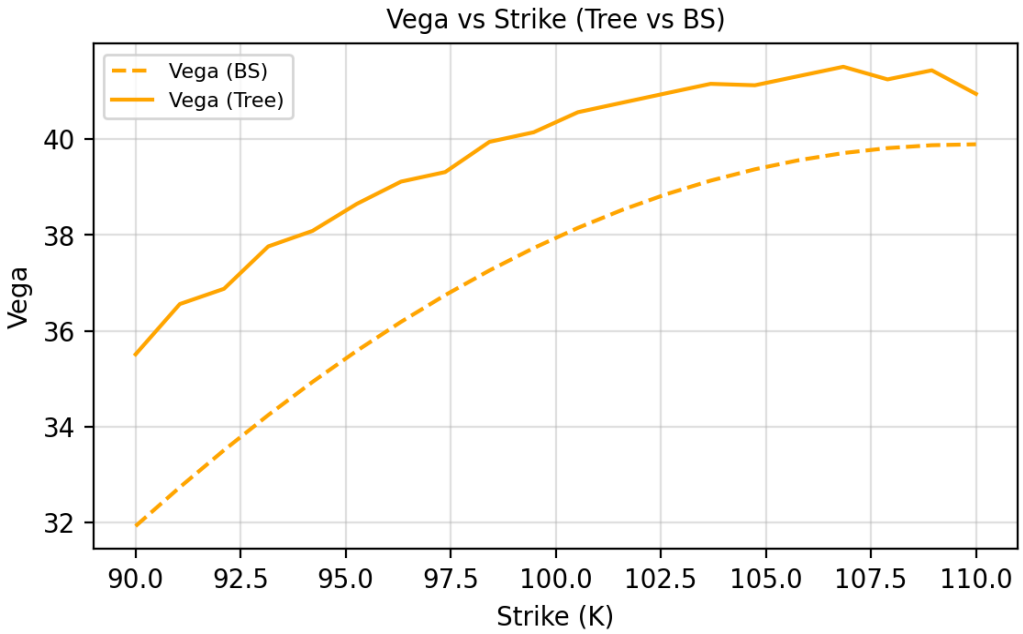
Une telle amélioration permettrait d'aller chercher rapidement des N très grands ($> 100\ 000$) pour atteindre une précision ($Error\ Tree$) de 1 € pour 22 790 options.

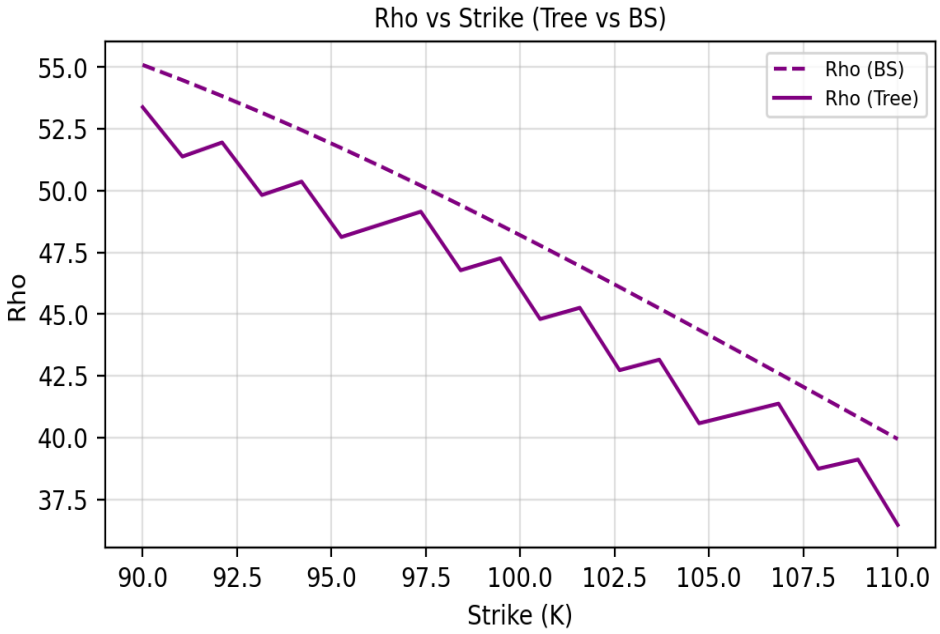
Actuellement notre précision est de 1 € pour 2280 options (0.0004 € / option) et 309 secondes de temps d'exécution. Ce qui, en plus d'être lent, est insuffisant dans le cadre d'une banque qui vend des options par paquet de 10 000 à 100 000 options (selon le type de client : entreprises, institutionnels, …).

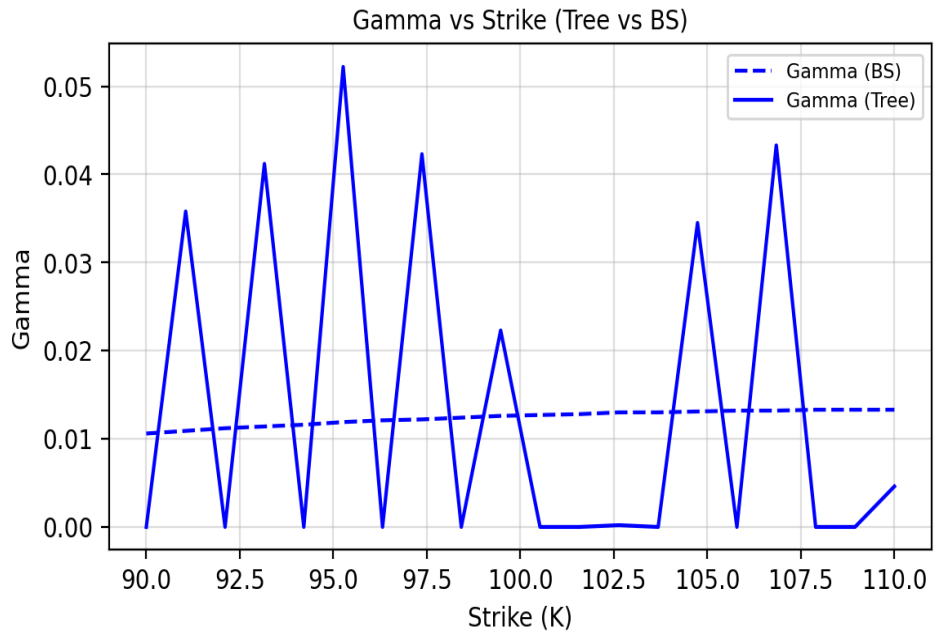
**Greeks Strikes :**

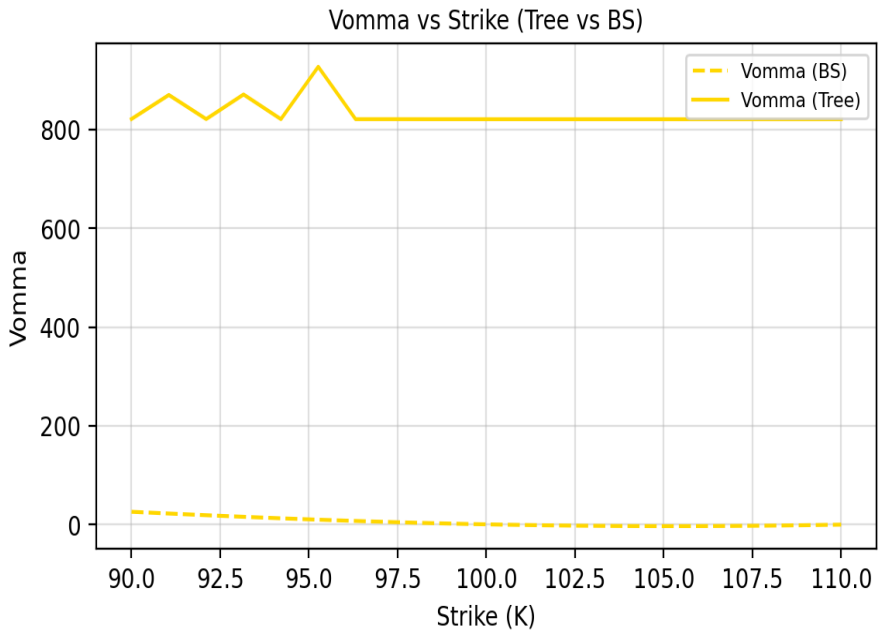


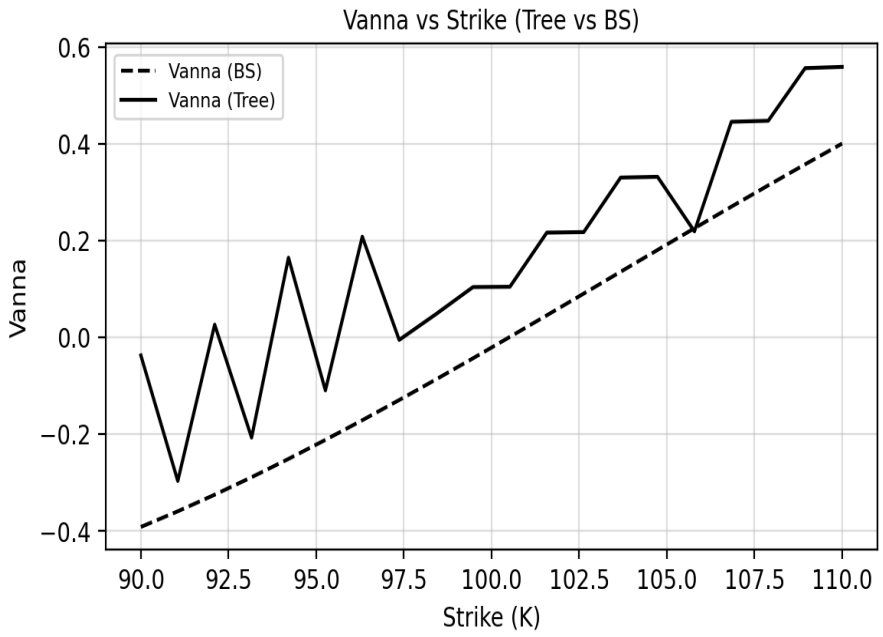




Vega vs Strike (Tree vs BS)
Vega (BS)
Vega (Tree)
Vega
40
38
36
34
32
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Strike (K)

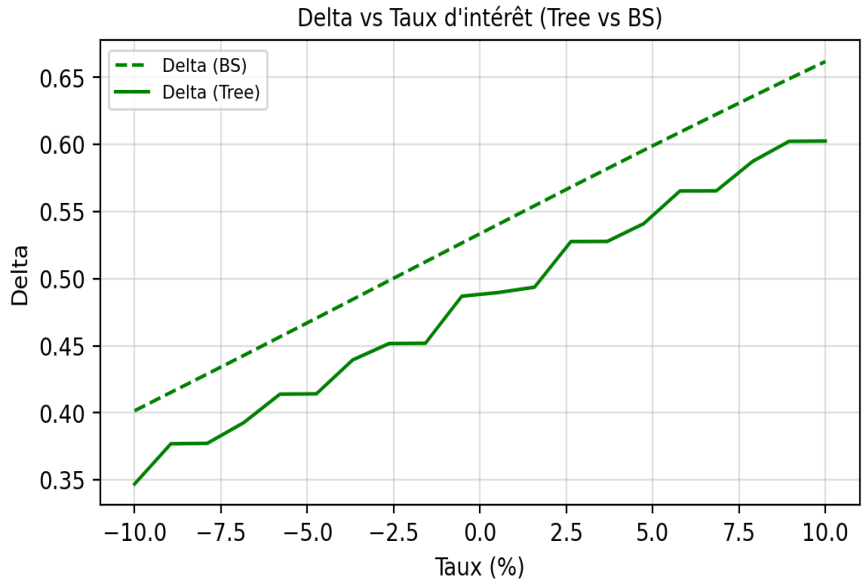

Rho vs Strike (Tree vs BS)
Rho (BS)
Rho (Tree)
Rho
55.0
52.5
50.0
47.5
45.0
42.5
40.0
37.5
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Strike (K)

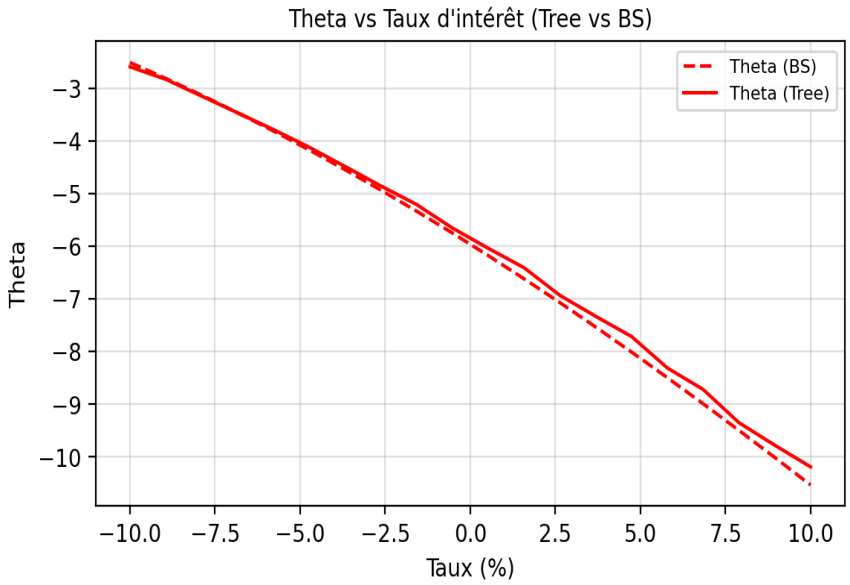

Gamma vs Strike (Tree vs BS)
Gamma (BS)
Gamma (Tree)
Gamma
0.05
0.04
0.03
0.02
0.01
0.00
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Strike (K)

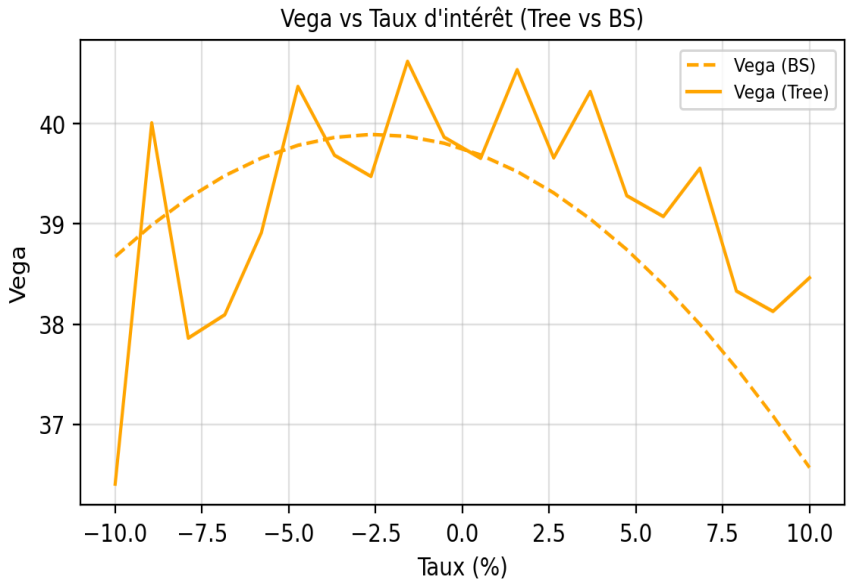

Vomma vs Strike (Tree vs BS)
Vomma (BS)
Vomma (Tree)
Vomma
800
600
400
200
0
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0
Strike (K)

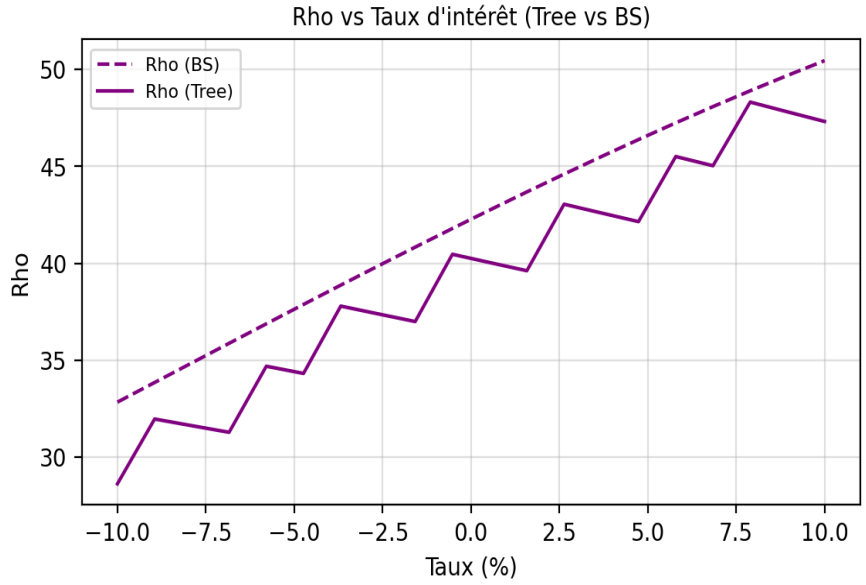

Vanna vs Strike (Tree vs BS)
Vanna (BS)
Vanna (Tree)
Vanna
Strike (K)
0.6
0.4
0.2
0.0
−0.2
−0.4
90.0
92.5
95.0
97.5
100.0
102.5
105.0
107.5
110.0

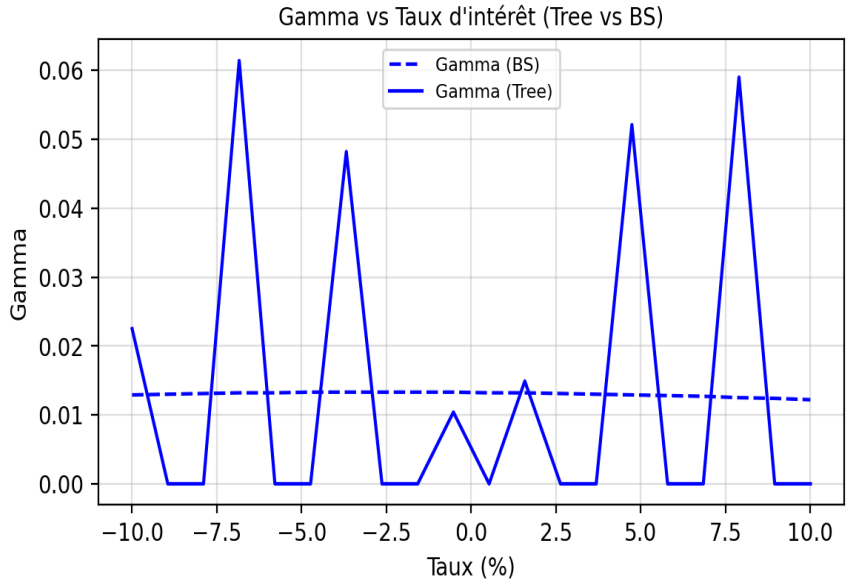
**Greeks Rate :**

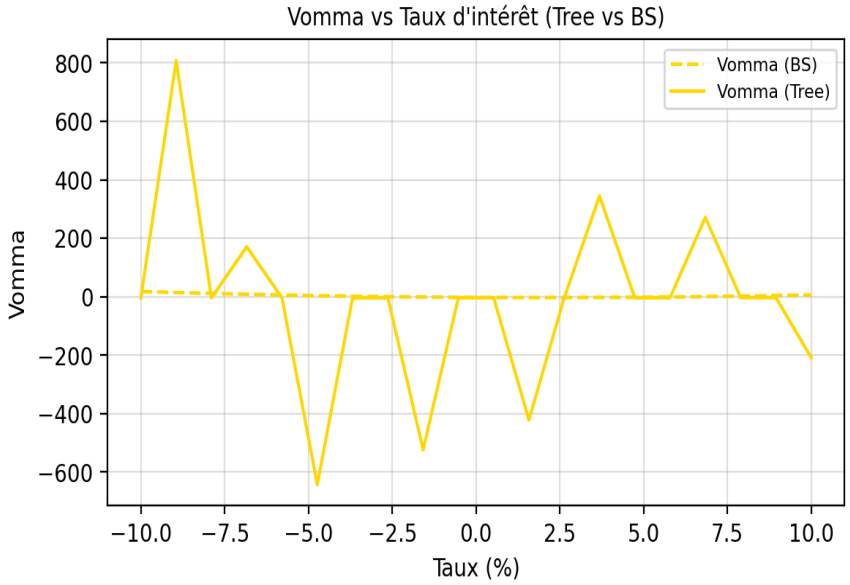

Delta vs Taux d'intérêt (Tree vs BS)
Delta (BS)
Delta (Tree)
Delta
0.65
0.60
0.55
0.50
0.45
0.40
0.35
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

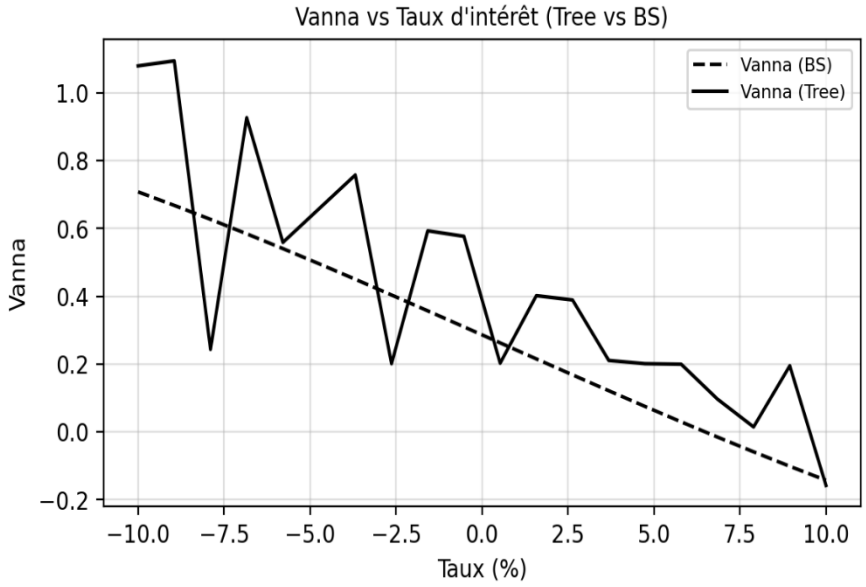

Theta vs Taux d'intérêt (Tree vs BS)
Theta (BS)
Theta (Tree)
Theta
−3
−4
−5
−6
−7
−8
−9
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)


Vega vs Taux d'intérêt (Tree vs BS)
Vega (BS)
Vega (Tree)
Vega
40
39
38
37
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

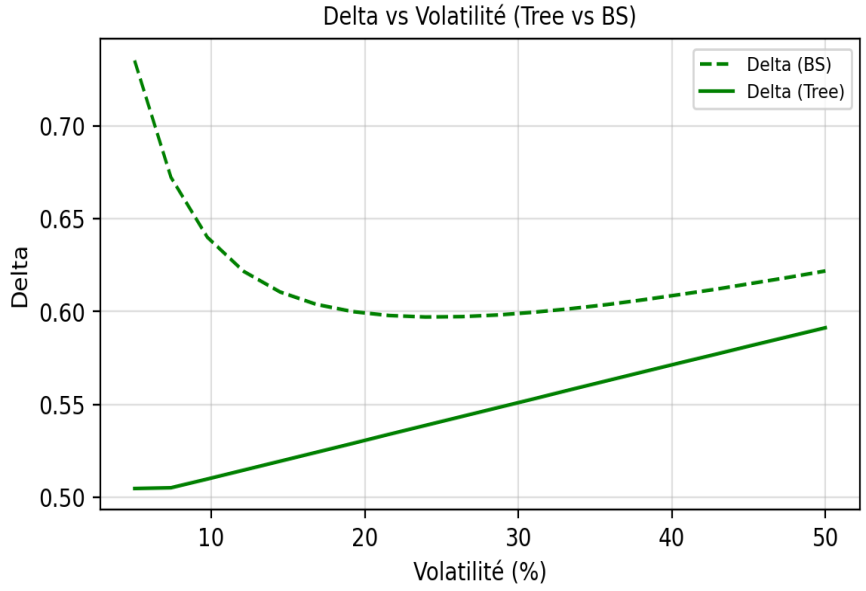

Rho vs Taux d'intérêt (Tree vs BS)
Rho (BS)
Rho (Tree)
Rho
50
45
40
35
30
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

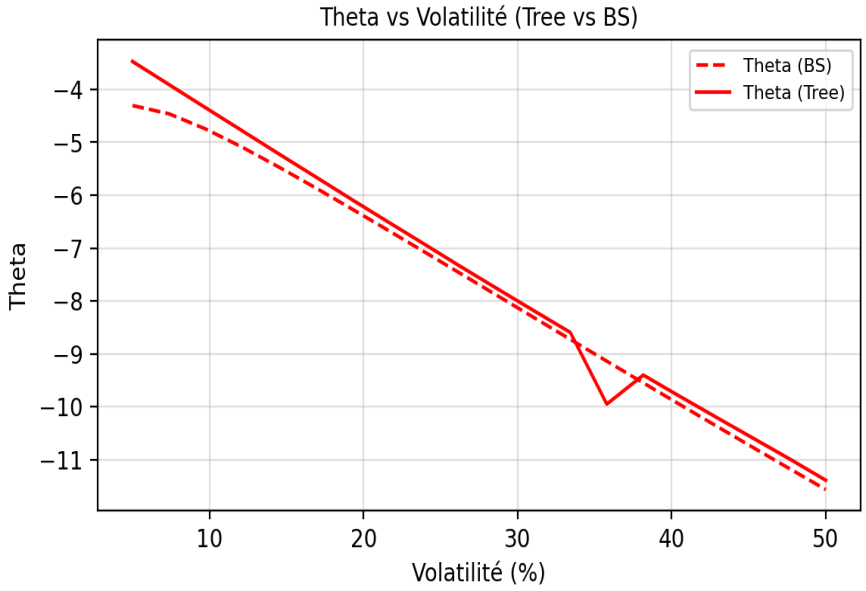

Gamma vs Taux d'intérêt (Tree vs BS)
Gamma (BS)
Gamma (Tree)
Gamma
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

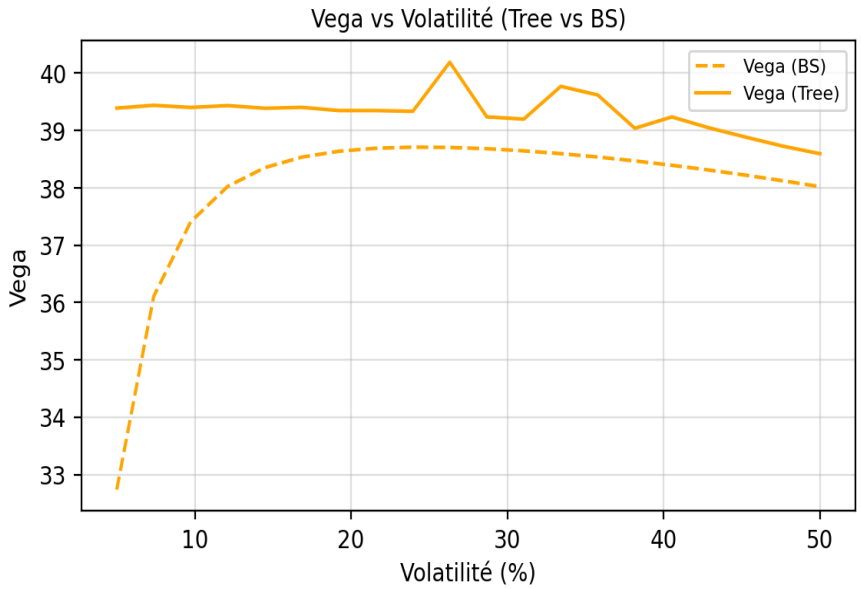

Vomma vs Taux d'intérêt (Tree vs BS)
Vomma (BS)
Vomma (Tree)
Vomma
800
600
400
200
0
−200
−400
−600
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

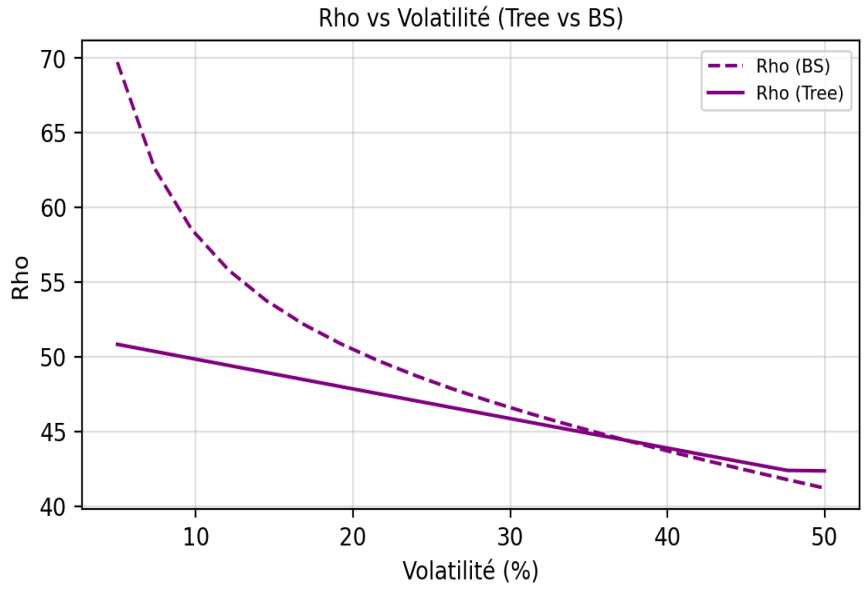

Vanna vs Taux d'intérêt (Tree vs BS)
Vanna (BS)
Vanna (Tree)
Vanna
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
Taux (%)

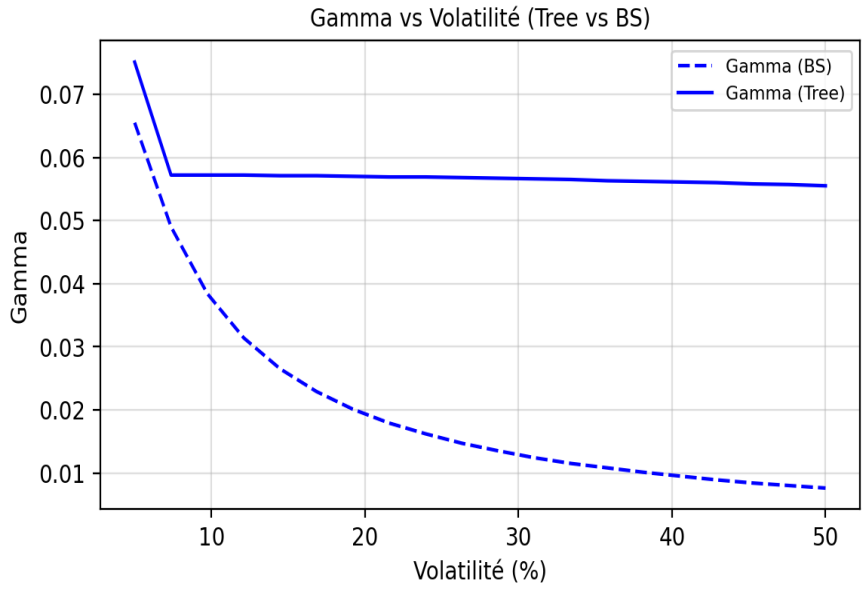
Greeks Vol :

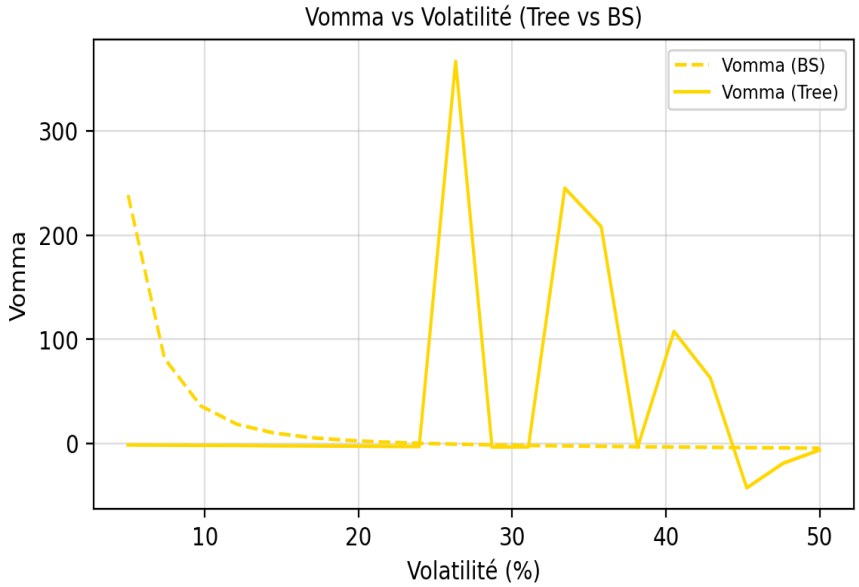

Delta vs Volatilité (Tree vs BS)
Delta (BS)
Delta (Tree)
Delta
0.70
0.65
0.60
0.55
0.50
10
20
30
40
50
Volatilité (%)

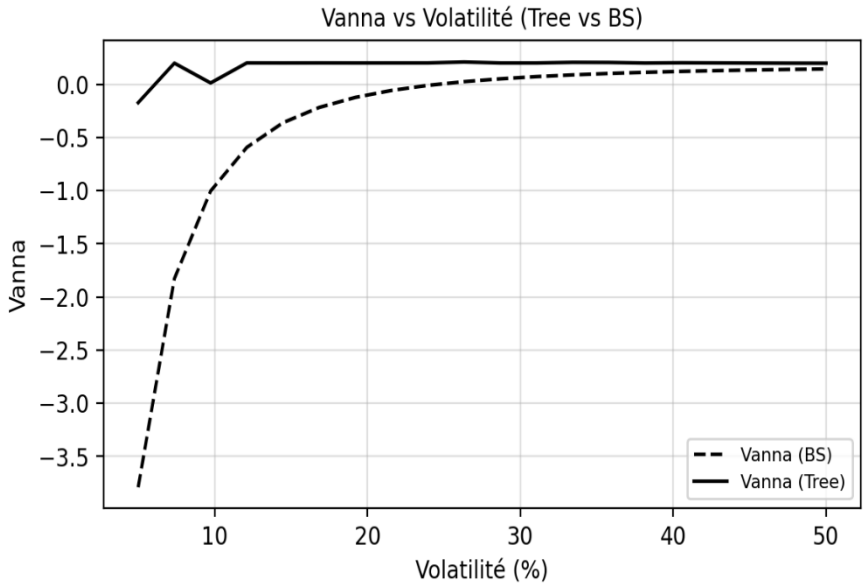

Theta vs Volatilité (Tree vs BS)
Theta (BS)
Theta (Tree)
Theta
−4
−5
−6
−7
−8
−9
−10
−11
10
20
30
40
50
Volatilité (%)


Vega vs Volatilité (Tree vs BS)
Vega (BS)
Vega (Tree)
Vega
Volatilité (%)
40
39
38
37
36
35
34
33
10
20
30
40
50


Rho vs Volatilité (Tree vs BS)
Rho (BS)
Rho (Tree)
Rho
Volatilité (%)
70
65
60
55
50
45
40
10
20
30
40
50


Gamma vs Volatilité (Tree vs BS)
Gamma (BS)
Gamma (Tree)
Gamma
Volatilité (%)
0.07
0.06
0.05
0.04
0.03
0.02
0.01
10
20
30
40
50


Vomma vs Volatilité (Tree vs BS)
Vomma (BS)
Vomma (Tree)
Vomma
300
200
100
0
10
20
30
40
50
Volatilité (%)


Vanna vs Volatilité (Tree vs BS)
Vanna
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
−3.5
Vanna (BS)
Vanna (Tree)
10
20
30
40
50
Volatilité (%)