

MASTER 2 ECONOMIE & INGÉNIERIE FINANCIÈRE - 272  
UNIVERSITÉ PARIS DAUPHINE - PSL

---

## Pricing d'options : Application du modèle Trinomial en Python & VBA

---

*Auteurs :*  
Naïm LEHBIBEN  
Cyprien TARDIVEL

17 novembre 2023

# Table des matières

1	Récapitulatif des Missions et Résultats	1
2	Difficultés Rencontrées	2
3	Extensions	3
4	Annexes	4

# Chapitre 1

## Récapitulatif des Missions et Résultats

Notre projet s'est concentré sur le développement et la mise en œuvre d'un ensemble de tâches précises, visant à renforcer notre capacité à évaluer le prix des options. Voici le détail des avancées majeures que nous avons réalisées :

**Modélisation en Python et VBA :** Nous avons conçu un modèle robuste capable de pricer les options américaines et européennes, avec ou sans l'inclusion de dividendes. Grâce à des méthodes récursives sophistiquées, notre solution s'intègre parfaitement à la fois en Python et en VBA, démontrant une flexibilité et une performance élevées.

**Pruning Dynamique :** Nous avons intégré une fonctionnalité de "pruning" dans notre modèle, permettant d'optimiser l'arbre trinomial en éliminant les branches de probabilités marginales. Cette fonctionnalité est ajustable à volonté par l'utilisateur via une interface Excel, ce qui offre une personnalisation approfondie et une amélioration de la performance du modèle [4.1]. Pour réaliser cette technique, nous avons préféré utiliser la méthode des probabilités totales plutôt que la méthode de l'écart type au nœud médian.

**Visualisation Avancée de l'Arbre Trinomial :** Nous avons mis en place une fonctionnalité avancée de visualisation de l'arbre trinomial dans Excel, permettant de représenter non seulement les prix du sous-jacent à chaque nœud, mais également d'autres métriques telles que les probabilités de transition et les probabilités totales d'atteindre chaque nœud. Cette extension paramétrable dans Python ajoute une dimension analytique précieuse et augmente la transparence de notre modèle.

### Analyse de Convergence :

Notre première analyse graphique illustre la convergence oscillatoire du modèle trinomial vers celui de Black & Scholes en fonction du nombre de pas [4.2]. Cette observation est cruciale car elle montre que notre modèle devient de plus en plus précis avec un nombre accru de pas, malgré des fluctuations périodiques qui s'atténuent avec le temps.

La seconde analyse, représentée graphiquement, révèle l'effet du prix d'exercice sur la convergence du modèle trinomial vers Black & Scholes [4.3]. Lorsque le prix d'exercice approche du spot, nous observons des écarts notables entre les deux modèles, soulignant l'importance du choix du strike dans l'évaluation des options.

Ces travaux d'analyse de convergence valident la fiabilité de notre modèle dans des conditions de marché variables et confirment son utilité en tant qu'outil d'évaluation.

## Chapitre 2

# Difficultés Rencontrées

Au cours de notre projet, nous avons été confrontés à plusieurs obstacles techniques, notamment liés à l'organisation de notre code et à la gestion des performances.

**Python :** Initialement, notre démarche de développement était centralisée autour d'un unique fichier Main, contenant l'intégralité de notre code. Cette approche s'est rapidement avérée peu pratique et non conforme aux principes de la programmation orientée objet, qui privilégie une structure modulaire et décomposée. En reconnaissant cette inadéquation, nous avons procédé à une refonte de notre architecture en séparant notre code en plusieurs fichiers, chacun correspondant à une classe spécifique de notre modèle.

Cette réorganisation, bien que bénéfique pour la lisibilité et la maintenance du code, n'a pas été sans conséquences. Nous avons été confrontés à des problèmes de dépendances cycliques, un écueil courant dans la programmation orientée objet lorsque les modules s'importent mutuellement de manière récursive. Résoudre ces dépendances a été un processus complexe qui, in fine, a impacté les performances de notre solution de pricing Python. La capacité du modèle à effectuer des calculs rapides s'est vue réduite, passant d'une possibilité de 2000 pas à environ 1000 pas, marquant ainsi une diminution significative de notre capacité à simuler des scénarios de prix sur des arbres trinomiaux très détaillés.

**VBA :** Pour VBA, nous avons rencontré des difficultés similaires. Malgré les optimisations, la performance du pricing demeure limitée à 500 pas. Bien que cette limitation soit suffisante pour obtenir un prix qui converge vers une valeur correcte - et donc reste opérationnelle pour la plupart des applications pratiques - elle reste en deçà de la capacité théorique de la méthode récursive, qui peut gérer jusqu'à environ 2500 pas.

Ces problématiques ont été des points d'apprentissage cruciaux pour notre équipe, mettant en évidence l'importance d'une conception soignée et d'une planification anticipée, surtout lorsqu'il s'agit de projets de développement logiciel complexe.

## Chapitre 3

# Extensions

L'une des améliorations significatives que nous avons apportées à notre modèle est le calcul des Grecques en Python. Traditionnellement, ces indicateurs de risque sont obtenus à l'aide du modèle de Black & Scholes, qui, toutefois, n'offre pas la possibilité de prendre en compte les dividendes lors du pricing des options. Afin de surmonter cette limitation, notre équipe a innové en adoptant une approche numérique approximative utilisant notre modèle trinomial.

Nous avons procédé au calcul de deux prix d'options distincts, chacun ajusté par une variation infime d'un paramètre spécifique. À partir de ces valeurs, nous avons dérivé une pente, qui nous a permis d'estimer la sensibilité de l'option vis-à-vis de ce paramètre. Cette approche nous a donné des résultats proches de ceux obtenus par le modèle de Black & Scholes, validant ainsi la précision de notre méthode.

Plus impressionnant encore, cette technique peut être généralisée pour évaluer les sensibilités d'ordre supérieur, telles que la Vanna, la Vomma, ou la Volga, pour lesquelles le modèle de Black & Scholes ne fournit pas de solution analytique. Notre capacité à calculer ces mesures de risque avancées démontre la polyvalence et l'adaptabilité de notre modèle trinomial, ainsi que sa pertinence pour des analyses de risque plus complexes et détaillées.

Nous avons cherché à enrichir notre modèle de pricing d'options par des méthodes numériques avancées, notamment pour le calcul des Grecques en Python. Notre tentative d'estimer la 'Theta' a montré un écart notable avec le modèle de Black & Scholes, suggérant que notre méthode pourrait nécessiter une amélioration pour cette mesure de dépréciation temporelle.

En complément, nous avons entamé la création d'un script destiné à exécuter une batterie de calculs des sensibilités, dans le but de dresser une cartographie des Grecques à travers des visualisations graphiques[4.4]. Cependant, faute de temps, ce projet n'a pas pu être finalisé. De surcroît, la vitesse d'exécution du script s'est avérée insuffisante, ce qui limite son utilité pratique dans l'état actuel. La poursuite de ce travail et son optimisation demeurent des axes d'amélioration importants pour l'avenir, pouvant déboucher sur un outil d'analyse de risque rapide et efficace.

Cette démarche démontre notre volonté d'innover et notre engagement à fournir des outils d'analyse de risque exhaustifs et précis, capables de supporter les décisions financières dans un marché en constante évolution.

# Annexes

FIGURE 4.1 – Prunning

[illegible]

FIGURE 4.2 – Convergence du modèle Trinomial vers Black &amp; Scholes

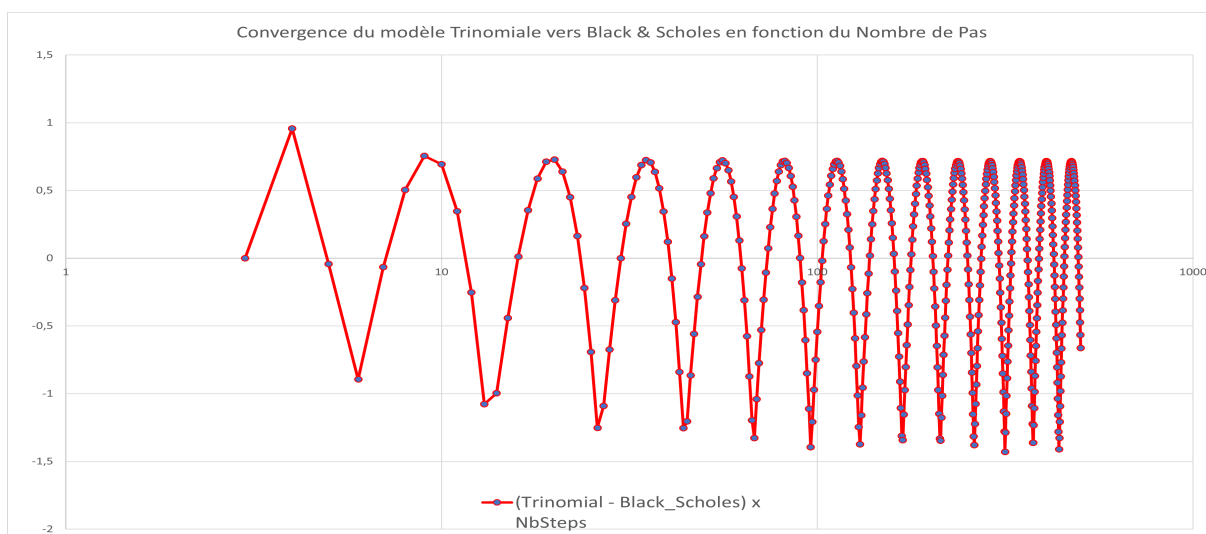


FIGURE 4.3 – Convergence du modèle Trinomial vers Black & Scholes en fonction du strike

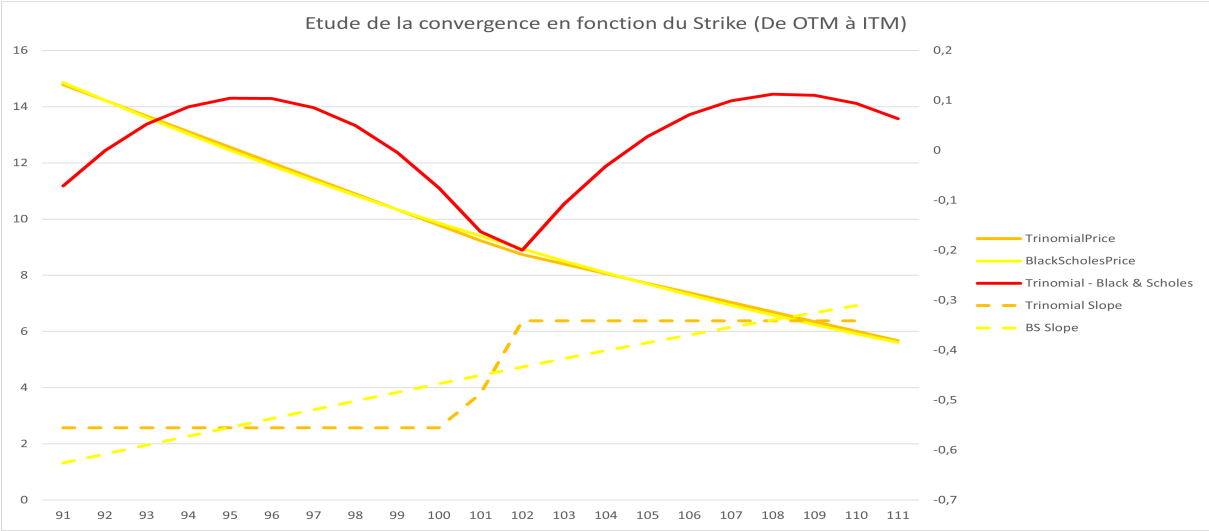


FIGURE 4.4 – Delta d'un Call européen

