

Luc Sagnes

Jessica Gourdon

RAPPORT PROJET MÉTHODES NUMÉRIQUES

RÉSOLUTION DE L'ÉQUATION DE BLACK-SCHOLES ET DE LA CHALEUR

Table des matières

Introduction	2
Équation de Black-Scholes	3
Écriture de l'équation	3
Notation	3
Hypothèse du modèle	3
Condition aux limites et condition initiale	4
Lien entre L'équation de Black-Scholes et la chaleur	4
Résolution Analytique	5
Résolution numérique	6
Méthode d'Euler Explicite pour l'équation de la chaleur	7
Méthode d'Euler Implicite pour l'équation de la chaleur	8
Résolution numérique de l'équation de Black-Scholes	11
Résultat	12
Précision des modèles	12
Influence du nombre de points	15
Importance du coefficient d'Euler	Erreur ! Signet non défini.
Conclusion	19

Introduction

Dans le monde de la finance, depuis des années, les acteurs font face à une réalité : le prix des actions évolue constamment au cours du temps.

Lorsque qu'un acteur achète ou vend une action, il est intéressant pour ce dernier de se couvrir du risque correspondant à la situation suivante : le prix de l'action diminue ou augmente à l'encontre de sa volonté. Par exemple, un investisseur achète une action dont le prix ne cesse de chuter par la suite.

Ce risque peut, en partie, être couvert par le biais d'un contrat d'option. Celui-ci permet à son souscripteur de se couvrir contre la hausse ou baisse des prix d'un actif sous-jacent. En réalité, le contrat représente un droit (et non une obligation) d'achat ou de vente d'un actif à un prix donné, le prix d'exercice, sur une période donnée.

Plusieurs méthodes ont été mises en place concernant la gestion des risques des portefeuilles. Certains d'entre eux, comme la méthode de Monte-Carlo, se basent sur des discrétisations d'équations différentielles stochastiques. Des modèles de discrétisation ont alors été créés afin de converger vers la valeur d'un processus stochastique.

C'est dans ce contexte que l'équation de Black-Scholes apparaît. Celle-ci permet de supprimer le caractère stochastique d'une action afin de la rendre complètement déterministe. Bien que cette équation n'assure pas une parfaite fiabilité, elle reste tout de même utilisée dans de nombreux domaines de la finance et permet d'obtenir une évaluation des prix d'options efficace.

L'objectif de ce projet est de mettre en place différents schémas afin de déterminer une solution numérique de l'équation de Black-Scholes pour ensuite comparer nos résultats à la solution analytique de cette équation. Par la suite, nous déterminerons l'erreur de nos modèles, étudierons leurs convergences et évaluerons l'influence des différents paramètres pris en compte.

Équation de Black-Scholes

Écriture de l'équation

Le modèle de Black-Scholes est une équation différentielle qui permet de déterminer l'évolution du prix d'une option Européenne, Call ou Put. Elle s'écrit de la manière suivante :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

Notation

Nous allons mettre en place un système de notation qui sera utilisée tout au long du rapport :

- S : valeur de l'actif sous-jacent
- r : rendement sans risque (constant ici)
- σ : volatilité
- t : temps
- $V(t, S)$: valeur de l'option (Soit un Call C ou un Put P) qui représente la valeur de l'option à un temps donné concernant un actif sous-jacent de valeur S
- K : prix d'exercice (Strike), valeur de l'actif sous-jacent à partir de laquelle nous pouvons exercer notre option

Hypothèse du modèle

Le modèle de Black-Scholes sous-entend de nombreuses hypothèses permettant de rendre l'équation possible et d'obtenir de bons résultats :

- Le prix de l'actif sous-jacent S suit un mouvement brownien géométrique et donc sa volatilité σ est constante. Cette hypothèse est cependant fausse dans la vie réelle, car les volatilités ne sont pas constantes à travers le temps.
- Il n'y a pas d'opportunité d'arbitrage. Cette hypothèse est courante dans de nombreux modèles financiers afin d'obtenir un modèle équilibré.
- Il est possible d'effectuer des ventes à découvert.
- Il existe un taux d'intérêt sans risque constant. Cette hypothèse peut également s'avérer fausse dans la vie réelle, car les taux peuvent varier au cours du temps.
- Aucun dividende n'est versé, dans le cas d'une action, entre le moment de l'évaluation de l'option et son échéance.

Une fois ces hypothèses validées, on parle de cas Black-Scholes. Ces hypothèses ne correspondent pas à la réalité des marchés financiers, mais elles permettent, en revanche, d'obtenir des résultats pertinents.

Condition aux limites et condition initiale

Afin de pouvoir résoudre l'équation de Black-Scholes, il est nécessaire de poser des conditions aux bords pour le prix de l'action S et le temps t . La valeur de l'actif S doit au départ être comprise en $[0 ; +\infty[$. En effet, le prix d'une action doit obligatoirement être une valeur positive. Pour ce qui est du temps, nous avons deux possibilités : la première consiste à fixer la condition initiale à la signature du contrat de l'option et la deuxième à la fixer à échéance du contrat. Cependant, la résolution de l'équation de Black-Scholes se fait par récurrence rétrograde, on fixe donc la condition en $t = T$.

Dans le cas d'un Call ($V(t, S) = C(t, S)$), à $t=T$, la valeur du call est donnée par la formule :

$$C(T, S) = \max(S - K, 0) \quad \forall S > 0$$

Si le prix de l'action S dépasse le strike K , nous pouvons exercer notre option. Nous gagnerons la différence de prix entre S et K . Dans le cas où le prix de l'action ne dépasse pas le strike, alors nous perdons notre option.

Pour les conditions aux limites, il faut regarder la valeur d'un call lorsque $S = 0$ et $S \rightarrow +\infty$:

Lorsque $S = 0$, on comprend bien que le bénéfice sera nul.

Donc, $C(0, t) = 0 \quad \forall t \in [0, T]$

Quand $S \rightarrow +\infty$, on a :

$C(t, S) = S - K$ donc à l'échéance $\rightarrow S - Ke^{-r(T-t)}$ représentant la valeur du strike actualisé au taux r entre t et T .

Dans le cas d'un put, on obtient, par un raisonnement symétrique, des conditions aux limites avec $P(T, S) = \max(K - S, 0)$ à l'échéance T ,

$P(t, 0) = Ke^{-r(T-t)}$ et $P(t, S) = 0$ $\underset{S \rightarrow \infty}{}$.

Lien entre L'équation de Black-Scholes et la chaleur

Il est possible de résoudre l'équation de Black-Scholes en se ramenant à l'équation de la chaleur. Cette dernière, très populaire, a été étudiée de nombreuses fois et se trouve être plus simple à résoudre.

L'équation de Black-Scholes sous forme d'un Call s'écrit de la manière suivante :

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0$$

En procédant à des changements de variables :

- Nouvelle variable temporelle : $\tau = \frac{1}{2} \sigma^2 (T - t)$, $\tau: 0 \rightarrow \frac{1}{2} \sigma^2 T$
- Nouvelle variable d'espace : $x = \ln(\frac{S}{K})$

Avec : $k = \frac{r}{\frac{1}{2}\sigma^2}$,

$$u(\tau, x) = e^{-\alpha x - \beta \tau} \frac{C(\tau, x)}{K}, \quad \alpha = \frac{-(k-1)}{2} \quad \text{et} \quad \beta = \alpha^2 + (k-1)\alpha - k$$

Après les différentes transformations, on obtient :

$$\frac{\partial C}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0 \quad \Leftrightarrow \quad \frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$$

Résolution Analytique

La résolution analytique de Black-Scholes se calcule en résolvant :

$$\begin{cases} \frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \\ u(\tau = 0, x) = u_0(x) \end{cases}$$

avec, pour un Call, $u_0(x) = \max(e^{\frac{k+1}{2}x} - e^{\frac{k-1}{2}x}, 0)$

La solution de cette équation est donnée par : $u(\tau, x) = \frac{1}{2\sqrt{\pi\tau}} \int_{-\infty}^{+\infty} u_0(y) e^{-\frac{(x-y)^2}{4\tau}} dy$

En faisant le calcul de l'intégrale et les changements de variables inverses, on obtient :

$$C(t, S) = S \mathcal{N}(d_1) - K e^{-r(T-t)} \mathcal{N}(d_2)$$

$$\text{Avec : } d_1 = \frac{\ln(\frac{S}{K}) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \quad \text{et} \quad d_2 = \frac{\ln(\frac{S}{K}) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

$$\text{Et : } \mathcal{N}(d) = \frac{1}{2\sqrt{\pi}} \int_{-\infty}^d e^{-\frac{x^2}{2}} dx = P(\mathcal{N}(0,1) \leq d)$$

Qui est la fonction de répartition d'une loi normale centrée réduite.

Résolution numérique

L'objectif de ce projet est de parvenir à résoudre l'équation de Black-Scholes. Pour cela, nous commencerons par résoudre numériquement l'équation de la chaleur, dont la résolution est plus simple. Nous allons ensuite nous servir des changements de variables décrits dans ce rapport afin d'obtenir la solution de l'équation de Black-Scholes. Pour rappel, l'équation de la chaleur est :

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$$

Afin de résoudre numériquement une équation aux dérivées partielles, nous devons discrétiser le domaine d'étude considéré en temps et en espace. Le domaine d'étude sera donc découpé, à l'exemple d'une grille, en un nombre fini de points et d'intervalles. La solution sera calculée à chacun de ces points.

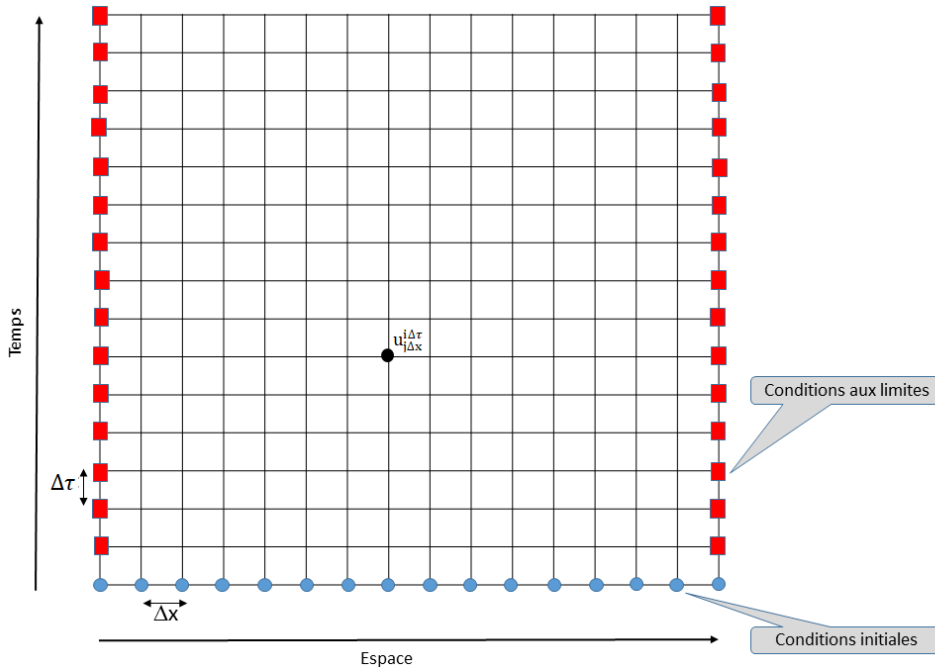
L'intervalle temporel $[0, \frac{1}{2}\sigma^2 T]$ sera discrétisé en $M+1$ points :

$$\tau_i = i\Delta\tau \text{ avec } 0 \leq i \leq M, \tau_0 = 0 \text{ et } \tau_M = M\Delta\tau = \frac{1}{2}\sigma^2 T$$

Nous n'avons pas de contrainte particulière pour l'intervalle en espace, nous choisissons donc N tel que :

$$x_j = j\Delta x \text{ avec } -N \leq j \leq N$$

On notera donc : $u_j^i = u(\tau_i, x_j) = u(i\Delta\tau, j\Delta x)$



On souhaite, à présent, calculer $\frac{\partial u}{\partial \tau}(\tau, x)$ en supposant connaître $\tau + \Delta\tau$ et $\tau - \Delta\tau$. Nous allons donc introduire plusieurs schémas aux différences finies pour approximer cela.

- Différences finies décentrées à droite

$$\frac{\partial u}{\partial \tau}(\tau, x) \approx \frac{u(\tau + \Delta\tau, x) - u(\tau, x)}{\Delta\tau}$$

L'erreur d'approximation est d'ordre 1, en $\mathcal{O}(\Delta\tau)$.

- Différences finies décentrées à gauche

$$\frac{\partial u}{\partial \tau}(\tau, x) \approx \frac{u(\tau, x) - u(\tau - \Delta\tau, x)}{\Delta\tau}$$

L'erreur d'approximation est également d'ordre 1, en $\mathcal{O}(\Delta\tau)$.

- Différences finies centrées

$$\frac{\partial u}{\partial \tau}(\tau, x) \approx \frac{u(\tau + \Delta\tau, x) - u(\tau - \Delta\tau, x)}{2\Delta\tau}$$

L'erreur d'approximation est d'ordre 2, en $\mathcal{O}(\Delta\tau^2)$. Cette formule est donc beaucoup plus précise.

Méthode d'Euler Explicite pour l'équation de la chaleur

Tout d'abord, nous avons choisi d'implémenter le schéma d'Euler explicite. Celui-ci utilise un schéma aux différences finies décentrées à droite.

$$\frac{u_j^{i+1} - u_j^i}{\Delta\tau} \approx \frac{u_{j-1}^i - 2u_j^i + u_{j+1}^i}{\Delta x^2}$$

Une écriture équivalente de ce schéma est :

$$u_j^{i+1} \approx \alpha u_{j-1}^i + (1 - 2\alpha)u_j^i + \alpha u_{j+1}^i \text{ avec } \alpha = \frac{\Delta\tau}{\Delta x^2}$$

Avec ce schéma, si l'on connaît la solution à l'instant $i\Delta\tau$ aux points de l'espace $j\Delta x$, $(j + 1)\Delta x$ et $(j - 1)\Delta x$, alors on peut calculer directement (« explicitement ») la solution à l'instant $(i+1)\Delta\tau$ au point $j\Delta x$.

On tient en compte des conditions aux limites :

$$u_j^0 = u_0(j\Delta x) \text{ avec } -N \leq j \leq N$$

$$u_{-N}^i = u_{-\infty}(i\Delta\tau, -N\Delta x,) \text{ avec } 0 \leq i \leq M$$

$$u_N^i = u_{+\infty}(i\Delta\tau, N\Delta x) \text{ avec } 0 \leq i \leq M$$

L'erreur du schéma est de l'ordre $\mathcal{O}(\Delta\tau + \Delta x^2)$.

Cependant, ce schéma est stable si et seulement si $\alpha < \frac{1}{2}$.

Nous allons, d'une part, implémenter directement ce schéma. D'autre part, nous allons implémenter la résolution de ce schéma sous forme matricielle tel que :

$$\begin{pmatrix} 1-2\alpha & \alpha & 0 & \dots & 0 \\ \alpha & 1-2\alpha & \alpha & & 0 \\ 0 & \alpha & \ddots & & \\ \vdots & \vdots & \ddots & & \alpha \\ 0 & 0 & 0 & \dots & 1-2\alpha \end{pmatrix} \begin{pmatrix} u_{-N+1}^{i-1} \\ \vdots \\ u_0^{i-1} \\ \vdots \\ u_{N-1}^{i-1} \end{pmatrix} + \begin{pmatrix} \alpha u_{-N}^i \\ 0 \\ \vdots \\ 0 \\ \alpha u_N^i \end{pmatrix} = \begin{pmatrix} u_{-N+1}^i \\ \vdots \\ u_0^i \\ \vdots \\ u_{N-1}^i \end{pmatrix}$$

\downarrow
 A

\downarrow
 U_{i-1}

\downarrow
 αb_{i-1}

\downarrow
 U_i

Nous allons donc comparer ces deux méthodes explicites.

Méthode d'Euler Implicite pour l'équation de la chaleur

Ensuite, nous avons choisi d'implémenter le schéma d'Euler implicite. Celui-ci utilise un schéma aux différences finies décentrées à gauche.

$$\frac{u_j^i - u_j^{i-1}}{\Delta\tau} \approx \frac{u_{j-1}^i - 2u_j^i + u_{j+1}^i}{\Delta x^2}$$

On obtient donc :

$$u_j^{i-1} \approx -\alpha u_{j-1}^i + (1 + 2\alpha)u_j^i - \alpha u_{j+1}^i \text{ avec } \alpha = \frac{\Delta\tau}{\Delta x^2}$$

Contrairement au schéma d'Euler explicite, la solution à l'instant $i\Delta\tau$ ne se fait pas directement, elle est implicite. En effet, le seul membre connu de cette équation est celui à l'instant $(i-1)\Delta\tau$ au point $j\Delta x$. Ici, le membre connu est donc combinaison linéaire des membres inconnus.

Sous forme matricielle, on obtient le schéma suivant :

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1+2\alpha & -\alpha & & 0 \\ 0 & -\alpha & \ddots & & \\ \vdots & \vdots & \ddots & & -\alpha \\ 0 & 0 & 0 & \dots & 1+2\alpha \end{pmatrix} \begin{pmatrix} u_{-N+1}^i \\ \vdots \\ u_0^i \\ \vdots \\ u_{N-1}^i \end{pmatrix} = \begin{pmatrix} u_{-N+1}^{i-1} \\ \vdots \\ u_0^{i-1} \\ \vdots \\ u_{N-1}^{i-1} \end{pmatrix} + \begin{pmatrix} \alpha u_{-N}^i \\ 0 \\ \vdots \\ 0 \\ \alpha u_N^i \end{pmatrix}$$

$$\begin{array}{cccc}
 \downarrow & \downarrow & \downarrow & \downarrow \\
 A & U_i & U_{i-1} & \alpha b_i
 \end{array}$$

On cherche donc à résoudre le système : $A U_i = U_{i-1} + \alpha b_i$.

La première manière à laquelle on pense pour réaliser cela est l'inversion de la matrice A qui est constante au fil des pas de temps. Cependant, la méthode de résolution la plus adaptée ici, en matière d'économie de coût de stockage, et étant donné que A est une matrice tridiagonale, est la décomposition LU.

On va donc décomposer A grâce à la décomposition LU afin d'obtenir une matrice triangulaire inférieure L et une matrice triangulaire supérieure U. On a :

$$\begin{pmatrix} 1+2\alpha & -\alpha & 0 & \dots & 0 \\ -\alpha & 1+2\alpha & -\alpha & & 0 \\ 0 & -\alpha & \ddots & & \\ \vdots & \vdots & \ddots & & -\alpha \\ 0 & 0 & 0 & \dots & 1+2\alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{-N+2} & 1 & \ddots & \vdots \\ 0 & l_{-N+3} & \ddots & \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & l_{N-1} & 1 \end{pmatrix} \begin{pmatrix} d_{-N+1} & u_{-N+1} & 0 & 0 \\ 0 & d_{-N+2} & u_{-N+2} & 0 \\ 0 & 0 & d_{-N+3} & \ddots \\ \vdots & \vdots & \ddots & u_{N-2} \\ 0 & 0 & \dots & d_{N-1} \end{pmatrix}$$

$$\begin{array}{ccc}
 \downarrow & & \downarrow \\
 L & & U
 \end{array}$$

$$= \begin{pmatrix} d_{-N+1} & u_{-N+1} & 0 & 0 & \dots & 0 \\ l_{-N+2}d_{-N+1} & l_{-N+2}u_{-N+1} + d_{-N+2} & u_{-N+2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

Tout d'abord on identifie les premiers termes :

$$u_j = -\alpha, \quad d_1 = 1 + 2\alpha, l_1 d_1 = -\alpha \text{ donc } l_1 = \frac{-\alpha}{1+2\alpha}$$

Ensuite, en regardant la matrice U :

$$d_{-N+1} = 1 + 2\alpha \text{ et } d_j = 1 + 2\alpha - \frac{\alpha^2}{d_{j-1}} \text{ avec } -N+2 \leq j \leq N-1$$

On regarde également la matrice L :

$$l_j = -\frac{\alpha}{d_j} \text{ avec } -N+1 \leq j \leq N-2$$

Le système : $AU_i = U_{i-1} + \alpha b_i$ équivalent à $LUU_i = U_{i-1} + \alpha b_i$ peut se réécrire :

$$\begin{cases} LX = U_{i-1} + \alpha b_i \\ X = U U_i \end{cases}$$

La 1^e équation équivaut à :

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ l_1 & 1 & & \vdots \\ \vdots & \ddots & \ddots & \\ 0 & 0 & l_{n-1} & 1 \end{pmatrix} \begin{pmatrix} x_{-N+1} \\ \vdots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} u_{-N+1}^{i-1} \\ \vdots \\ u_{N-1}^{i-1} \end{pmatrix} + \begin{pmatrix} \alpha u_{-N}^i \\ 0 \\ \vdots \\ \alpha u_N^i \end{pmatrix}$$

En résolvant ce système en partant de x_{-N+1} , on trouve :

$$x_{-N+1} = u_{-N+1}^{i-1} + \alpha u_{-N}^i$$

$$x_j = u_j^{i-1} - l_{j-1} x_{j-1} \text{ avec } -N+2 \leq j \leq N-2$$

$$x_{N-1} = u_{N-1}^{i-1} + \alpha u_N^i - l_{N-2} x_{N-2}$$

La 2^e équation équivaut à :

$$\begin{pmatrix} d_1 & -\alpha & 0 & \dots & 0 \\ 0 & d_2 & \ddots & & \vdots \\ \vdots & & \ddots & & -\alpha \\ 0 & \dots & 0 & d_n \end{pmatrix} \begin{pmatrix} u_{-N+1}^i \\ \vdots \\ u_{N-1}^i \end{pmatrix} = \begin{pmatrix} x_{-N+1} \\ \vdots \\ x_{N-1} \end{pmatrix}$$

En résolvant ce système en partant cette fois de x_{N-1} on trouve :

$$u_{N-1}^i = \frac{x_{N-1}}{d_{N-1}} \quad \text{et} \quad u_j^i = \frac{x_j + \alpha u_{j+1}^i}{d_j} \quad \text{avec} \quad -N+1 \leq j \leq N-2$$

Nous allons donc d'abord implémenter cette méthode implicite, reposant sur une factorisation « LU » à la main. Nous allons également tester une autre méthode consistante à résoudre directement le système matriciel avec les outils de résolution de Python et comparer nos résultats.

Résolution numérique de l'équation de Black-Scholes

Nous avons également réalisé une résolution directe de l'équation de Black-Scholes, sans passer par l'équation de la chaleur.

On rappelle l'équation de Black-Scholes pour un Call:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC = 0$$

La résolution directe de Black-Scholes se réalise de manière rétrograde puisque nous connaissons initialement la condition finale.

Le schéma explicite, dans ce cas, de Black-Scholes est donné par :

$$\frac{C_j^i - C_j^{i-1}}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{C_{j+1}^i - 2C_j^i + C_{j-1}^i}{\Delta S^2} + rS_j \frac{C_{j+1}^i - C_{j-1}^i}{2\Delta S} - rC_j^i = 0$$

Equivalent à :

$$C_j^{i-1} = C_j^i + \frac{1}{2}\sigma^2 S_j^2 \frac{\Delta t}{\Delta S^2} (C_{j+1}^i - 2C_j^i + C_{j-1}^i) + rS_j \frac{\Delta t}{2\Delta S} (C_{j+1}^i - C_{j-1}^i) - r\Delta t C_j^i$$

Le schéma est stable sous la condition :

$$\frac{\Delta t}{\Delta S^2} < \frac{1}{\sigma^2 S_{max}^2}$$

Nous allons donc implémenter directement ce schéma.

Le schéma implicite, lui, sera donné par :

$$\frac{C_j^{i+1} - C_j^i}{\Delta t} + \frac{1}{2}\sigma^2 S_j^2 \frac{C_{j+1}^i - 2C_j^i + C_{j-1}^i}{\Delta S^2} + rS_j \frac{C_{j+1}^i - C_{j-1}^i}{2\Delta S} - rC_j^i = 0$$

Equivalent à :

$$C_j^{i+1} = C_j^i - \frac{1}{2} \sigma^2 S_j^2 \frac{\Delta t}{\Delta S^2} (C_{j+1}^i - 2C_j^i + C_{j-1}^i) - r S_j \frac{\Delta t}{2\Delta S} (C_{j+1}^i - C_{j-1}^i) + r \Delta t C_j^i$$

Ou encore (avec $S_j = j\Delta S$) :

$$C_j^{i+1} = C_{j-1}^i \left(rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t \right) + C_j^i (1 + \sigma^2 j^2 \Delta t + r \Delta t) + C_{j+1}^i \left(-rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t \right)$$

Le terme connu (C_j^{i+1}) est donc combinaison linéaire des termes inconnus.

Sous forme matricielle, on a :

$$\begin{pmatrix} C_1^{i+1} \\ \vdots \\ C_j^{i+1} \\ \vdots \\ C_{nbS-2}^{i+1} \end{pmatrix} = \begin{pmatrix} 1 + \sigma^2 j^2 \Delta t + r \Delta t & -rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t & 0 \\ rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t & \ddots & \ddots \\ & \ddots & \ddots \end{pmatrix} \begin{pmatrix} C_1^i \\ \vdots \\ C_j^i \\ \vdots \\ C_{nbS-2}^i \end{pmatrix} + \begin{pmatrix} (rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t) C_0^{i+1} \\ 0 \\ \vdots \\ 0 \\ (-rj \frac{\Delta t}{2} - \frac{1}{2} \sigma^2 j^2 \Delta t) C_{nbS-1}^{i+1} \end{pmatrix}$$

Notre résolution consistera donc à résoudre ce système qui aura pour inconnu le vecteur des C^i .

Résultats

Comparaison des différents modèles

Nous avons donc créé deux codes Python. Dans le premier se trouve les méthodes d'Euler implicites et explicites en passant par l'équation de la chaleur. Dans le deuxième se trouve la résolution directe par l'équation de Black Scholes.

En passant par l'équation de la chaleur, nous avons fait deux types d'implémentation : l'une avec les solutions itératives en appliquant les formules décrites précédemment et l'autre en utilisant une approche matricielle. Nous voulions observer si les temps de calculs pouvaient varier selon le type de l'implémentation.

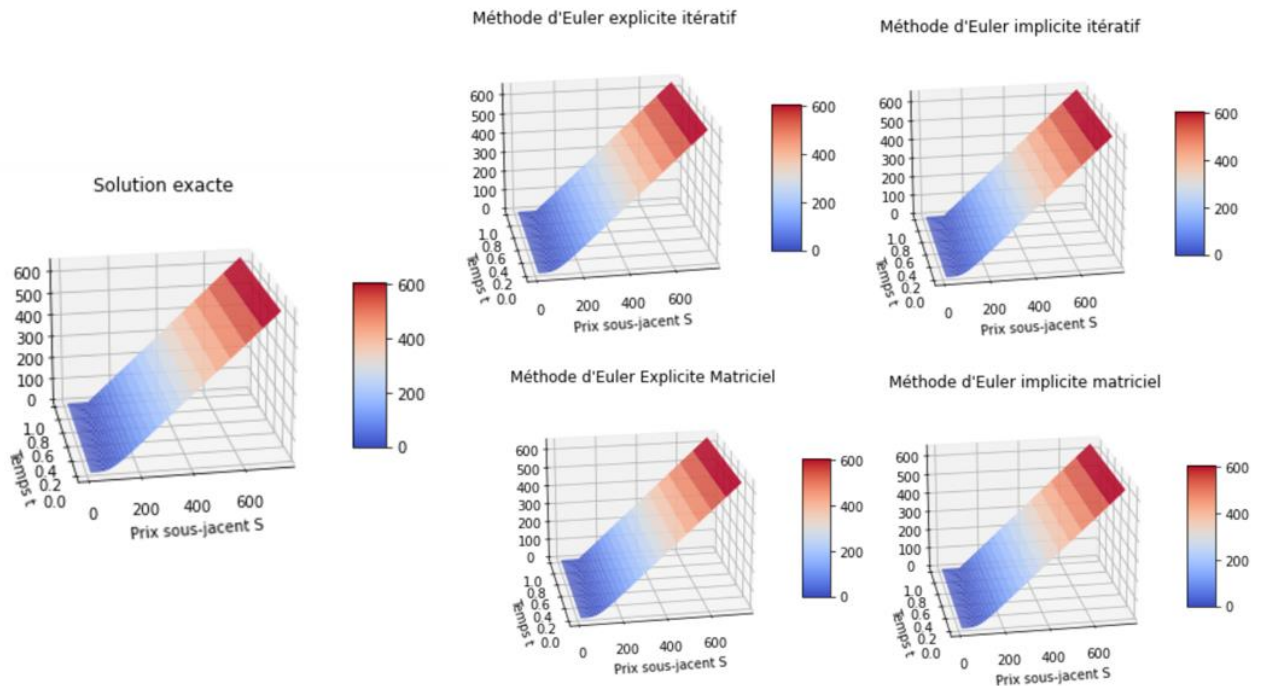
Nous avons aussi codé la solution analytique et nous avons comparé l'erreur de nos modèles.

- Résolution par l'équation de la chaleur

Nous avons fixé dans un premier temps comme paramètre :

T=1, K=100, $r=0.05$, $\sigma=0.4$, M=150, N=60 avec 150 points en temps et 60 points en espace et notre coefficient **$\alpha=0.48$** . Le prix maximal du sous-jacent est : 738 euros.

Voici les graphiques obtenus :



Comme on peut le voir, on retrouve bien la forme du tracé d'une option Call. De plus, avec un $\alpha < 0,5$, les deux méthodes sont fonctionnelles. Voici les erreurs des différentes méthodes avec ces paramètres :

```
Point en espace : 60
Point en Temps : 150
Coef Euler : 0.4800000000000001
erreur méthode Euler explicite Itératif: 1.013874917631165
erreur méthode Euler explicite Matriciel: 5.607501802972835
erreur méthode Euler implicite LU : 1.6711632798632485
erreur méthode Euler implicite Matriciel: 1.671163279863451

Nombre de points : 9000
Temps d'exécution pour la méthode Euler Explicite Itératif : 0.109375
Temps d'exécution pour la méthode Euler Explicite Matriciel : 0.015625
Temps d'exécution pour la méthode Euler Implicite Itératif : 0.203125
Temps d'exécution pour la méthode Euler Implicite Matriciel : 0.0625
```

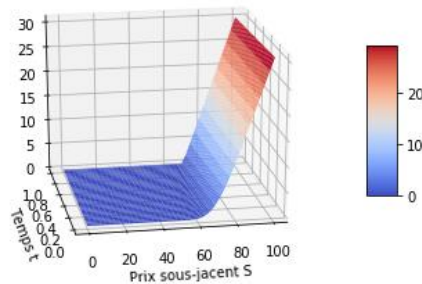
Nous pouvons voir que nous obtenus des résultats plutôt satisfaisant avec une erreur inférieure à 6€ pour tous nos modèles. Ici, notre modèle le plus précis est Euler explicite implémenté itérativement mais nous pouvons voir que les méthodes implicites possèdent une erreur similaire à 60 centimes près.

- Résolution directe de l'équation de Black-Scholes

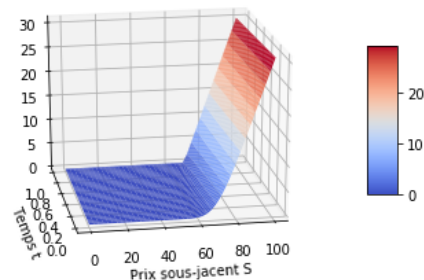
Pour cette résolution, nous avons fixé dans un premier temps le nombre de points en espace **nbS=40** et le nombre de points en temps **nbT=70**, **K=70**, **T=1**, **r=0.01** et **$\sigma=0.1$** . Le prix maximal du sous-jacent est 100.

Les graphiques obtenus sont :

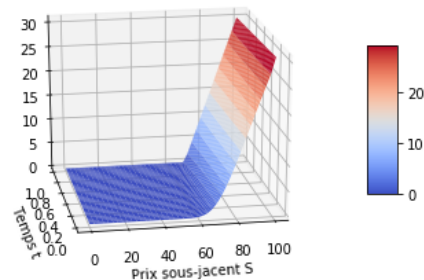
Evolution du prix de l'option Call en fonction du prix de l'actif sous-jacent et du temps (Solution exacte)



Evolution du prix de l'option Call en fonction du prix de l'actif sous-jacent et du temps (Méthode d'Euler explicite)



Evolution du prix de l'option Call en fonction du prix de l'actif sous-jacent et du temps (Méthode d'Euler implicite)



Voici les erreurs ainsi que les temps d'exécution obtenus :

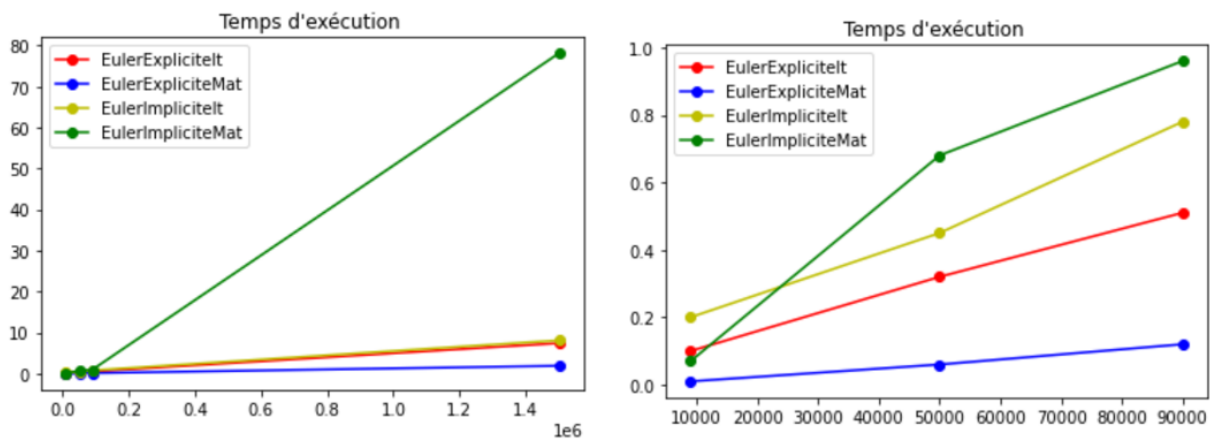
```
Nombre de points en temps: 70
Nombre de points en espace: 40
Temps d'exécution - méthode d'Euler explicite: 0.01196908950805664
Temps d'exécution - méthode d'Euler implicite: 0.005957603454589844
Erreur méthode Euler explicite: 0.2246186913076396
Erreur méthode Euler implicite: 5.530731741500949
```

Comme nous pouvons le voir, ici encore, nous obtenons une erreur inférieure à 6€ pour tous nos modèles. Ici, notre modèle le plus précis est Euler explicite qui est bien plus précis que la méthode d'Euler implicite.

Influence du nombre de points sur le temps d'exécution

- Résolution par l'équation de la chaleur

Nous avons voulu regarder le temps de calcul de nos algorithmes selon le nombre de points que nous prenions. Voici deux graphiques dont l'un pour lequel on a rajouté une valeur correspondant à 1 500 000 points :



Comme on peut le voir, le temps d'exécution de notre méthode d'Euler Implicite Matricielle explose contrairement aux autres méthodes qui semblent, elles, augmenter de 10 lorsque nous avons 10 fois plus de points. De plus, on peut observer que, mis à part la méthode implicite matricielle qui explose, les méthodes matricielles sont globalement plus rapides que les méthodes itératives.

Concernant la méthode d'Euler implicite matricielle, nous avons choisi de se servir de la fonction `np.linalg.solve(A,B)` qui permet de calculer l'inverse d'une matrice. Le problème est que le calcul d'inversion de matrice est très lourd et très coûteux donc les temps de calculs explosent assez rapidement dès que nous mettons un nombre conséquent de points.

On remarque, de plus, que le modèle d'Euler explicite matriciel est notre modèle le plus efficace en termes de rapidité mais c'est aussi le modèle ayant une précision légèrement inférieure aux autres. On détaillera les précisions par la suite dans le rapport.

- Résolution directe de l'équation de Black-Scholes

Pour comparer nos temps d'exécution, nous avons repris les mêmes paramètres que décrits dans la section précédente. Nous avons fait varier le nombre de points en le multipliant par 10 à chaque étape.

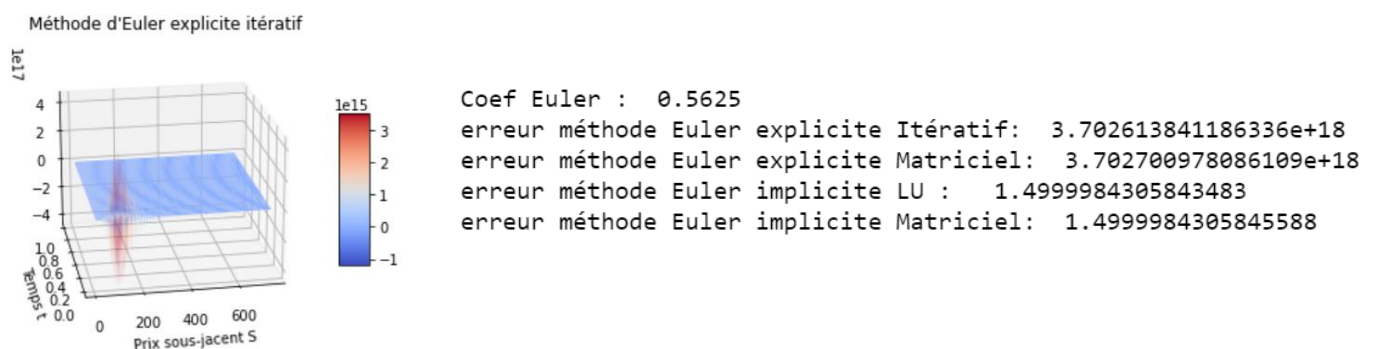
Résolution explicite	Résolution implicite
nbS:40, nbT:70 ->2800 points Temps_e=0.0119	nbS:40, nbT:70 ->2800 points Temps_i=0.0059
nbS:40, nbT:700 ->28000 points Temps_e=0.0916	nbS:40, nbT:700 ->28000 points Temps_i=0.0328
nbS:40, nbT:7000 ->280000 points Temps_e=0.9961	nbS:40, nbT:7000 ->280000 points Temps_i=0.3044

On peut observer qu'en multipliant par 10 le nombre de points la première fois, le temps d'exécution de la méthode explicite est multiplié par 7. Celui de la méthode implicite, lui est multiplié par 5. Lorsque l'on remultiplie par 10 le nombre de points, les temps sont tous deux multipliés par 10. Par ailleurs, on peut observer que la résolution implicite de l'équation est 2 à 3 fois plus rapide que la résolution explicite. Ceci s'explique par le fait que nous avons dans notre code pour la résolution explicite une boucle en espace en plus que nous devons appliquer à chaque itération de temps alors que dans notre modèle implicite, nous n'avons qu'une boucle en temps.

Influence de la variation du nombre de points en temps et en espace sur la précision du modèle

- Résolution par l'équation de la chaleur

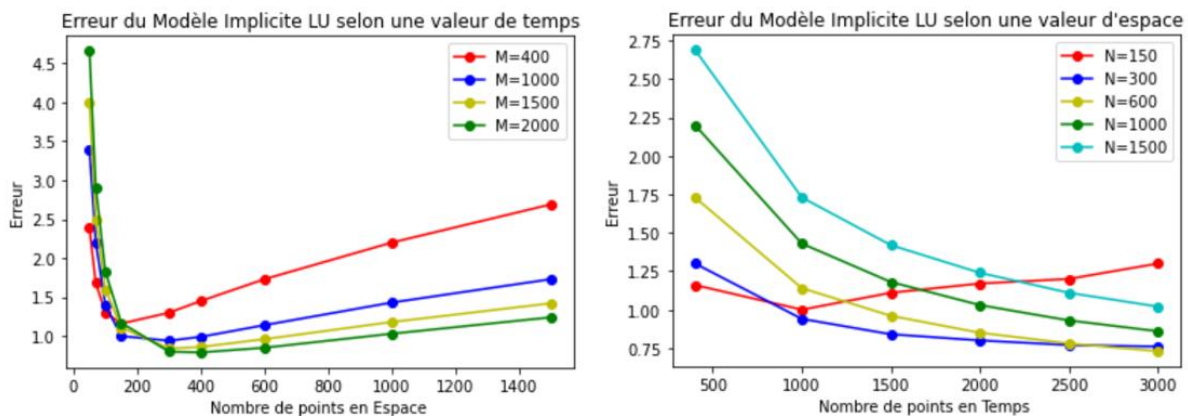
Pour la méthode d'Euler Explicite, la contrainte d'avoir un $\alpha < 0,5$ est primordiale car elle permet de conserver la stabilité du modèle. En effet, la solution explosera le cas échéant :



Concernant le modèle d'Euler Explicite, la méthode itérative donne des résultats plus précis que la méthode matricielle mais cette dernière s'exécute légèrement plus rapidement. Le temps de calcul de la méthode n'étant pas non plus si éloigné de la méthode matricielle, nous pensons que dans le cas où nous pouvons satisfaire les conditions de stabilité, la méthode itérative serait à privilégier puisqu'il génère la plus petite erreur.

Cependant, dans de nombreux cas nous ne serons pas en possibilité de respecter ces conditions et nous aurons donc tout intérêt à nous servir de la méthode d'Euler Implicite. Dans ce cas-là, les erreurs de nos deux modèles sont identiques par rapport à la solution analytique. Cependant, nous rencontrons un problème avec la méthode matricielle qui nécessite de calculer l'inverse d'une matrice, faisant exploser les temps de calculs plus le nombre de points total est grand. Notre méthode itérative avec décomposition LU est donc notre modèle le plus efficace puisque ce modèle offre d'excellents résultats avec un temps de calcul très raisonnable sans avoir à respecter de conditions particulières.

Nous allons donc, pour la suite, nous concentrer sur la méthode d'Euler Implicite avec décomposition LU. Notre objectif était de déterminer le nombre de points à choisir afin de minimiser l'erreur de notre modèle. Pour cela nous avons modélisé l'évolution de l'erreur, d'une part en fixant le nombre de points en temps et en faisant varier le nombre de points en espace (*graphique de gauche*) puis en fixant le nombre de points en espace et en faisant varier le nombre de points en temps (*graphique de droite*).



Nous nous sommes rendus compte, en analysant ces graphiques, que l'erreur minimale était atteinte lorsque : $\lim_{M \rightarrow \infty} \frac{M}{N} = 5$.

En effet, pour M=400,1000,1500,2000 nous avons obtenu notre erreur minimale avec N=150,300,300,400 soit l'équivalent sous forme de ratio : $\frac{M}{N} = 2,66; 3,33; 5; 5$.

Si nous souhaitons donc obtenir une erreur de plus en plus petite, on doit augmenter le nombre de points en gardant en tête ce ratio d'équilibre, il faut 5 fois plus de points en temps qu'en espace.

- Résolution directe de l'équation de Black-Scholes

Nous avons comparé les erreurs trouvées pour les méthodes explicites et implicites en faisant varier le nombre de points en temps et en espace en gardant, dans un premier temps les paramètres énoncés précédemment et rappelés dans le tableau ci-dessous.

Résolution explicite r = 0.01 # rendement sigma = 0.1 # volatilité	Résolution implicite
nbS:40, nbT:15, alpha=0.0108 > borne=0.0099 E_e=0.16	nbS:40, nbT:15, alpha=0.0108 > borne=0.0099 E_i=2.15
nbS:40, nbT:70, alpha=0.0022 < borne=0.0099 E_e= 0.22	nbS:40, nbT:70, alpha=0.0022 < borne=0.0099 E_i=5.53
nbS:40, nbT:350, alpha=0.00043 < borne=0.0099 E_e=0.60	nbS:40, nbT:350, alpha=0.00043 < borne=0.0099 E_i=12.43
nbS:40, nbT:700, alpha=0.000217 < borne=0.0099 E_e= 0.87	nbS:40, nbT:700, alpha=0.000217 < borne=0.0099 E_i= 17.59

On peut observer que lorsque l'on augmente le nombre de points en temps, les erreurs pour les deux méthodes vont augmenter. Par ailleurs lorsque l'on augmente le nombre de points en espace, les erreurs vont diminuer mis à part pour la méthode explicite qui, au franchissement de la CFL, va diverger. Nous n'avons pas réalisé des tests aussi poussés pour cette résolution que pour la résolution par l'équation de la chaleur. Cependant, au vu des évolutions des erreurs nous pensons qu'un ratio nombre de points en temps/ nombre de points en espace doit également être respecté ici.

Résolution explicite r = 0.01 # rendement sigma = 0.1 # volatilité	Résolution implicite
nbS:40, nbT:70, alpha=0.0022 < borne=0.0099 E_e= 0.22	nbS:40, nbT:70, alpha=0.0022 < borne=0.0099 E_i=5.53
nbS:400, nbT:70, alpha=0.230 > borne=0.0099 E_e=4.12e+103	nbS:400, nbT:70, alpha=0.230 > borne=0.0099 E_i=2.19
nbS:1000, nbT:70, alpha=1.446 > borne=0.0099 E_e=inf	nbS:1000, nbT:70, alpha=1.446 > borne=0.0099 E_i=1.47

Enfin, on peut observer que si l'on augmente les valeurs du rendement et de la volatilité, les erreurs seront plus grandes. Notamment, on peut voir que l'erreur de la méthode implicite est bien trop grande pour que le résultat soit exploitable. Nous devons donc avoir une erreur dans notre code. Tout de même, si la volatilité augmente, le risque devient plus grand et donc le spectre de valeurs pris par l'option est plus grand ce qui peut expliquer que nos modèles génèrent plus d'erreur.

Résolution explicite r = 0.4 # rendement sigma = 0.2 # volatilité	Résolution implicite
nbS:40, nbT:70, alpha=0.0022 < borne=0.00249 Erreur_e= 1.314	nbS:40, nbT:70, alpha=0.0022 < borne=0.00249 Erreur_i=147.22
nbS:40, nbT:350, alpha=0.00043 < borne=0.00249 Erreur_e=1.542	nbS:40, nbT:350, alpha=0.0022 < borne=0.00249 Erreur_i=329.30
nbS:40, nbT:700, alpha=0.000217 < borne=0.00249 Erreur_e= 2.32	nbS:40, nbT:700, alpha=0.000217 < borne=0.00249 Erreur_i= 465.73

Conclusion

L'objet de ce projet était de présenter différents modèles mathématiques permettant la résolution de l'équation de Black Scholes pour le pricing d'option puis de les implémenter.

Cette étude nous a montré qu'il était important d'effectuer un travail en amont sur papier afin de déterminer s'il est possible de transformer le schéma initial afin d'obtenir de meilleurs résultats, notamment sur le temps de calcul comme la décomposition LU d'Euler Implicite.

Nous avons aussi grâce à ce projet pu renforcer davantage nos connaissances dans ce domaine.