



GRENOBLE INP - ENSIMAG
LABORATOIRE JEAN KUNTZMANN



EDP pour les options asiatiques

Introduction à la recherche en laboratoire
Kévin BONKOSKI

2ème année - Ingénierie pour la finance

Grenoble - mai 2015

Contexte de l'étude

Ce travail de recherche a été menée sous le tutorat de Madame Bidégaray-Fesquet Brigitte au sein du Laboratoire Jean Kuntzmann (LJK).

L'équipe EDP fait partie du Département MAD (Modèles et Algorithmes Déterministes) et comporte des chercheurs travaillant sur des sujets de modélisation, d'analyse et de calcul scientifique appliqué aux sciences du vivant et aux sciences des matériaux.

Au travers de leurs recherches, les membres de l'équipes utilisent des techniques permettant de résoudre des équations aux dérivées partielles (EDP).

Or, dans le domaine de la Finance, certaines de ces méthodes sont reprises et ainsi ré-utilisées.

L'objectif principal de mon étude était de travailler sur la résolution d'EDP appliquées aux options asiatiques. Cette méthode numérique moins utilisée pour «pricer» des produits dérivés que celle de Monte-Carlo peut dans certains cas être intéressante.

Cette méthode n'étant pas dans le cadre du programme de la filière Ingénierie Financière de deuxième année de l'ENSIMAG, cette Introduction à la Recherche en Laboratoire (IRL) m'a permis de découvrir le monde de la recherche tout en approfondissant ma formation d'un point de vue financière.

Ce module m'a permis de comparer le monde de la recherche publique avec la R&D que l'on peut retrouver en entreprises comme la recherche quantitative en Finance.

Table des matières

1	Introduction	3
1.1	Un bref rappel sur les options	3
1.2	L’option Asiatique	3
1.3	Les différentes méthodes de pricing	4
1.4	Le travail réalisé	5
2	Le modèle	6
2.1	Le modèle de Black–Scholes pour les options vanille	6
2.1.1	Les changements de variables vers l’équation de la chaleur	7
2.1.2	La résolution de l’équation de la chaleur	9
2.1.3	Les limites de la forme fermée	10
2.2	Options asiatiques	10
3	Implémentation numérique	12
3.1	La résolution de l’EDP pour une option vanille	12
3.1.1	La méthode explicite	13
3.1.2	La méthode implicite	16
3.1.3	La méthode de Crank–Nicolson	17
3.2	La résolution de l’EDP pour les options asiatiques	20
3.2.1	La mise en équation	20
3.2.2	La méthode de splitting	21
4	Cas tests	23
4.1	Tests des options vanille	23
4.1.1	Visualisation de différentes méthodes en fonction du temps	23
4.1.2	Influence de la volatilité	23
4.1.3	Influence du taux d’intérêt	26
4.1.4	Influence de la maturité	26
4.1.5	Stabilité des méthodes	26
4.1.6	Cas où la discrétisation n’est pas uniforme	33
4.2	Tests des options asiatiques	34
5	Conclusion	38
6	Annexes	39
6.1	Algorithme PYTHON qui permet de pricer une option vanille ou asiatique	39
6.2	Fichier Json permettant la lecture des paramètres	44

1 Introduction

1.1 Un bref rappel sur les options

Depuis le milieu des années 70, les marchés financiers ont subi de grandes transformations. En effet, les différents taux que l'on peut trouver dans la nature (taux d'intérêt, taux de change...) sont devenus très volatiles et instables. C'est dans le but de couvrir les risques dus à ce genre de situation, que les institutions financières ont commencé à créer des marchés organisés. En effet, de nombreux produits financiers ont vu le jour : c'est le début des produits dérivés.

Les options sont des produits financiers que l'on peut retrouver sur des marchés organisés et sur le marché OTC (marché de gré à gré). Il existe deux types d'option :

- celle d'achat (Call) qui permet à son détenteur d'avoir le droit d'acheter une certaine quantité d'actif sous-jacent à une valeur et une date fixée à l'avance,
- celle de vente (Put) qui permet à son détenteur d'avoir le droit de vendre une certaine quantité d'actif sous-jacent à une valeur et une date fixée à l'avance.

Le prix d'exercice ou strike sera noté K par la suite. La date maximale qui est fixée à l'avance est appelée échéance ou maturité. Dès le départ, on pouvait observer deux grandes familles d'options :

- les options américaines qui se distinguent par la possibilité d'un exercice à tout moment entre la signature du contrat et la maturité
- les options européennes qui, quant à elles, se voient dans l'obligation d'être exercées à la date de la maturité.

Par la suite, des options plus complexes ont commencé à émerger sur les marchés financiers telles que les options asiatiques, barrières, choosen, etc. Contrairement aux options vanille (classiques), les options asiatiques prennent d'autres paramètres en compte que le strike K , la maturité et bien entendu la volatilité σ et le taux d'intérêt r .

1.2 L'option Asiatique

Vers la fin des années 80, à Tokyo, de nouveaux types de contrats ont vu le jour, portant sur des barils de pétrole.

En temps normal, le choix d'un détenteur d'une option vanille par exemple

européenne, est directement lié au prix initial du sous-jacent. Ceci ne dépend en aucun cas de la tendance du prix du sous-jacent entre la date initiale et la date finale. De même pour l'option américaine, cela dépendra du moment où l'option est exercée. Or ces options coûtent relativement cher. C'est pourquoi, les options asiatiques ont été créées afin d'essayer de remédier à ce problème.

Une option asiatique [4] permet à son détenteur de prendre comme référence le cours moyen d'un sous-jacent et non son prix final. Bien entendu, le prix d'exercice sera encore fixé à l'avance. La moyenne de ce cours, bien que souvent prise comme arithmétique, est parfois géométrique. Cette moyenne peut être observée sur la durée entière de l'option mais aussi bien sur le dernier mois ou sur la dernière semaine par exemple. De plus, les observations peuvent être journalières, hebdomadaires, etc. C'est aux parties de décider de ces caractéristiques.

1.3 Les différentes méthodes de pricing

Rappelons que les produits financiers que sont les options ont un coût lors de l'achat : c'est la prime. Elle sera en générale notée C ou P suivant que l'option est une option d'achat ou de vente. Réussir à estimer cette prime est un domaine de recherche en mathématiques financières à part entière : c'est ce que l'on appelle couramment le pricing d'options.

Depuis 1973, les travaux de recherche de F. Black, M. Scholes [1] ainsi que de R.C. Merton ont permis de trouver une formule exacte permettant de pricer les options européennes sous certaines conditions. Cependant, compte tenu des caractéristiques des autres types d'options, il n'existe pas encore de formules exactes permettant de déterminer leurs primes respectives. C'est pourquoi on a l'habitude d'utiliser des méthodes numériques. L'efficacité de celles-ci se caractérise par un compromis entre le temps de calcul et l'erreur induite (précision de la méthode).

La plupart du temps, on se restreindra aux trois techniques d'évaluation suivantes.

- Le recours aux arbres binomiaux, induit par le modèle de Cox–Ross–Rubinstein (CRR).
- La méthode de Monte–Carlo qui est une méthode probabiliste et stochastique. Elle consiste à générer un ensemble de trajectoires plausibles et ainsi à calculer la valeur de la prime pour chaque trajectoire. On prend ensuite la moyenne de toutes ces estimations pour obtenir la valeur moyenne de la prime. C'est une méthode qui est assez coûteuse

mais ne se soucie pas des conditions de bords.

- La résolution de l'EDP via la technique des différences finies ou autres techniques de résolution d'EDP. Cette méthode se résume à discrétiser les différentes variables de l'équation et à la résoudre numériquement.

1.4 Le travail réalisé

Comme nous l'avons vu précédemment, les options asiatiques sont des options exotiques dépendant du prix moyen du sous-jacent dans un intervalle de temps donné. Elles donnent lieu à des primes inférieures aux options vanille. Dans leur variante sous forme d'équation aux dérivées partielles, les options asiatiques donnent lieu à des équations avec une variable temporelle et deux dérivées d'espace : le cours du sous-jacent noté S et la moyenne de ses cours A . Cette forme est semblable à d'autres modèles très courants comme le modèle d'Heston. Notons que S et A doivent être considérées comme des variables indépendantes.

Bien entendu, il n'existe pas de formule fermée pour les options asiatiques. Elles peuvent être simulées grâce à des méthodes de type Monte-Carlo, mais le plus efficace (en terme de temps de calcul) consiste à utiliser la version équation aux dérivées partielles de ce modèle et à utiliser typiquement une méthode de directions alternées. Ce projet m'a permis d'explorer les performances comparées des différentes méthodes (Monte-Carlo, différences finies, avec et sans raffinement), ainsi que le cas où seule des valeurs discrètes du sous-jacent sont intégrées.

Par la suite, je me suis limité à l'étude des options de vente (Put). L'étude des options d'achat (Call), quant à elle, découle assez simplement de celle précédente car les pay-offs de ces options à l'instant final sont complémentaires comme le montre la figure 1 page 6.

Dans un premier temps, j'ai donc étudié le modèle de l'EDP sur les options européennes. Afin de vérifier la cohérence de cette technique de résolution, j'ai comparé les résultats avec ceux obtenus grâce à la formule fermée de Black-Scholes. Par la suite, je me suis intéressé au pricing des options asiatiques en résolvant l'EDP par méthode de splitting. Bien qu'intéressante d'un point de vue complexité, celle-ci nécessite une bonne étude des conditions de bords. Or parfois, cette étude est assez délicate car nous n'arrivons pas forcément à trouver des conditions de bords permettant d'obtenir un résultat convenable.

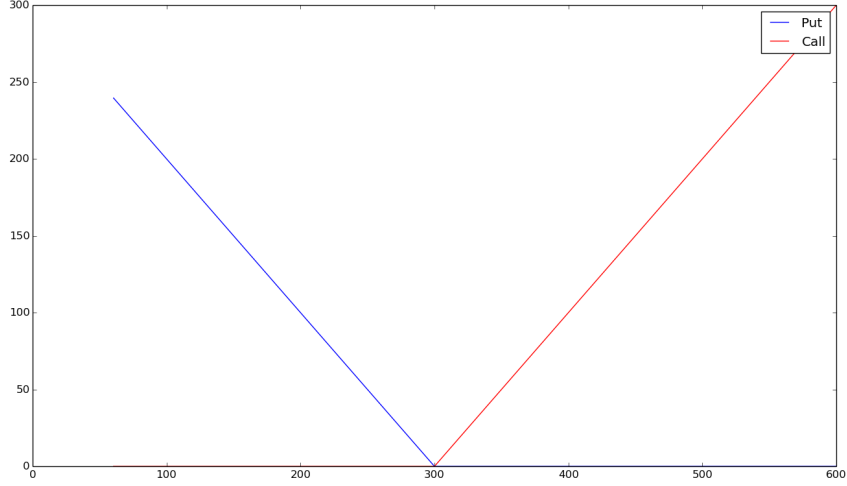


FIGURE 1 – Pay-off d’une option vanille d’achat $\max(S_t - K, 0)$ et vente $\max(K - S_t, 0)$

2 Le modèle

Dans un premier temps, nous allons étudier la formule fermée du modèle de Black-Scholes pour une option vanille européenne.

2.1 Le modèle de Black–Scholes pour les options vanille

Comme il est décrit dans l’article fondateur de 1973 [2], nous devons supposer dans un premier temps l’hypothèse de base qui est que le prix du sous-jacent suit le mouvement brownien géométrique :

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

où W_t est un processus de Wiener et σ est la volatilité.

Nous introduisons par ailleurs la notion de porte-monnaie qui représente la richesse de l’individu à l’instant t . Un porte-monnaie, qui sera constitué d’une part α d’actions d’un sous-jacent S et d’une part β d’obligations (considérées sans-risques d’où l’équivalence avec des zéro-coupons), vaudra ainsi

$$\Pi_t = \alpha S_t + \beta B_t,$$

où S_t est le prix de l'action du sous-jacent sur laquelle s'applique l'option et B_t est défini comme suit : $dB_t = rB_t dt$ où r est le taux sans-risque.

Après avoir utilisé l'hypothèse de non arbitrage, c'est-à-dire qu'il n'est pas possible de gagner de l'argent sans prendre de risque sur un marché financier, nous appliquons le lemme :

Lemme 2.1. *Soit un processus d'Itô X_t , un processus stochastique de la forme $X_t = X_0 + \int_0^t \mu_s ds + \int_0^t \sigma_s dB_s$ ou tel que $dX_t = \mu_t dt + \sigma_t dB_t$, si $f(X_t, t)$ est une fonction de classe $C^2(\mathbb{R} \times \mathbb{R}_+, \mathbb{R})$, alors :*

$$d(f(X_t, t)) = \frac{\partial f}{\partial t}(X_t, t)dt + \frac{\partial f}{\partial x}(X_t, t)dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(X_t, t)\sigma_t^2 dt.$$

Comme l'avaient démontré en 1973, F. Black et M. Scholes, nous retrouvons ainsi l'équation de Black-Scholes. Cette équation ainsi que l'égalité avec la fonction pay-off au temps final, donne donc le problème de Cauchy :

$$\begin{cases} \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0, \\ V(S, t) = \Phi(S). \end{cases} \quad (E_1)$$

2.1.1 Les changements de variables vers l'équation de la chaleur

Nous pouvons remarquer dans un premier temps qu'il y a dans l'équation de Black-Scholes (E_1) un problème de dégénérescence de l'opérateur elliptique, la partie $\frac{1}{2}\sigma^2 S^2 \partial^2 V / \partial S^2$. En effet, ce terme pose un problème lorsque S tend vers 0. C'est pourquoi, afin d'ôter la singularité en S , il est d'usage de poser le changement de variable :

$$\begin{cases} S = Ke^x, \\ z(x, t) = V(S, t). \end{cases}$$

Alors

$$\begin{aligned} \frac{\partial z}{\partial x} &= \frac{\partial V}{\partial S} \frac{dS}{dx} = Ke^x \frac{\partial V}{\partial S} = S \frac{\partial V}{\partial S}, \\ \frac{\partial^2 z}{\partial x^2} &= Ke^x \frac{\partial V}{\partial S} + Ke^x \frac{\partial^2 V}{\partial S^2} \frac{dS}{dx} = Ke^x \frac{\partial V}{\partial S} = S \frac{\partial V}{\partial S} + S^2 \frac{\partial^2 V}{\partial S^2}. \end{aligned}$$

On obtient ainsi que

$$\frac{\partial z}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 z}{\partial x^2} + (r - \frac{1}{2}\sigma^2) \frac{\partial z}{\partial x} - rz = 0. \quad (E_2)$$

Nous avons par la suite choisi de retourner la variable de temps également. En effet, choisir la variable de temps $\tau = (T - t)\sigma^2$ nous permet d'avoir une "notion" de temps qui avance et non un temps qui se décrémente. En différenciant, nous obtenons que

$$\frac{\partial z}{\partial \tau} = \frac{\partial V}{\partial t} \frac{dt}{d\tau} = -\frac{1}{\sigma^2} \frac{\partial V}{\partial t}.$$

Finalement, en posant $q = 2r/\sigma^2$, nous avons

$$-\frac{\partial z}{\partial \tau} + \frac{1}{2} \frac{\partial^2 z}{\partial x^2} + \frac{1}{2}(q - 1) \frac{\partial z}{\partial x} - \frac{1}{2} q z = 0.$$

Pour une option européenne classique, nous savons que nous pouvons utiliser un dernier changement de variables qui permet de trouver la solution du problème fermé de Black-Scholes : $z(x, t) = e^{\gamma x + \omega \tau} y(x, \tau)$. Nous obtenons donc que

$$\begin{aligned} \frac{\partial z}{\partial x} &= \gamma e^{\gamma x + \omega \tau} y(x, \tau) + e^{\gamma x + \omega \tau} \frac{\partial y}{\partial x}(x, \tau), \\ \frac{\partial^2 z}{\partial x^2} &= \gamma^2 e^{\gamma x + \omega \tau} y(x, \tau) + 2\gamma e^{\gamma x + \omega \tau} \frac{\partial y}{\partial x}(x, \tau) + e^{\gamma x + \omega \tau} \frac{\partial^2 y}{\partial x^2}(x, \tau), \\ \frac{\partial z}{\partial \tau} &= \omega e^{\gamma x + \omega \tau} y(x, \tau) + e^{\gamma x + \omega \tau} \frac{\partial y}{\partial \tau}(x, \tau). \end{aligned}$$

Ainsi, nous obtenons que

$$-\frac{\partial y}{\partial \tau} + \frac{1}{2} \frac{\partial^2 y}{\partial x^2} + \left(\gamma + \frac{1}{2}(q - 1) \right) \frac{\partial y}{\partial x} + \left(-\omega + \frac{1}{2}\gamma^2 + \frac{1}{2}\gamma(q - 1) - \frac{1}{2}q \right) y = 0$$

En remarquant que pour des γ et ω fixés comme suit

$$\begin{cases} \gamma = \frac{1}{2}(1 - q), \\ \omega = -\frac{1}{8}(1 + q)^2, \end{cases}$$

les derniers termes de l'équation s'annulent, l'équation précédente se réduit à celle de la chaleur :

$$\frac{\partial y}{\partial \tau} = \frac{1}{2} \frac{\partial^2 y}{\partial x^2}.$$

Nous avons inversé le temps. De ce fait, $V(S, T) = e^{\gamma x} y(x, 0)$. Rappelons que la fonction de pay-off s'écrit sous la forme $\Psi(S) = (K - S)_+$. L'équation prend donc pour conditions initiales

$$\begin{aligned} y(x, 0) &= e^{-\gamma x} \Psi(K e^x) = e^{-\gamma x} (K - K e^x)_+ \\ &= K e^{-\gamma x} (1 - e^x)_+. \end{aligned}$$

2.1.2 La résolution de l'équation de la chaleur

Repartons de l'équation de la chaleur trouvée précédemment :

$$\begin{cases} \frac{\partial y}{\partial \tau} = \frac{1}{2} \frac{\partial^2 y}{\partial x^2}, \\ y(x, 0) = K e^{-\gamma x} (1 - e^x)_+. \end{cases}$$

La solution du problème de Cauchy est connue et admet une formule fermée permettant de le résoudre :

$$\begin{aligned} y(x, \tau) &= \frac{1}{\sqrt{2\pi\tau}} \int_{-\infty}^{+\infty} y(\xi, 0) e^{-\frac{(x-\xi)^2}{2\tau}} d\xi \\ &= \frac{K}{\sqrt{2\pi\tau}} \int_0^{+\infty} e^{-\gamma\xi - \frac{(x-\xi)^2}{2\tau}} d\xi - \frac{K}{\sqrt{2\pi\tau}} \int_0^{+\infty} e^{\xi - \gamma\xi - \frac{(x-\xi)^2}{2\tau}} d\xi \\ &= \frac{K}{\sqrt{2\pi\tau}} \int_0^{+\infty} e^{-\gamma\xi - \frac{(x-\xi)^2}{2\tau}} d\xi - \frac{K}{\sqrt{2\pi\tau}} \int_0^{+\infty} e^{-(\gamma-1)\xi - \frac{(x-\xi)^2}{2\tau}} d\xi. \end{aligned}$$

Ceci s'écrit sous la forme

$$y(x, \tau) = \frac{K}{\sqrt{2\pi\tau}} \Theta(x, \tau, \gamma) - \frac{K}{\sqrt{2\pi\tau}} \Theta(x, \tau, \gamma - 1),$$

avec

$$\begin{aligned} \Theta(x, \tau, \gamma) &= \int_0^{+\infty} e^{-\gamma\xi - \frac{(x-\xi)^2}{2\tau}} d\xi \\ &= \sqrt{2\pi\tau} e^{-\gamma(x - \frac{1}{2}\gamma\tau)} \mathcal{F}_{\mathcal{N}}\left(\frac{x - \gamma\tau}{\sqrt{\tau}}\right), \end{aligned}$$

où $\mathcal{F}_{\mathcal{N}}$ est la fonction de répartition d'une loi normale centrée réduite à savoir $\mathcal{F}_{\mathcal{N}}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{\xi^2}{2}} d\xi$. Comme nous connaissons désormais

$$y(x, \tau) = K e^{-\gamma(x - \frac{1}{2}\gamma\tau)} \mathcal{F}_{\mathcal{N}}\left(\frac{x - \gamma\tau}{\sqrt{\tau}}\right) - K e^{-(\gamma-1)(x - \frac{1}{2}(\gamma-1)\tau)} \mathcal{F}_{\mathcal{N}}\left(\frac{x - (\gamma-1)\tau}{\sqrt{\tau}}\right),$$

il suffit d'utiliser les changements de variables que nous avons vu précédemment pour trouver la valeur de $V(S, t)$:

$$\begin{aligned} z(x, \tau) &= e^{\gamma x + \omega \tau}(x, \tau) \\ &= Ke^{(\gamma - \frac{1}{2})\tau} \mathcal{F}_{\mathcal{N}}\left(\frac{x - \gamma\tau}{\sqrt{\tau}}\right) - Ke^x \mathcal{F}_{\mathcal{N}}\left(\frac{x - (\gamma - 1)\tau}{\sqrt{\tau}}\right). \end{aligned}$$

Nous obtenons finalement la formule fermée :

$$V(S, t) = Ke^{-rt} \mathcal{F}_{\mathcal{N}}(-d_2) - S \mathcal{F}_{\mathcal{N}}(-d_1),$$

où

$$\begin{cases} d_2 = \frac{\ln(\frac{S}{K}) + (r - \frac{\sigma^2}{2})(T - t)}{\sigma\sqrt{T - t}}, \\ d_1 = d_2 + \sigma\sqrt{T - t} = \frac{\ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T - t)}{\sigma\sqrt{T - t}}. \end{cases}$$

Nous avons donc implémenté cette formule en PYTHON. Cette formule permet lorsque l'on fait varier le cours du sous-jacent initial de tracer une courbe représentant le prix de la prime d'une option vanille d'achat comme nous pouvons le voir dans la figure 2 page 11.

2.1.3 Les limites de la forme fermée

Cette formule ne fonctionne plus lorsque les hypothèses de départ divergent. Notamment, elle ne marche plus lorsque

- l'option prend en compte au moins deux actions ;
- r et σ varient en fonction du temps.

2.2 Options asiatiques

Nos hypothèses de départ dans le cas d'une option asiatique sont les mêmes que pour l'option vanille classique. Il faut cependant noter que le cours du sous-jacent peut être soit considéré comme continu, soit discontinu. Ainsi, les différentes moyennes en mathématiques peuvent être définies :

- la moyenne discrète arithmétique $\hat{S} = \frac{1}{N} \sum_{n=1}^N S_{t^n}$,
- la moyenne discrète géométrique $\hat{S} = (\prod_{n=1}^N S_{t^n})^{\frac{1}{N}}$,
- la moyenne continue arithmétique $\hat{S} = \frac{1}{T} \int_0^T S_t dt$,
- la moyenne continue géométrique $\hat{S} = \exp(\frac{1}{T} \int_0^T \ln S_t dt)$.

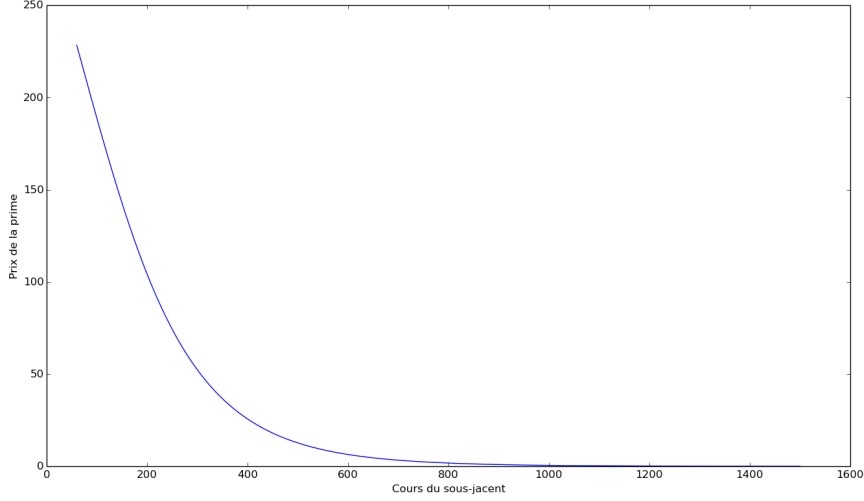


FIGURE 2 – Prix de la prime d’une option vanille obtenu grâce à la formule fermée de Black–Scholes — $K = 300$, $r = 4\%$ et $\sigma = 0.5$

Afin d’écrire un modèle EDP, j’ai choisi de travailler dans le cas continu en utilisant la moyenne arithmétique. Ainsi, je note

$$A = \int_0^t f(S_s, s) ds$$

où s est le temps et t l’instant donné.

Remarquons également que

- $dA = f(S, t)dt$,
- en vertu du lemme d’Itô, la fonction $V(S, t, A)$ est solution de

$$\frac{\partial V}{\partial t}(x, t, A) + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}(x, t, A) + rS \frac{\partial V}{\partial S}(x, t, A) - rV + f(S, t) \frac{\partial V}{\partial A}(x, t, A) = 0.$$

Or en sachant que $f(S, t) = S_t$ pour la moyenne arithmétique et en posant le même changement de variable que dans le cas de l’option vanille ($S = Ke^x$), nous obtenons

$$\frac{\partial \Phi}{\partial t}(x, t, A) + \frac{\sigma^2}{2} \frac{\partial^2 \Phi}{\partial x^2}(x, t, A) + \left(r - \frac{\sigma^2}{2}\right) \frac{\partial \Phi}{\partial x}(x, t, A) - r\Phi(x, t, A) + \frac{e^x}{K} \frac{\partial \Phi}{\partial A}(x, t, A) = 0. \quad (E_3)$$

Résoudre l’équation (E_3) permettrait de trouver la prime d’une option asiatique.

3 Implémentation numérique

Comme nous l'avons vu dans l'introduction, mon travail de recherche se résume à étudier la résolution des EDP (E_1) et (E_3) via la technique des différences finies. Nous devons dans un premier temps discrétiser les différentes variables de l'équation et par la suite, nous devons la résoudre numériquement.

3.1 La résolution de l'EDP pour une option vanille

Pour discrétiser l'équation aux dérivées partielles de Black-Scholes pour une option vanille, nous repartons de l'équation (E_1) vue dans la section précédente :

$$\frac{\partial \Phi}{\partial t}(x, t) + \alpha \frac{\partial^2 \Phi}{\partial x^2}(x, t) + \beta \frac{\partial \Phi}{\partial x}(x, t) + \gamma \Phi(x, t) = 0,$$

avec

$$\begin{cases} \alpha = \frac{\sigma^2}{2}, \\ \beta = r - \frac{\sigma^2}{2}, \\ \gamma = -r. \end{cases}$$

Dans un premier temps, j'ai considéré une distribution uniforme du temps et de l'espace. Dans la suite du rapport nous noterons Φ_i^n l'approximation de $\Phi(S_i, t_n)$.

En appliquant la formule des différences finies classiques du type

$$\frac{f(x+h) - f(x-h)}{2h} \simeq f'(x),$$

j'ai discrétisé l'intervalle de 0 à T en $N + 1$ intervalles uniformes. Chaque temps de discrétisation sera donc de la forme

$$\forall n \in \llbracket 0, N \rrbracket, t^n = t^0 + n\delta t = t^0 + n\frac{T}{N},$$

en respectant bien entendu $t^0 = 0$ et $t^N = T$. De la même manière, j'ai discrétisé le cours du sous-jacent qui est compris entre x_{min} et x_{max} en $M + 1$ pas d'"espace"

$$\forall i \in \llbracket 0, M \rrbracket, x_i = x_0 + n\frac{x_M - x_0}{M},$$

avec $x_0 = x_{min}$ et $x_M = x_{max}$.

3.1.1 La méthode explicite

Pour la résolution de l'EDP par une méthode explicite, j'ai utilisé les formules des différences finies suivantes :

$$\begin{aligned}\frac{\partial \Phi}{\partial x}(x_i, t^n) &\simeq \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2\delta x}, \\ \frac{\partial^2 \Phi}{\partial x^2}(x_i, t^n) &\simeq \frac{\Phi_{i+1}^n - 2\Phi_i^n + \Phi_{i-1}^n}{\delta x^2}, \\ \frac{\partial \Phi}{\partial t}(x_i, t^n) &\simeq \frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t}.\end{aligned}$$

En remplaçant ces termes dans l'équation (E_1) , nous obtenons

$$\begin{aligned}\frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t} &= \alpha \frac{\Phi_{i+1}^n - 2\Phi_i^n + \Phi_{i-1}^n}{\delta x^2} + \beta \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2\delta x} + \gamma \Phi_i^n, \\ \Phi_i^{n-1} &= \xi_1 \Phi_{i+1}^n + \xi_2 \Phi_i^n + \xi_3 \Phi_{i-1}^n,\end{aligned}$$

avec

$$\begin{cases} \xi_1 = \delta t \left(\frac{\alpha}{\delta x^2} + \frac{\beta}{2\delta x} \right), \\ \xi_2 = \delta t \left(\frac{1}{\delta t} - \frac{2\alpha}{\delta x^2} + \gamma \right), \\ \xi_3 = \delta t \left(\frac{\alpha}{\delta x^2} - \frac{\beta}{2\delta x} \right). \end{cases}$$

A chaque itération n , je cherche à calculer le vecteur $(\Phi_1^{n-1}, \Phi_2^{n-1}, \dots, \Phi_{M-1}^{n-1})$, en respectant les conditions aux bords suivantes :

- une condition de Neumann pour Φ_M ,
- une condition de Dirichlet pour Φ_0 .

La condition de Neumann se traduit par

$$\frac{\partial \Phi}{\partial x}(x = x_M) = 0 \Leftrightarrow \forall n, \Phi_M^n = \Phi_{M-1}^n.$$

La condition de Dirichlet se traduit quant à elle à l'instant t par

$$\Phi_0^n = (K - S_0)e^{-r(T-t^n)},$$

où T est la maturité et t^n l'instant donné lors de la $(N - n)$ -ème itération. Nous obtenons ainsi l'équation vectorielle :

$$\begin{pmatrix} \Phi_1^{n-1} \\ \Phi_2^{n-1} \\ \vdots \\ \vdots \\ \Phi_{M-1}^{n-1} \end{pmatrix} = \begin{pmatrix} \xi_2 & \xi_1 & 0 & \cdots & 0 \\ \xi_3 & \xi_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \xi_2 & \xi_1 \\ 0 & \cdots & 0 & \xi_3 & \xi_1 + \xi_2 \end{pmatrix} \begin{pmatrix} \Phi_1^n \\ \Phi_2^n \\ \vdots \\ \vdots \\ \Phi_{M-1}^n \end{pmatrix} + \begin{pmatrix} \xi_3(K - S_0)e^{-r(t^N - t^n)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Ce calcul revient donc à calculer le vecteur Φ^{n-1} tel que

$$\Phi^{n-1} = A\Phi^n + C^n.$$

Comme la matrice A est une matrice tridiagonale, nous avons choisi de créer en PYTHON notre propre structure de données pour stocker les valeurs de coefficients diagonaux de la matrice. Cela permet en particulier d'éviter de stocker tous les zéros qui ne servent à rien. Cette structure de donnée se rapproche des matrices sparses en MATLAB. Nous avons donc codé la multiplication matrice-vecteur (Algorithm 1).

Algorithm 1 mult(A,X) : Algorithme de multiplication d'une matrice tri-diagonale A avec un vecteur X - D'un point de vue structure de données, A est simplement 3 vecteurs A_0, A_1 et A_2

```

M = size(X)
Vect[0] = X[0] * A1[0] + X[1] * A[0,0]
for i ∈ [1; M - 2] do
    Vect[i] = X[i] * A1[i] + X[i - 1] * A2[i] + X[i + 1] * A0[i]
end for
Vect[M - 1] = X[M - 1] * A1[M - 1] + X[M - 2] * A2[M - 1]
return Vect

```

Pour résoudre cette l'EDP (E_1), nous devons itérer cette étape jusqu'au temps initial $t^0 = 0$ (Algorithm 2).

Nous avons tracé le prix de la prime d'une option vanille grâce à cette méthode (figure 3, page 15).

Nous remarquons que cette courbe est exploitable. Cependant, cette méthode est effectivement instable. Par la suite nous étudierons sa stabilité (**Partie 4.1.5**). Pour éviter ce genre de problèmes, nous devons donc nous tourner vers une méthode implicite, où nous aurons à inverser une matrice plutôt que d'avoir un simple produit matrice-vecteur.

Algorithm 2 Black–Scholes-explicite($S_{min}, S_{max}, M, K, T, N, r, \sigma$) : Algorithme simplifié de résolution de Black–Scholes avec la méthode explicite

```

for  $s \in \text{tab}S$  do
     $Vect = [\max(K - s, 0)]$ 
end for
 $dt = T/N$ 
construction de la matrice  $A$ 
for  $n \in \llbracket 1; N \rrbracket$  do
     $t^n = T - ndt$ 
     $C^n[0] = \xi_3(K - S_{min})e^{-r(T-t^n)}$ 
     $Vect = \text{mult}(A, Vect)$ 
     $Vect_+ = C^n$ 
end for
return tracé de  $Vect$  en fonction de  $\text{tab}S$  (discrétisation des cours du sous-jacent)

```

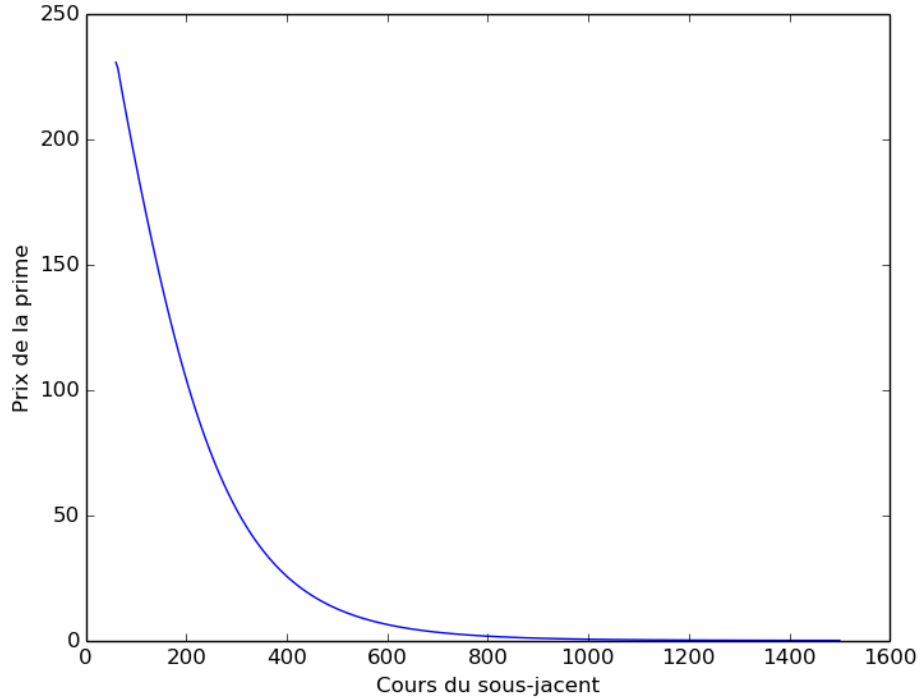


FIGURE 3 – Résolution de l’EDP avec la méthode explicite pour une option vanille - $K = 300, r = 4\%$ et $\sigma = 0.5$ — $\alpha\delta t/\delta x^2 = 0.962$

3.1.2 La méthode implicite

Pour la résolution de l'EDP par une méthode implicite, j'ai utilisé les formules de différences finies suivantes :

$$\begin{aligned}\frac{\partial \Phi}{\partial x}(x_i, t^{n-1}) &\simeq \frac{\Phi_{i+1}^{n-1} - \Phi_{i-1}^{n-1}}{2\delta x}, \\ \frac{\partial^2 \Phi}{\partial x^2}(x_i, t^{n-1}) &\simeq \frac{\Phi_{i+1}^{n-1} - 2\Phi_i^{n-1} + \Phi_{i-1}^{n-1}}{\delta x^2}, \\ \frac{\partial \Phi}{\partial t}(x_i, t^{n-1}) &\simeq \frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t}.\end{aligned}$$

En remplaçant ces termes dans l'équation (E_1) , nous obtenons

$$\begin{aligned}\frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t} &= \alpha \frac{\Phi_{i+1}^{n-1} - 2\Phi_i^{n-1} + \Phi_{i-1}^{n-1}}{\delta x^2} + \beta \frac{\Phi_{i+1}^{n-1} - \Phi_{i-1}^{n-1}}{2\delta x} + \gamma \Phi_i^{n-1}, \\ \Phi_i^n &= \zeta_1 \Phi_{i+1}^{n-1} + \zeta_2 \Phi_i^{n-1} + \zeta_3 \Phi_{i-1}^{n-1},\end{aligned}$$

avec

$$\begin{cases} \zeta_1 = -\delta t \left(\frac{\alpha}{\delta x^2} + \frac{\beta}{2\delta x} \right), \\ \zeta_2 = \delta t \left(\frac{1}{\delta t} + \frac{2\alpha}{\delta x^2} - \gamma \right), \\ \zeta_3 = \delta t \left(\frac{\beta}{2\delta x} - \frac{\alpha}{\delta x^2} \right). \end{cases}$$

J'ai également choisi de respecter les conditions aux bords suivantes :

- une condition de Neumann pour Φ_M ,
- une condition de Dirichlet pour Φ_0 .

La condition de Neumann se traduit par

$$\frac{\partial \Phi}{\partial x}(x = x_M) = 0 \Leftrightarrow \forall n, \Phi_M^{n-1} = \Phi_{M-1}^{n-1}.$$

La condition de Dirichlet se traduit à l'instant t par

$$\Phi_0^{n-1} = (K - S_0)e^{-r(T-t^{n-1})},$$

où T est la période et t^n l'instant donné lors de la $(N - (n - 1))$ -ième itération.

Nous obtenons ainsi le système linéaire suivant :

$$\begin{pmatrix} \Phi_1^n \\ \Phi_2^n \\ \vdots \\ \vdots \\ \Phi_{M-1}^n \end{pmatrix} = \begin{pmatrix} \zeta_2 & \zeta_1 & 0 & \cdots & 0 \\ \zeta_3 & \zeta_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \zeta_2 & \zeta_1 \\ 0 & \cdots & 0 & \zeta_3 & \zeta_1 + \zeta_2 \end{pmatrix} \begin{pmatrix} \Phi_1^{n-1} \\ \Phi_2^{n-1} \\ \vdots \\ \vdots \\ \Phi_{M-1}^{n-1} \end{pmatrix} + \begin{pmatrix} \zeta_3(K - S_0)e^{-r(T-t^{n-1})} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Ce calcul revient donc à chercher le vecteur Φ^{n-1} tel que

$$\Phi^n = B\Phi^{n-1} + C^n \Leftrightarrow \Phi^n - C^n = B\Phi^{n-1}.$$

Résoudre cette équation matricielle est plutôt facile car cette matrice est tridiagonale. Nous avons donc codé un algorithme qui est basé sur un pivot de Gauss (Algorithm 3).

Algorithm 3 *inverse(A,Y)* : Algorithme de résolution d'une équation du type $AX = Y$ avec A une matrice tridiagonale et X, Y deux vecteurs — D'un point de vue structure de données, A est simplement 3 vecteurs A_0, A_1 et A_2

```

M = size(Y)
X = zeros(M)
A0[0] = A0[0]/A1[0]
Y[0] = Y[0]/A1[0]
for  $i \in \llbracket 1, M-2 \rrbracket$  do
    temp = A1[i] - A2[i]A0[i-1]
    A0[i] =  $\frac{A0[i]}{temp}$ 
    Y[i] =  $\frac{Y[i] - A2[i]*Y[i-1]}{temp}$ 
end for
Y[M-1] =  $\frac{Y[M-1] - A2[M-1]Y[M-2]}{A1[M-1] - A2[M-1]A0[M-2]}$ 
X[M-1] = Y[M-1]
for  $i \in \llbracket M-2, 0 \rrbracket$  do
    X[i] = Y[i] - A0[i]X[i+1]
end for
return X

```

Pour résoudre cette EDP, nous devons itérer cette étape jusqu'au temps initial $t^0 = 0$ (Algorithm 4).

Nous avons tracé le prix de la prime d'une option vanille grâce à cette méthode (figure 4, page 18).

3.1.3 La méthode de Crank–Nicolson

Le schéma de Crank–Nicolson est une méthode qui permet d'obtenir un résultat avec un ordre supérieur (ici un ordre 2 en espace). Cette technique est en fait une sorte de combinaison linéaire entre la méthode implicite et la méthode explicite et plus précisément une moyenne entre les deux méthodes précédentes.

Algorithm 4 Black–Scholes-implicite($S_{min}, S_{max}, M, K, T, N, r, \sigma$) : Algorithme simplifié de résolution de Black–Scholes avec la méthode implicite

```

for  $s \in \text{tab}S$  do
     $Vect = \max(K - s, 0)$ 
end for
 $dt = T/N$ 
construction de la matrice  $B$ 
for  $n \in \llbracket 1; N \rrbracket$  do
     $t^n = T - ndt$ 
     $C^n[0] = \zeta_3(K - S_0)e^{-r(T-t^n)}$ 
     $Y = Vect - C^n$ 
     $Vect = \text{inverse}(B, Y)$ 
end for

```

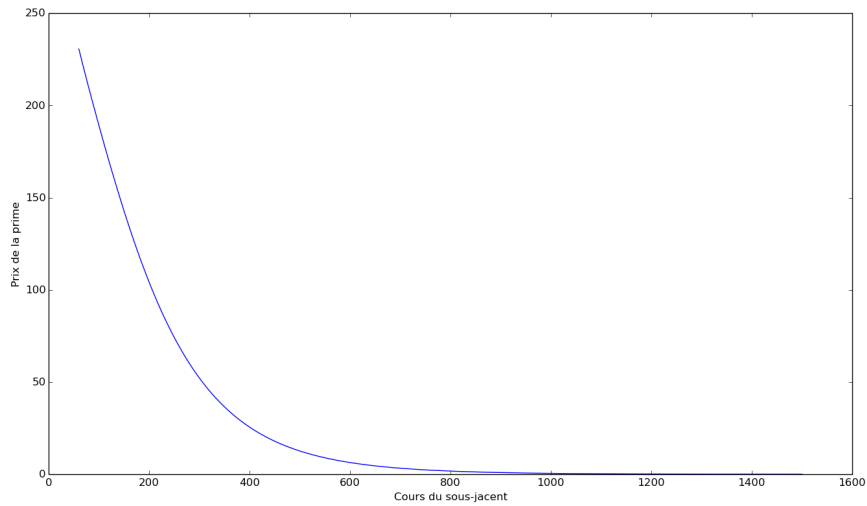


FIGURE 4 – Résolution de l'EDP avec la méthode implicite pour une option vanille - $K = 300, T = 1, r = 4\%$ et $\sigma = 0.5$

Pour la résolution de l'EDP par cette méthode, j'ai utilisé les formules des différences finies suivantes :

$$\begin{aligned}\frac{\partial \Phi}{\partial x}(x_i, t^{n+\frac{1}{2}}) &\simeq \theta \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2\delta x} + (1-\theta) \frac{\Phi_{i+1}^{n-1} - \Phi_{i-1}^{n-1}}{2\delta x}, \\ \frac{\partial^2 \Phi}{\partial x^2}(x_i, t^{n+\frac{1}{2}}) &\simeq \theta \frac{\Phi_{i+1}^n - 2\Phi_i^n + \Phi_{i-1}^n}{\delta x^2} + (1-\theta) \frac{\Phi_{i+1}^{n-1} - 2\Phi_i^{n-1} + \Phi_{i-1}^{n-1}}{\delta x^2}, \\ \frac{\partial \Phi}{\partial t}(x_i, t^{n+\frac{1}{2}}) &\simeq \frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t}.\end{aligned}$$

En prenant dans ces formules un θ quelconque, ça serait un θ -schéma. La méthode de Crank–Nicolson impose que $\theta = 1/2$. En remplaçant ces termes dans l'équation (E_1), nous obtenons

$$\begin{aligned}\frac{\Phi_i^n - \Phi_i^{n-1}}{\delta t} &= \alpha \left(\theta \frac{\Phi_{i+1}^n - 2\Phi_i^n + \Phi_{i-1}^n}{\delta x^2} + (1-\theta) \frac{\Phi_{i+1}^{n-1} - 2\Phi_i^{n-1} + \Phi_{i-1}^{n-1}}{\delta x^2} \right) \\ &+ \beta \left(\theta \frac{\Phi_{i+1}^n - \Phi_{i-1}^n}{2\delta x} + (1-\theta) \frac{\Phi_{i+1}^{n-1} - \Phi_{i-1}^{n-1}}{2\delta x} \right) + \gamma (\theta \Phi_i^n + (1-\theta) \Phi_i^{n-1}),\end{aligned}$$

$$\zeta_1 \Phi_{i+1}^{n-1} + \zeta_2 \Phi_i^{n-1} + \zeta_3 \Phi_{i-1}^{n-1} = \xi_1 \Phi_{i+1}^n + \xi_2 \Phi_i^n + \xi_3 \Phi_{i-1}^n,$$

avec

$$\left\{ \begin{array}{l} \xi_1 = \frac{\alpha}{2} \frac{\delta t}{\delta x^2} - \frac{\beta}{4} \frac{\delta t}{\delta x}, \\ \xi_2 = 1 - \alpha \frac{\delta t}{\delta x^2} + \frac{\gamma}{2} \delta t, \\ \xi_3 = \frac{\alpha}{2} \frac{\delta t}{\delta x^2} + \frac{\beta}{4} \frac{\delta t}{\delta x}, \\ \zeta_1 = -\frac{\alpha}{2} \frac{\delta t}{\delta x^2} + \frac{\beta}{4} \frac{\delta t}{\delta x}, \\ \zeta_2 = 1 + \alpha \frac{\delta t}{\delta x^2} - \frac{\gamma}{2} \delta t, \\ \zeta_3 = -\frac{\alpha}{2} \frac{\delta t}{\delta x^2} - \frac{\beta}{4} \frac{\delta t}{\delta x}. \end{array} \right.$$

J'ai également choisi de respecter les conditions aux bords suivantes :

- une condition de Neumann pour Φ_M ,
- une condition de Dirichlet pour Φ_0 .

La condition de Neumann se traduit par

$$\frac{\partial \Phi}{\partial x}(x = x_M) = 0 \Leftrightarrow \forall n, \left\{ \begin{array}{l} \Phi_{M-1}^{n-1} = \Phi_M^{n-1}, \\ \Phi_{M-1}^n = \Phi_M^n, \end{array} \right.$$

La condition de Dirichlet se traduit quant à elle aux instants t^{n-1} et t^n par

$$\left\{ \begin{array}{l} \Phi_0^{n-1} = (K - S_0)e^{-r(T-t^{n-1})}, \\ \Phi_0^n = (K - S_0)e^{-r(T-t^n)} \end{array} \right.$$

Nous obtenons ainsi l'équation vectorielle :

$$\begin{aligned}
& \begin{pmatrix} \zeta_2 & \zeta_1 & 0 & \cdots & 0 \\ \zeta_3 & \zeta_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \zeta_2 & \zeta_1 \\ 0 & \cdots & 0 & \zeta_3 & \zeta_1 + \zeta_2 \end{pmatrix} \begin{pmatrix} \Phi_1^{n-1} \\ \Phi_2^{n-1} \\ \vdots \\ \vdots \\ \Phi_{M-1}^{n-1} \end{pmatrix} \\
= & \begin{pmatrix} \xi_2 & \xi_1 & 0 & \cdots & 0 \\ \xi_3 & \xi_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \xi_2 & \xi_1 \\ 0 & \cdots & 0 & \xi_3 & \xi_1 + \xi_2 \end{pmatrix} \begin{pmatrix} \Phi_1^n \\ \Phi_2^n \\ \vdots \\ \vdots \\ \Phi_{M-1}^n \end{pmatrix} + \begin{pmatrix} (\xi_3 - \zeta_3 e^{r\delta t})[K e^{-r(t^N - t^n)}] \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\end{aligned}$$

Cela se ramène donc à trouver le vecteur Φ^{n-1} tel que

$$B\Phi^{n-1} = A\Phi^n + C^n$$

Ce calcul est uniquement composé d'opérations explicites et d'une inversion de matrice. Tous les algorithmes correspondants ont déjà été développés pour les méthodes implicites et explicites. Nous avons donc grâce à cette résolution, tracé la prime d'une option vanille en résolvant l'EDP via la méthode de Crank–Nicolson. Nous l'avons comparé à la formule fermée de Black–Scholes (figure 5, page 21).

3.2 La résolution de l'EDP pour les options asiatiques

3.2.1 La mise en équation

Pour discrétiser l'équation aux dérivées partielles de Black-Scholes pour une option asiatique, nous repartons de l'équation (E_3) vue dans la première section :

$$\frac{\partial \Phi}{\partial t}(x, t, A) + \alpha \frac{\partial^2 \Phi}{\partial x^2}(x, t, A) + \beta \frac{\partial \Phi}{\partial x}(x, t, A) + \gamma \Phi(x, t, A) + \frac{e^x}{K} \frac{\partial \Phi}{\partial A}(x, t, A) = 0$$

avec

$$\begin{cases} \alpha = \frac{\sigma^2}{2}, \\ \beta = r - \frac{\sigma^2}{2}, \\ \gamma = -r. \end{cases}$$

Résoudre l'équation (E_3) permettrait de trouver la prime d'une option asiatique. Cette EDP pourrait se résoudre par différentes méthodes. J'ai pour ma part, souhaité étudier la méthode de splitting.

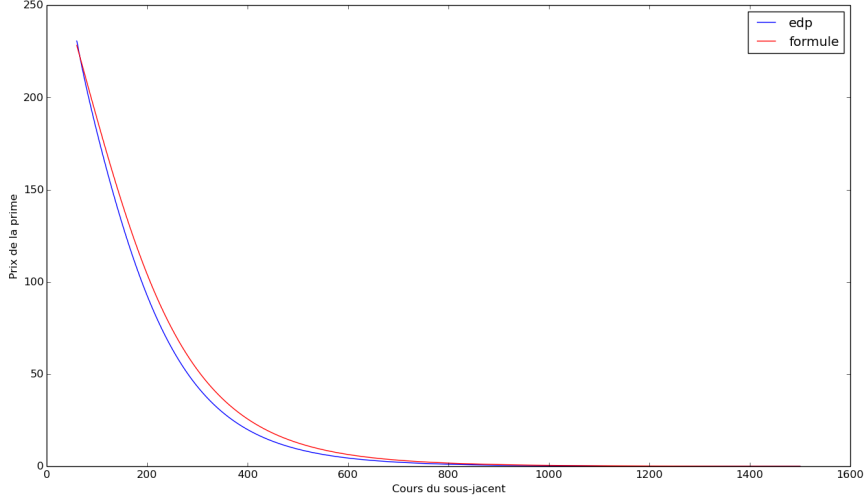


FIGURE 5 – Résolution de l’EDP avec la méthode de Crank–Nicolson et de la formule fermée pour une option vanille — $K = 300, T = 1, r = 4\%$ et $\sigma = 0.5$

3.2.2 La méthode de splitting

Le principe sur un exemple basique

$$\begin{cases} \frac{dx}{dt}(t) = (a + b)x(t), \\ x(0) = x_0. \end{cases}$$

La méthode de splitting a pour but de résoudre des EDP en résolvant l’EDP direction par direction. Dans ce cas, il faut dans un premier temps, trouver y solution de

$$\begin{cases} \frac{dy}{dt}(t) = by(t), \\ y(0) = x_0. \end{cases}$$

En prenant $y(t) = x_0 e^{bt}$ comme condition initiale, il suffit de résoudre

$$\begin{cases} \frac{dx}{dt}(t) = ax(t), \\ x(0) = y(t). \end{cases}$$

Et finalement

$$\hat{x}(t) = x_0 e^{bt} e^{at}.$$

La mise en application sur l'EDP

$$\frac{\partial \Phi}{\partial t}(x, t, A) + \alpha \frac{\partial^2 \Phi}{\partial x^2}(x, t, A) + \beta \frac{\partial \Phi}{\partial x}(x, t, A) + \gamma \Phi(x, t, A) + \frac{e^x}{K} \frac{\partial \Phi}{\partial A}(x, t, A) = 0.$$

Rappelons que pour une option asiatique, nous ne considérons plus le pay-off $(K - S)_+$ mais désormais $(K - A)_+$. La résolution de ce problème de Cauchy va donc se faire en deux étapes. La première consiste à trouver z pour tout A , solution de

$$\begin{cases} \frac{\partial z}{\partial t}(x, t, A) + \alpha \frac{\partial^2 z}{\partial x^2}(x, t, A) + \beta \frac{\partial z}{\partial x}(x, t, A) + \gamma z(x, t, A) = 0, \\ z(x, 0, A) = (K - A)_+. \end{cases}$$

La deuxième étape est de résoudre une équation d'advection en utilisant par exemple la méthode des caractéristiques. J'ai choisi, pour ma part, de résoudre l'équation d'advection par différences finies.

$$\begin{cases} \frac{\partial \Phi}{\partial t}(x, t, A) + \frac{e^x}{K} \frac{\partial \Phi}{\partial A}(x, t, A) = 0, \\ \Phi(x, t, 0) = z(x, t, A). \end{cases}$$

Nous savons que la solution de cette équation d'advection sera de la forme $\Phi(x, t, A) = f(x, A - \alpha t)$ et

$$\begin{aligned} \Phi(x, t^{n+1}, A) &= f(x, A - \alpha t^{n+1}) = f(x, A - \delta t - \alpha t^n) \\ &= \Phi(x, t^n, A - \alpha \delta t). \end{aligned}$$

Par la suite, nous noterons $\Phi_{i,j}^n$ l'approximation de $\Phi(x_i, t^n, A_j)$. En appliquant les formules des différences finies, on obtient que

$$\begin{aligned} &\frac{\Phi_{i,j}^{n+1} - \Phi_{i,j}^n}{\delta t} + \frac{e^x}{K} \frac{\Phi_{i,j+1}^{n+1} - \Phi_{i,j}^{n+1}}{\delta A}, \\ \Phi_{i,j}^n &= \frac{\delta t}{\delta A} \frac{e^x}{K} \Phi_{i,j+1}^{n+1} + \left(1 - \frac{\delta t}{\delta A} \frac{e^x}{K}\right) \Phi_{i,j}^{n+1} \\ &= \Phi_{i,j}^{n+1} + \frac{\delta t}{\delta A} \frac{e^x}{K} (\Phi_{i,j+1}^{n+1} - \Phi_{i,j}^{n+1}). \end{aligned}$$

On a effectivement essayé d'éviter l'inversion d'une matrice bi-diagonale en évitant une méthode implicite. Pour résoudre cette EDP, nous devons itérer cette étape jusqu'au temps initial $t^0 = 0$ (Algorithm 5).

Algorithm 5 Algorithme de splitting permettant de résoudre l'équation (E_3)

```
for  $n \in \llbracket 1; N \rrbracket$  do  
     $tn = T - ndt$   
    for  $k \in \llbracket 0; M - 2 \rrbracket$  do  
         $\Phi^i(:, k) = \text{Black-Scholes-Crank-Nicolson}();$   
    end for  
    for  $j \in \llbracket 0; M - 2 \rrbracket$  do  
         $\Phi^i(j, :) = \text{advection}();$   
    end for  
end for
```

4 Cas tests

Dans cette partie, nous essayons de résoudre les EDP permettant d'obtenir le prix de la prime de différentes options. Nous essayons également de faire bouger certaines caractéristiques par exemple.

4.1 Tests des options vanille

Dans un premier temps, nous avons travaillé sur un modèle où la discrétisation en pas de temps et d'espace est uniforme.

4.1.1 Visualisation de différentes méthodes en fonction du temps

J'ai, dans un premier temps, représenté les différentes primes obtenues pour une option vanille de vente (Put) dans le cas de la méthode implicite (figure 6) et de celle de Crank-Nicolson (figure 7). Nous pouvons remarquer que lorsque nous nous rapprochons de l'instant $t^0 = 0$, la courbe rappelle de plus en plus la solution de la formule fermée de Black-Scholes.

4.1.2 Influence de la volatilité

Par la suite, j'ai étudié l'influence de la volatilité sur l'estimation de la prime d'une option vanille. On se rend compte que la courbe rappelle de plus en plus la solution de la formule fermée de Black-Scholes lorsque la volatilité augmente (figure 8).

Cependant lorsque la volatilité est trop grande comme par exemple $\sigma = 1.5$ et que la maturité est de $T = 3ans$, alors la courbe devient étrange (figure 9).

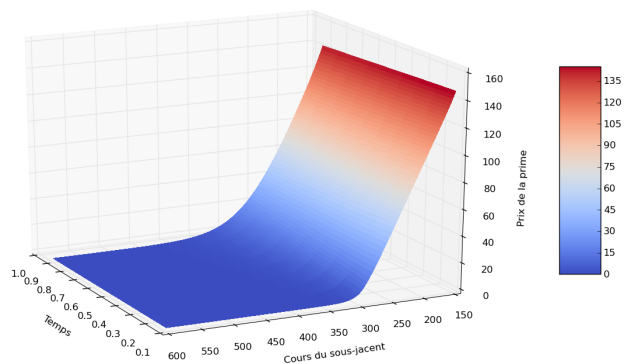


FIGURE 6 – Résolution de l'EDP avec la méthode implicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$

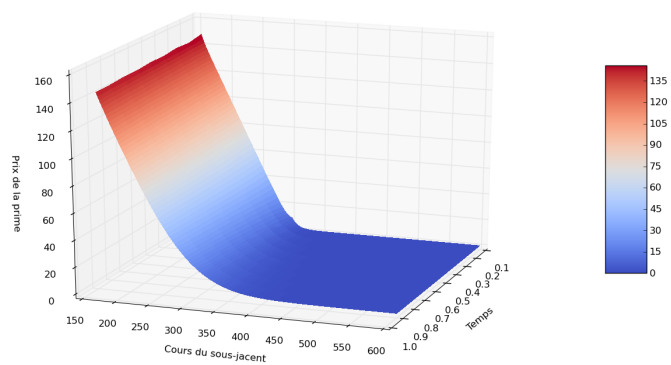


FIGURE 7 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$

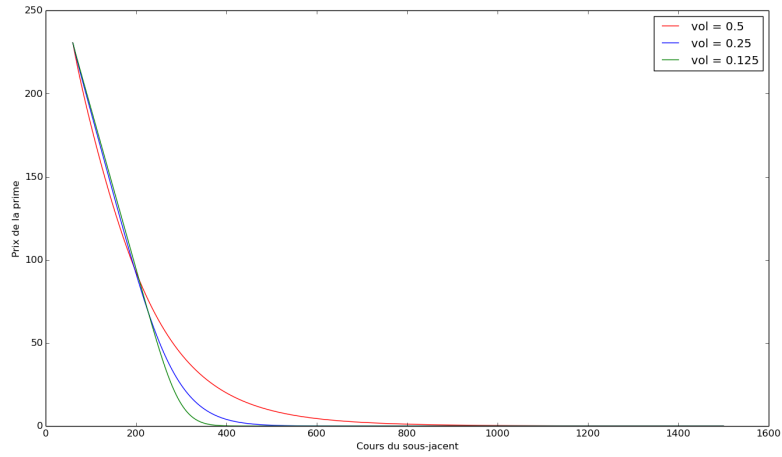


FIGURE 8 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $T = 1$ et $r = 4\%$

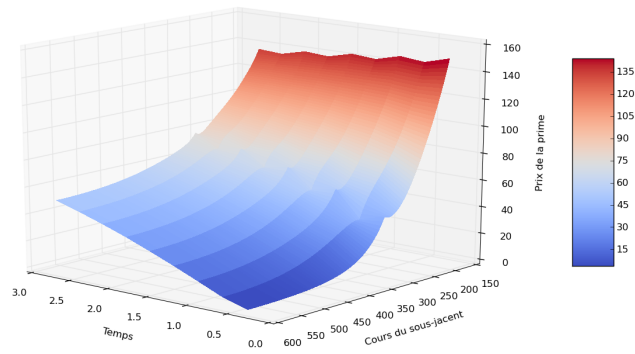


FIGURE 9 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $T = 3$, $\sigma = 1.5$ et $r = 4\%$

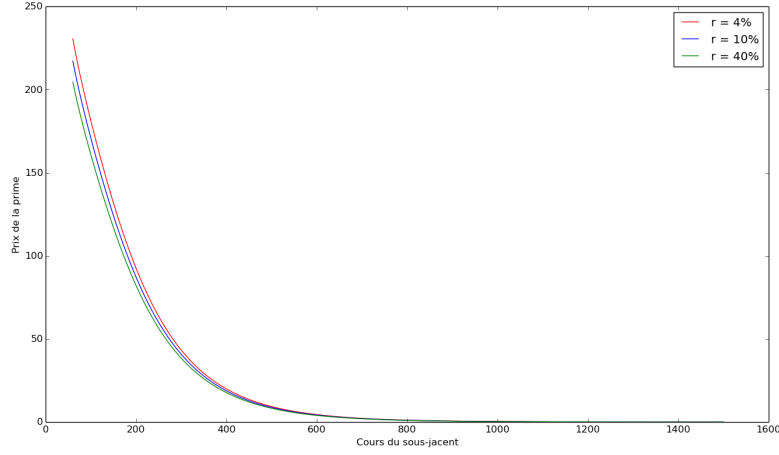


FIGURE 10 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300, T = 1$ et $\sigma = 0.5$

4.1.3 Influence du taux d'intérêt

J'ai également étudié l'influence du taux d'intérêt sur l'estimation de la prime d'une option vanille. On se rend compte que plus le taux d'intérêt augmente, plus la condition au bord gauche sera faible (figure 10). Cependant, cela n'a pas un gros impact sur le tracé général de la courbe.

4.1.4 Influence de la maturité

Par la suite, je me suis intéressé à l'influence de la maturité sur l'estimation de la prime d'une option vanille. On se rend compte que plus la maturité est grande, plus la courbe va avoir une limite non nulle en S_{max} (figure 11).

Ce phénomène a lieu pour une résolution de l'EDP avec la méthode de Crank–Nicolson mais également pour la formule fermée (figure 12).

Nous pouvons bien voir ce phénomène lorsque l'on trace les courbes en 3 dimensions (figures 13 et 14).

4.1.5 Stabilité des méthodes

La méthode explicite pose des problèmes de stabilité. La discrétisation doit effectivement respecter

$$\frac{\delta t}{\delta x^2} < \frac{1}{2\alpha} = \frac{1}{\sigma^2}.$$

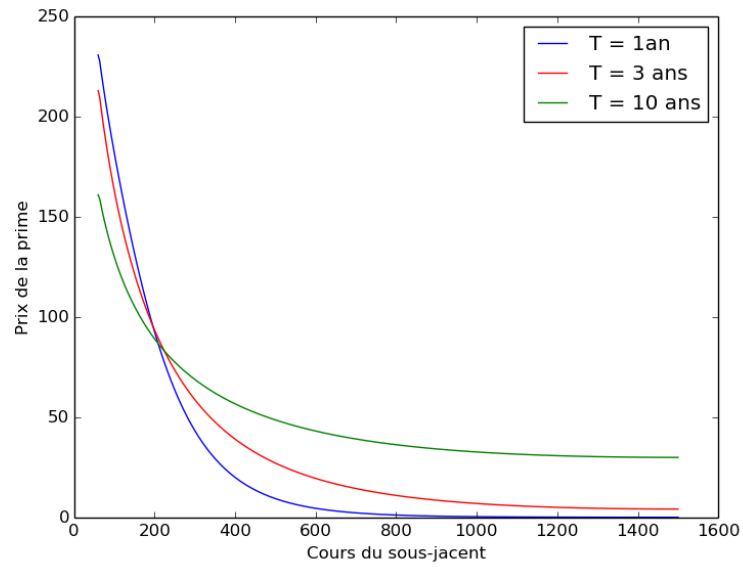


FIGURE 11 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$

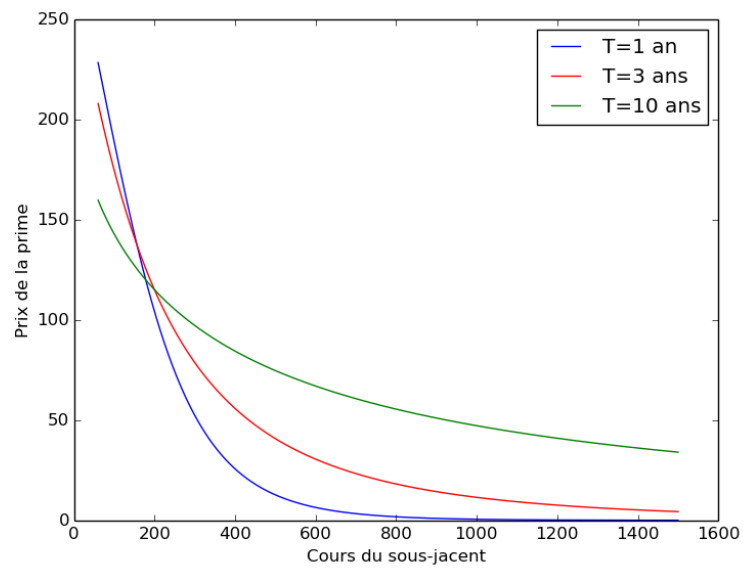


FIGURE 12 – Résolution de l'EDP avec la formule fermée pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$

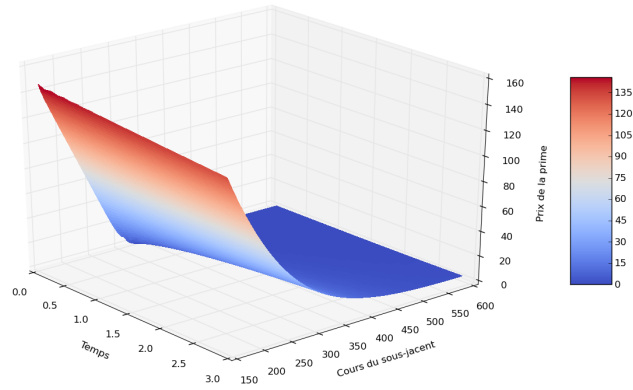


FIGURE 13 – Résolution de l’EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300, r = 4\%$ et $\sigma = 0.5$ — Tracé en fonction du temps

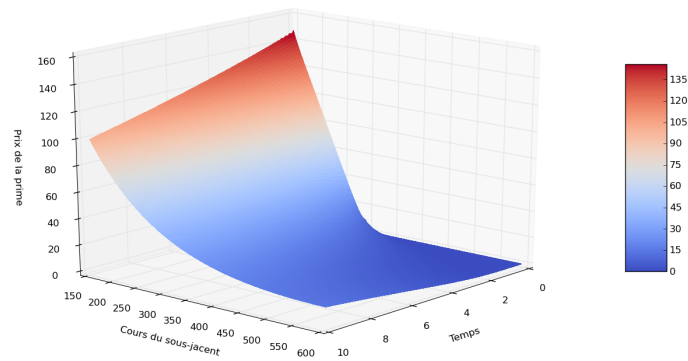


FIGURE 14 – Résolution de l’EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300, r = 4\%$ et $\sigma = 0.5$ — Tracé en fonction du temps

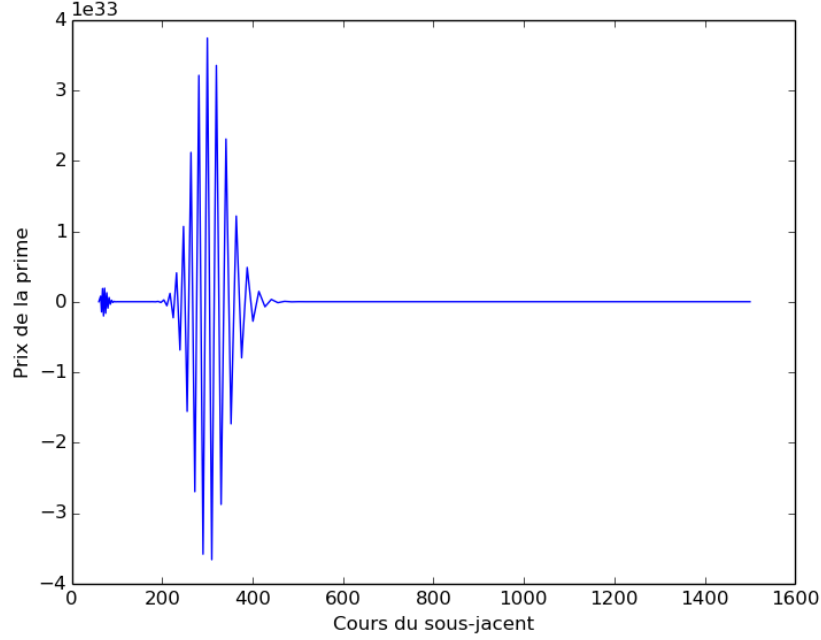


FIGURE 15 – Résolution de l’EDP avec la méthode explicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 30$

Plus particulièrement

$$M < \frac{x_M - x_0}{\sigma} \sqrt{\frac{N}{T}},$$

et en appliquant nos valeurs numériques

$$M < 2.795\sqrt{N} \quad (C_1)$$

Lorsque nous prenons des valeurs M et N qui ne respectent pas la condition (C_1) alors nous obtenons des courbes qui ne peuvent pas être exploitables (figures 15 et 16).

Cependant lorsque M et N vérifient la condition C_1 alors la courbe devient exploitable (figure 17).

La méthode implicite semble quant-à-elle beaucoup plus stable dans le sens où j’ai testé différentes combinaisons possibles de M et de N . Dans tous les cas, la courbe est exploitable car elle est inconditionnellement stable (figures 18, 19, 20 et 21).

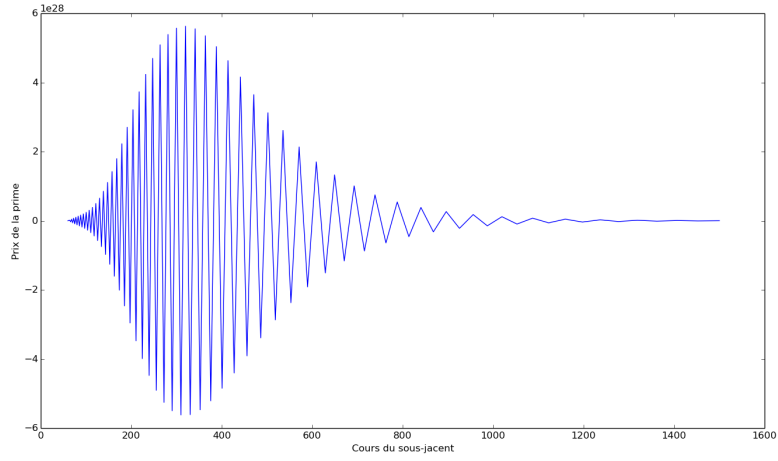


FIGURE 16 – Résolution de l'EDP avec la méthode explicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 200$

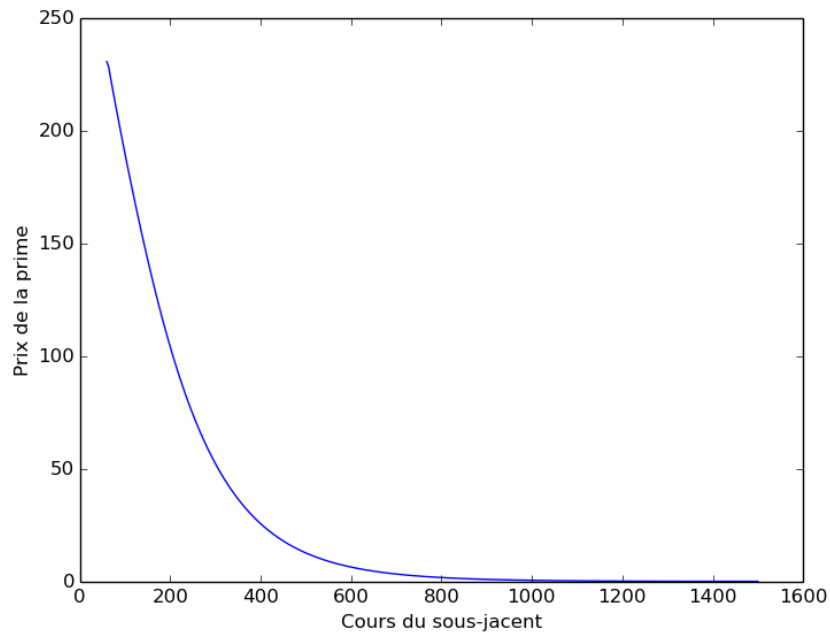


FIGURE 17 – Résolution de l'EDP avec la méthode explicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 1300$

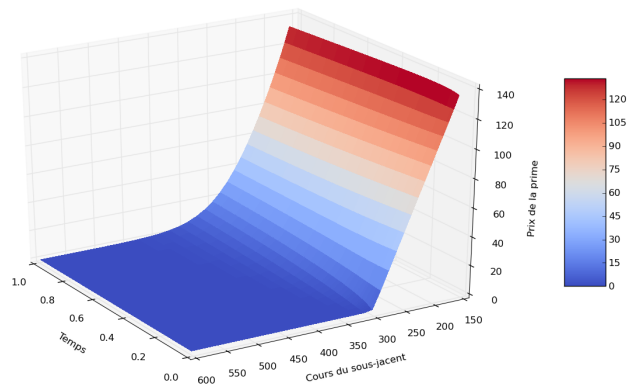


FIGURE 18 – Résolution de l'EDP avec la méthode implicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 30$ et $N = 100$

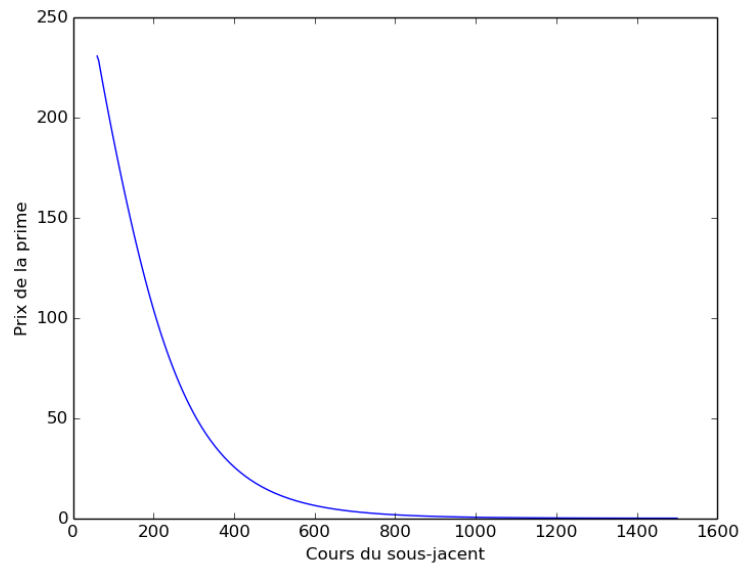


FIGURE 19 – Résolution de l'EDP avec la méthode implicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 1300$

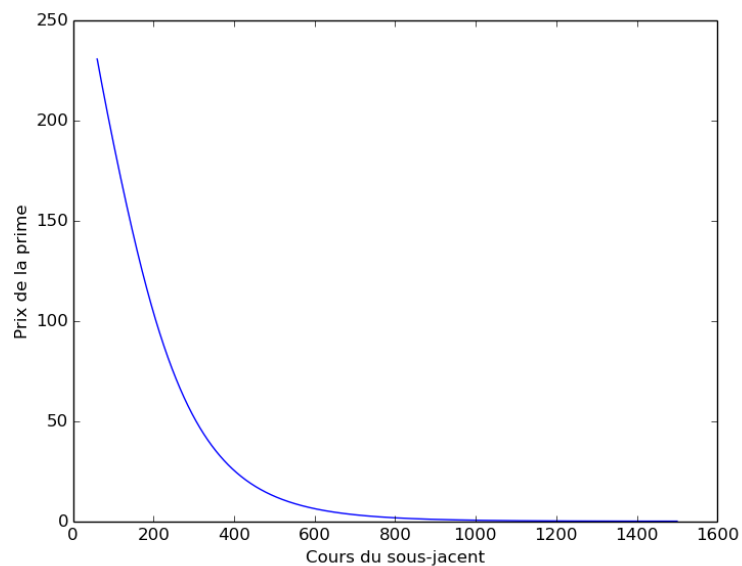


FIGURE 20 – Résolution de l'EDP avec la méthode implicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 1300$ et $N = 100$

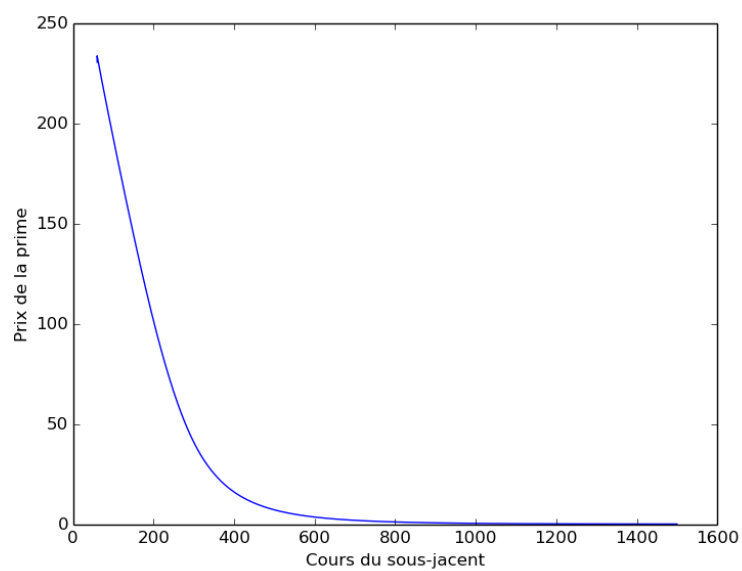


FIGURE 21 – Résolution de l'EDP avec la méthode implicite pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 10000$ et $N = 30$

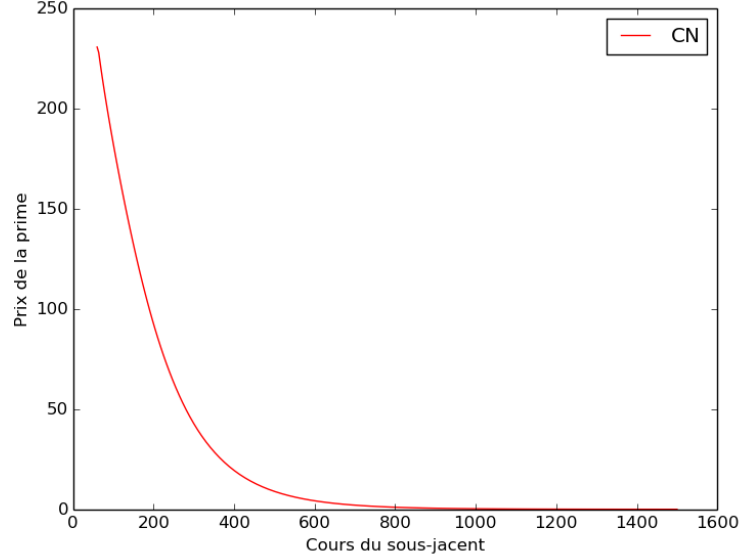


FIGURE 22 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 30$

La méthode de Crank–Nicolson semble comme la méthode implicite beaucoup plus stable que la méthode explicite. J'ai testé différentes combinaisons possibles de M et de N . Dans les deux cas extrêmes, la courbe est exploitable car cette méthode est également inconditionnellement stable (figures 22 et 23).

4.1.6 Cas où la discrétisation n'est pas uniforme

Nous aurions pu également implémenter une discrétisation de l'espace (ici le cours du sous-jacent S) qui ne serait pas uniforme. Ce raffinement autour de la valeur $S_t = K$ permet d'avoir de meilleurs résultats autour de la "singularité" du pay-off.

Sachant que $S_{min} = S_0$ et $S_{max} = S_M$, il faut discrétiser le cours du sous-jacent en $M + 1$ pas d'espace

$$\forall i \in \llbracket 0, M \rrbracket, S_i = K + \frac{K}{5} \text{sh}(\xi_i)$$

avec

$$\xi_i = \text{argsh} \left(\frac{S_0 - K}{K/5} \right) + \frac{i}{M} \left(\text{argsh} \left(\frac{S_M - K}{K/5} \right) - \text{argsh} \left(\frac{S_0 - K}{K/5} \right) \right).$$

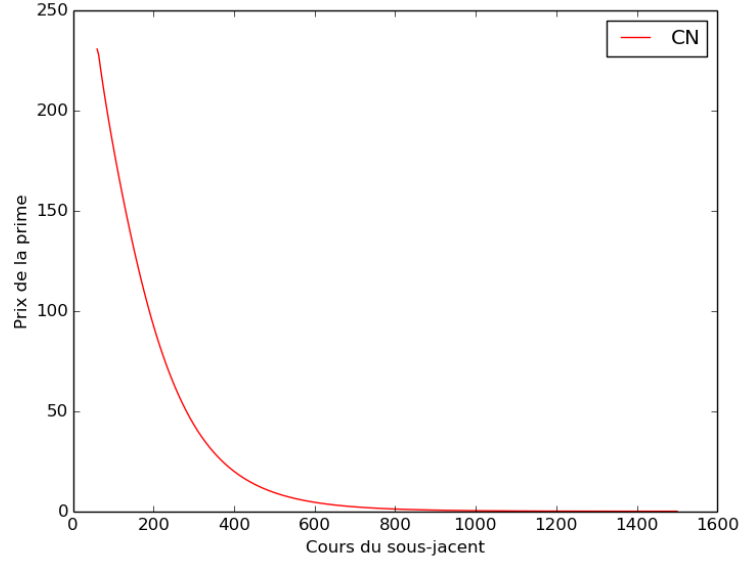


FIGURE 23 – Résolution de l'EDP avec la méthode de Crank–Nicolson pour une option vanille — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $M = 100$ et $N = 1300$

4.2 Tests des options asiatiques

Rappelons que pour les options asiatiques, le pay-off change par rapport aux options vanille; nous prenons désormais $\Psi(x, T, A) = (K - A)_+$ (figure 24).

A partir de ce pay-off à chaque itération, nous devons résoudre une itération de Black-Scholes "normal" en prenant comme pay-off pour chaque A le pay-off d'une option vanille. C'est ici que se pose le problème de conditions aux bords. En ôtant par exemple l'équation d'advection et en prenant que l'on doit avoir $z(x, 0, A) = (K - A)_+$ à chaque A , nous obtenons la figure 25.

A la fin des itérations au temps $t^0 = 0$, nous obtenons les figures 26 et 27. Nous pouvons remarquer que l'équation d'advection pose également des problèmes de bords, mais la courbe ne présente pas de discontinuité majeure.

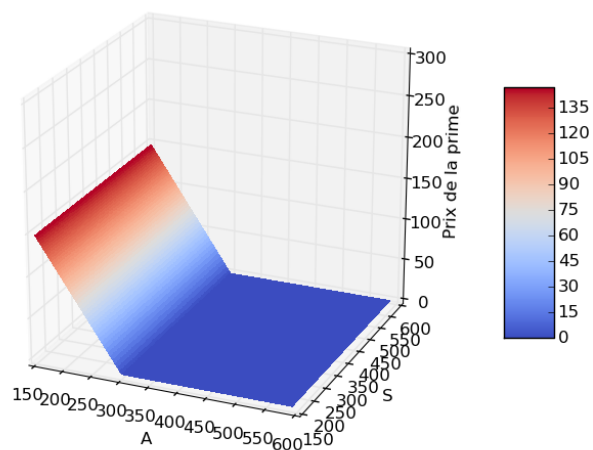


FIGURE 24 – Pay-off d'une option asiatique — $K = 300, r = 4\%$ et $\sigma = 0.5$
— $t^n = T$

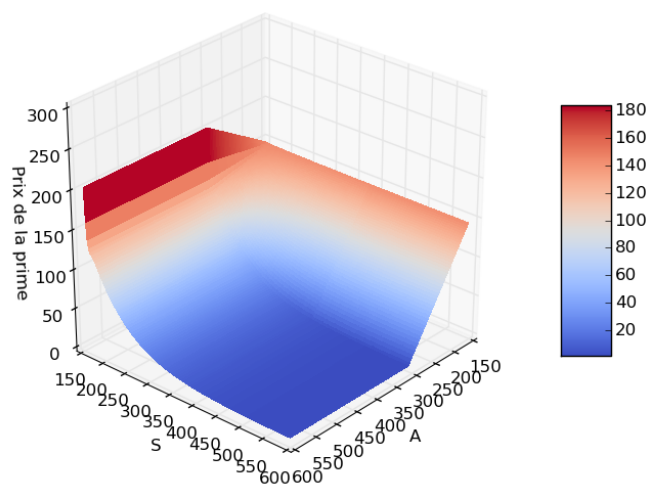


FIGURE 25 – Itérations de Black-Scholes "classiques" sur une option asiatique — $K = 300, r = 4\%$ et $\sigma = 0.5$ — $t^n = 0$

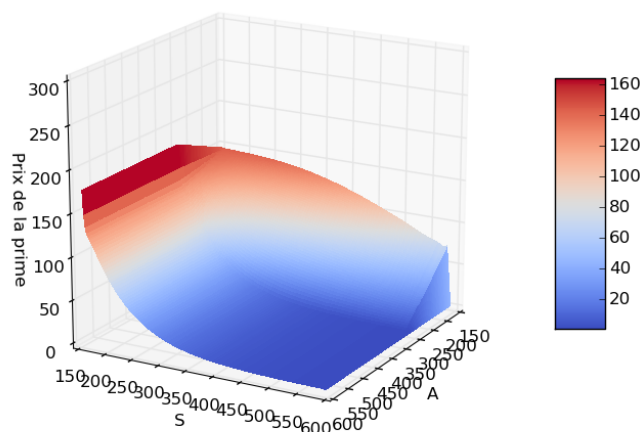


FIGURE 26 – Résolution de l'EDP d'une option asiatique via une méthode de splitting — $K = 300, r = 4\%$ et $\sigma = 0.5$ — $t^n = 0$ — $N = 10$

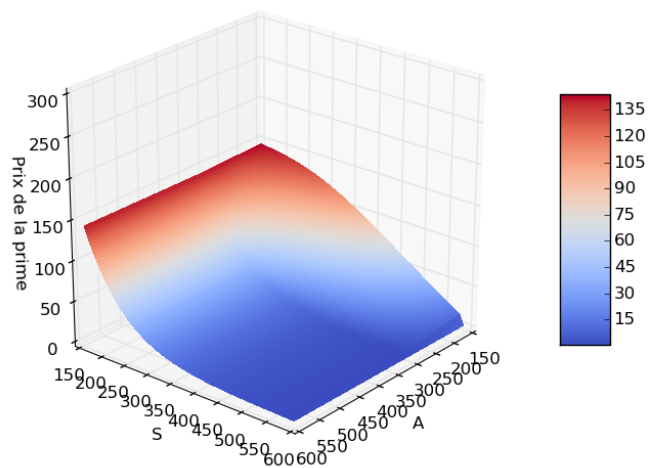


FIGURE 27 – Résolution de l'EDP d'une option asiatique via une méthode de splitting — $K = 300, r = 4\%$ et $\sigma = 0.5$ — $t^n = 0$ — $N = 100$

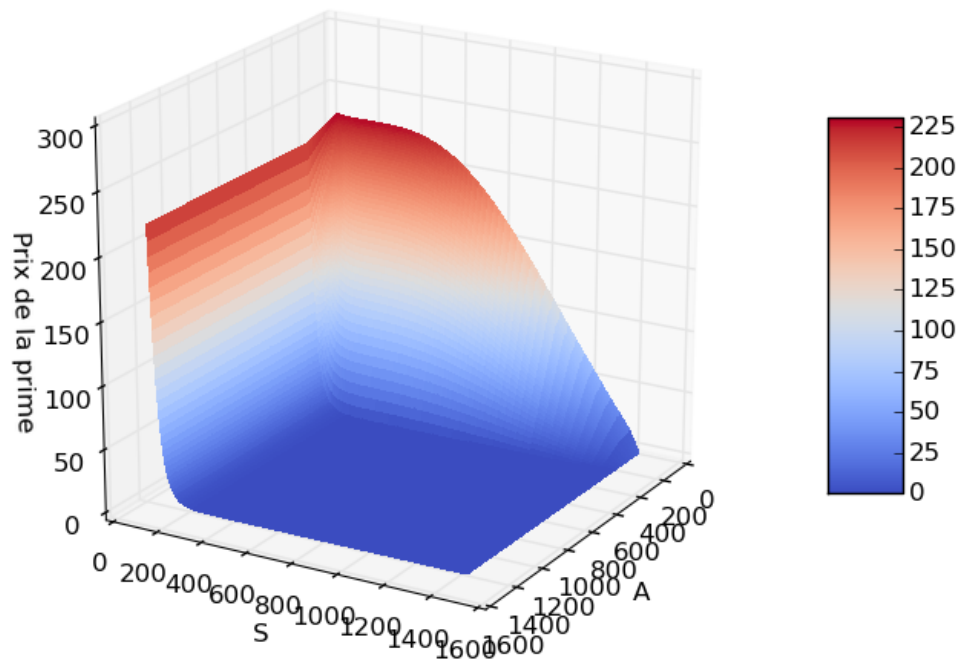


FIGURE 28 – Résolution de l'EDP d'une option asiatique via une méthode de splitting — $K = 300$, $r = 4\%$ et $\sigma = 0.5$ — $t^n = 0$

5 Conclusion

Lorsque j'étais en Prépa, j'avais déjà travaillé sur le pricing des options vanille. En effet, lors de mon TIPE (Travail d'Initiative Personnel Encadré), j'avais étudié l'approche par arbres binomiaux (*Les arbres binomiaux au service de la finance*). Cette approche permet de rester en temps discret et de pricer des options européennes à partir du modèle CRR.

Cependant ce modèle présente beaucoup de faiblesse surtout en ce qui concerne la rapidité de calcul. C'est pourquoi, j'ai travaillé ce semestre avec l'aide de Madame Bidégaray-Fesquet sur les *EDP pour les options asiatiques*. En effet, pricer une option par la résolution d'une EDP peut-être une méthode alternative à celle de Monte-Carlo, très souvent utilisée dans le monde de la finance.

Dans un premier temps, j'ai travaillé sur les options vanille. J'ai essayé de comparer les résultats obtenus par la résolution de l'EDP de Black-Scholes et la formule fermée. Dans ce cas simple, les conditions aux bords restent assez simples et les deux modèles donnent des résultats similaires.

Cependant, lorsque nous souhaitons étudier les options asiatiques, la plupart des recherches déjà réalisées posent un changement de variables $u = S/A$ [3] permettant de se ramener à une variable spatiale au lieu de deux. De cette manière, l'étude des conditions de bord reste assez simple.

Néanmoins, nous avons choisi d'essayer de résoudre cette EDP en gardant les deux variables spatiales et la variable temporelle. Comme nous avons supposé que la variable qui représente le cours du sous-jacent et celle qui représente la moyenne de ce cours restent indépendantes, alors il est assez difficile d'imposer de "bonnes conditions" de bord. A ce jour, je n'ai pas encore trouvé les conditions permettant d'obtenir un résultat correct.

6 Annexes

6.1 Algorithme PYTHON qui permet de pricer une option vanille ou asiatique

Algorithme permettant de visualiser la prime d'une

- option vanille obtenue grâce à
 - la formule fermée de Black–Scholes,
 - la résolution de l'EDP (Crank–Nicolson avec discrétisation homogène),
 - la résolution de l'EDP (méthode implicite avec discrétisation homogène)
 - la résolution de l'EDP (méthode du prix de explicite avec discrétisation homogène)
- option asiatique obtenue par la résolution de l'EDP (Splitting — Crank–Nicolson avec discrétisation homogène).

```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import matplotlib.pyplot as plt
from numpy import *
import json
from math import exp
from math import erf
from math import log
from math import sqrt

def graph(vect, string):
    plt.plot(vect[1], vect[0], 'r-', label = string)
    plt.xlabel("S")
    plt.ylabel("Prix de la prime")
    plt.legend()
    plt.show()

def vanille_edp(Smin, Smax, M, K, N, r, tN, vol, theta):

    # Definition des constantes de l'EDP
    alpha = float(vol)**2/2
    beta = float(r)- alpha
    gamma = - float(r)

    # Definition des pas de discretisations
    dx = (log(Smax)-log(Smin))/float(M)
    tabX = [log(Smin) + (i+1)*dx for i in range(1,M)]
```



```

tabS = [exp(x) for x in tabX]
dt = tN/float(N)

vect = [max(K-s,0) for s in tabS]

# Crank Nicolson
if (theta == 0.5):
    string = "Crank-Nicolson"
# Definition des coefficients de la matrice explicite
ksi1 = dt*(alpha/(2*dx**2) - beta/(4*dx))
ksi2 = 1 + dt*(gamma/2 - alpha/(dx**2))
ksi3 = dt*(alpha/(2*dx**2) + beta/(4*dx))

# Definition des coefficients de la matrice implicite
zeta1 = -dt*(alpha/(2*dx**2) + beta/(4*dx))
zeta2 = 1 + dt*(-gamma/2 + alpha/(dx**2))
zeta3 = -dt*(alpha/(2*dx**2) - beta/(4*dx))

for i in range(1,N):
    tn = tN - i*dt;
    A = tridiag(ksi1,ksi2,ksi3,len(vect))
    B = tridiag(zeta1,zeta2,zeta3,len(vect))
    cond = zeros(len(vect));
    cond[0]=(ksi3-zeta3*exp(r*dt))*(K-Smin)*exp(-r*(tN-tn))
    vect = mult(A,vect);
    vect += cond;
    vect = inverse(B, vect)

# Methode implicite
elif (theta ==1):
    string = "Imp"
    # Definition des coefficients de la matrice implicite
    zeta1 = -dt*(alpha/(dx**2) + beta/(2*dx))
    zeta2 = 1 + dt*(-gamma + 2*alpha/(dx**2))
    zeta3 = -dt*(alpha/(dx**2) - beta/(2*dx))

    for i in range(1,N):
        tn = tN - i*dt
        B = tridiag(zeta1,zeta2,zeta3,len(vect))
        cond = zeros(len(vect))
        cond[0]=(zeta3)*(K-Smin)*exp(-r*(tN-tn))
        vect = inverse(B, vect-cond)

elif (theta == 0):
    string = "Exp"
    # Definition des coefficients de la matrice explicite
    ksi1 = dt*(alpha/(dx**2) + beta/(2*dx));
    ksi2 = 1 + dt*(gamma - 2*alpha/(dx**2));
    ksi3 = dt*(alpha/(dx**2) - beta/(2*dx));

```

```

vect = [max(K-s,0) for s in tabS];

for i in range(1,N):
    tn = tN - i*dt
    # Calcule simplement Phi(i-1)=A*Phi(i)+Conditions
    A = tridiag(ksi1, ksi2, ksi3, len(vect));
    cond = zeros(len(vect))
    cond[0]=(ksi3)*(K-Smin)*exp(-r*(tN-tn))
    # Condition de Dirichlet
    vect = mult(A,vect)
    vect += cond
    #vect = itExp(ksi1,ksi2,ksi3,K,Smin,r,tN,tn,dt, vect)

else :
    print("Ce n'est pas possible")

tabSbis = [Smin]+tabS+[Smax];
vectbis = [(K-Smin)*exp(-r*(tN))]+vect.tolist()+[vect[len(vect)-1]]
result = [vectbis, tabSbis]
graph(result, string)
return 0

def formule_fermee(Smin, Smax, M, K, r, tN, vol):

    ds = (Smax-Smin)/float(M);
    tabS = [Smin+i*ds for i in range(0,M+1)]
    tabd2 = [(log(i/float(K)))+(r-(vol*vol)/2)*tN) for i in tabS]
    tabd2 = [i/(float(vol)*sqrt(tN)) for i in tabd2];
    tabd1 = [i + vol*sqrt(tN) for i in tabd2];
    tabd1 = [tabS[i]*(1+erf(-tabd1[i]/sqrt(2)))/2 for i in range(0,M+1)];
    tabd2 = [K*exp(-r*tN)*(1+erf(-tabd2[i]/sqrt(2)))/2 for i in range(0,M+1)];
    vect = [tabd2[i]-tabd1[i] for i in range(0,M+1)];

    result = [vect, tabS];

    graph(result, "Formule fermee")
    return 0

def asiat(Smin, Smax, M, K, N, r, tN, vol):

    print 'initialisation des matrices'

    # Definition des constantes de l'EDP
    alpha = vol**2/2
    beta = r - alpha
    gamma = -r

```

```

# Definition des pas de discretisations
dx = (log(Smax)-log(Smin))/M
tabX = [log(Smin) + k*dx for k in range(1,M)]
tabS = [exp(x) for x in tabX]
tabA = tabS
dt = tN/N
print 'cfl:' + str(dt/dx/dx*vol*vol)

# Definition des coefficients de la matrice explicite
ksi1 = dt*(alpha/(2*dx**2) - beta/(4*dx))
ksi2 = 1 + dt*(gamma/2 - alpha/(dx**2))
ksi3 = dt*(alpha/(2*dx**2) + beta/(4*dx))

# Definition des coefficients de la matrice implicite
zeta1 = -dt*(alpha/(2*dx**2) + beta/(4*dx))
zeta2 = 1 + dt*(-gamma/2 + alpha/(dx**2))
zeta3 = -dt*(alpha/(2*dx**2) - beta/(4*dx))

vect=zeros((M-1,M-1))
for j in range(0,M-1):
    for k in range(0,M-1):
        vect[j,k] = max(K-tabA[k],0)

print 'boucle en temps'
for n in range(1,N):
    tn = tN - n*dt
    print tn
    temp = zeros((M-1,M-1))
    for k in range(0,M-1): # boucle sur a
        A = tridiag(ksi1,ksi2,ksi3,len(vect))
        B = tridiag(zeta1,zeta2,zeta3,len(vect))
        vect[:,k] = mult(A,vect[:,k])
        vect[0,k] = vect[0,k] + (ksi1-zeta1*exp(r*dt))*(K-Smin)*exp(-r*(tN-tn))
        vect[:,k] = inverse(B, vect[:,k])
    for j in range(0,M-2): # boucle sur x
        vectjk = vect[j,:] # vect[j,k]
        vectj1k = vect[j+1,:] # vect[j+1,k]
        temp[j,:] = vectjk + (tabX[j]*dt)*(vectj1k - vectjk)/(K*dx)
    vect = temp;

print 'visualisation'
fig = plt.figure()
ax = fig.gca(projection='3d')
X, Y = meshgrid(tabS,tabA)
surf = ax.plot_surface(X, Y, vect, rstride=1, cstride=1, cmap=cm.coolwarm, linewidth=0)
maxZ = max(tabS)-max(tabA)
ax.set_zlim(0, K)
plt.xlabel("A")
plt.ylabel("S")

```

```

ax.set_zlabel("Prix de la prime")
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()

def tridiag(a, b, c, M):

    # Initialise la matrice tridiagonale
    # mat[:,0]=a sur-diagonale, mat[:,1]=b diagonale, mat[:,2]=c sous-diagonale
    mat = zeros((M,3))
    mat[:,0] = a
    mat[:,1] = b
    mat[:,2] = c
    # ligne 0, condition de Dirichlet
    mat[0,2] = 0.
    # ligne M-1, condition de Neumann
    mat[M-1,0] = 0.
    mat[M-1,1] = b+a
    return mat;

def mult(A,x):
    M = len(x)
    vect = zeros(M)
    vect[0]=x[0]*A[0,1]+x[1]*A[0,0]
    for i in range(1,M-1):
        vect[i]=x[i]*A[i,1]+x[i-1]*A[i,2]+x[i+1]*A[i,0]
    vect[M-1]=x[M-1]*A[M-1,1]+x[M-2]*A[M-1,2]
    return vect

def inverse(B,x):
    M = len(x)
    vect = zeros(M)
    B[0,0]=B[0,0]/B[0,1]
    x[0]=x[0]/B[0,1]
    for i in range(1,M-1):
        temp = B[i,1] - B[i,2]*B[i-1,0]
        B[i,0] = B[i,0]/temp
        x[i] = (x[i]-B[i,2]*x[i-1])/temp
    x[M-1] = (x[M-1]-B[M-1,2]*x[M-2])/(B[M-1,1]-B[M-1,2]*B[M-2,0])
    vect[M-1] = x[M-1]
    for i in range(M-2,-1,-1):
        vect[i] = x[i] - B[i,0]*vect[i+1]
    return vect

def main():
    fichier = open("test.json", "r")
    param = json.load(fichier)
    fichier.close()
    Smin = float(param["Smin"])

```

```

Smax = float(param["Smax"])
M = int(param["M"])
K = float(param["K"])
N = int(param["N"])
r = float(param["r"])
tN = float(param["tN"])
vol = float(param["vol"])
theta = float(param["theta"])
type = int(param["type"])
if (type == 0):
    vanille_edp(Smin, Smax, M, K, N, r, tN, vol, theta)
elif (type == 1):
    formule_fermee(Smin, Smax, M, K, r, tN, vol)
else:
    asiat(Smin, Smax, M, K, N, r, tN, vol)

main()

```

6.2 Fichier Json permettant la lecture des paramètres

Afin de rentrer tester mon code PYTHON, directement en console, j'ai utiliser des fichiers **.json** permettant ainsi de pouvoir récupérer les paramètres sans les retaper à chaque fois.

En plus des paramètres classiques, j'ai utilisé le paramètre **theta** permettant de savoir si nous souhaitons utiliser une méthode de Crank–Nicolson ($\theta = 0.5$), implicite ($\theta = 1$) ou explicite ($\theta = 0$). De plus, le paramètre **type** permet de sélectionner la méthode de résolution souhaitée :

- 0 correspond à résolution de l'EDP d'une option vanille,
- 1 correspond à la formule fermée de Black–Scholes,
- 2 correspond à la résolution de l'EDP d'une option asiatique.

```

{
  "Smin":60,
  "Smax":1500,
  "M": 100,
  "K": 300,
  "N": 100,
  "r": 0.04,
  "tN": 1,
  "vol":0.50,
  "theta":1,
  "type":2
}

```

References

- [1] F. Black and M. Scholes. The pricing of options and corporate liabilities. 81(3):637–654, May-June 1973.
- [2] J.C. Hull. *Options, Futures and Other Derivatives*. Prentice Hall, 6th edition, 2005.
- [3] P.A. Forsyth R. Zvan and K. Vetzal. Robust numerical methods for pde models of Asian options.
- [4] L.C.G. Rogers and Z. Shi. The value of an Asian option. 32(4):1077–1088, December 1995.
- [5] R. U. Seydel. *Tools for Computational Finance*. Springer, 5th edition.

Remerciements

À Madame Bidégaray–Fesquet pour avoir accepté de devenir ma tutrice pendant ces derniers mois et ainsi de m’aider sur ce sujet qui m’a permis de découvrir beaucoup de choses que ce soit en dans le domaine des mathématiques, de la finance et même dans celui de l’informatique. Je la remercie plus particulièrement pour sa disponibilité et son indulgence sur mes connaissances qui étaient peut-être trop jeunes au début de ce projet. En dépit de son emploi du temps chargé, elle a quand même réussi à m’aider et m’écouter lorsque je rencontrais des difficultés.

A toute l’équipe du laboratoire du LJK et plus particulièrement l’équipe EDP qui m’a entourée et grâce à laquelle j’ai pu découvrir la vie d’un laboratoire de recherche, assister à un séminaire, etc.

Enfin à l’ENSIMAG, aux chercheurs et particulièrement à Madame Cani qui en mettant en place ces modules, nous permettent de découvrir et d’approfondir des connaissances que nous n’étudions pas forcément dans le cadre du tronc-commun de nos années d’études.