

# Rapport du Projet Mathématique

10 Mai 2022

**Adib HABBOU**  
**Adel KEBLI**  
**Rabab KHATIB**

**Parte 2 :**  
**Modèle de Black-Scholes**  
**Convergence des prix**  
**EDP de Black-Scholes**

# Table des matières

<b>1</b>	<b>Modèle de Cox-Ross-Rubinstein</b>	<b>3</b>
<b>2</b>	<b>Modèle de Black-Scholes</b>	<b>3</b>
2.1	Question 11 - Formule de $S_t$ . . . . .	3
2.2	Question 12 - Fonction pricer_MC . . . . .	3
2.3	Question 13 - Graphique de $Prix_{MC}^{(n)}$ . . . . .	4
2.4	Question 14 - Convergence presque-sûre de $(Prix_{MC}^{(n)})_{n \in \mathbb{N}}$ . . . . .	4
2.5	Question 15 - Fonction put_BS . . . . .	5
2.6	Question 16 - Test put_BS . . . . .	5
2.7	Question 17 - Comparaison pricer_MC et put_BS . . . . .	5
2.8	Question 18 - Graphique 3D de put_BS . . . . .	6
2.9	Question 19 - Comparaison pricer_2 et put_BS . . . . .	7
2.10	Question 20 - Comparaison des méthodes d'approche numérique de l'EDP de Black-Scholes . . . . .	7
2.10.1	Différences finies explicites . . . . .	8
2.10.2	Différences finies implicites . . . . .	9
2.10.3	Méthode de Crank-Nicolson . . . . .	10
2.10.4	Erreurs relatives . . . . .	10
2.10.5	Comparaison et conclusions . . . . .	11
<b>3</b>	<b>Conclusion</b>	<b>11</b>

## 1 Modèle de Cox-Ross-Rubinstein

## 2 Modèle de Black-Scholes

### 2.1 Question 11 - Formule de $S_t$

La fonction logarithme est de classe  $C^2$ , donc on peut appliquer la formule d'Itô à  $\log(S_T)$ .

$$\begin{aligned} d\log(S_t) &= \log'(S_t)dS_t + \frac{|\sigma S_t|^2}{2}\log''(S_t)dt \\ d\log(S_t) &= \frac{dS_t}{S_t} - \frac{|\sigma S_t|^2}{2} \frac{dt}{S_t^2} \end{aligned}$$

Comme,  $dS_t = S_t(r dt + \sigma dB_t)$ , on peut remplacer  $dS_t$  dans l'expression.

$$\begin{aligned} d\log(S_t) &= r dt + \sigma dB_t - \frac{|\sigma|^2}{2} dt \\ \frac{d}{dt}\log(S_t) &= r + \sigma \frac{d}{dt}B_t - \frac{|\sigma|^2}{2} \end{aligned}$$

On intègre sur  $t$  des deux côtés entre 0 et  $+\infty$  :

$$\log(S_t) = r t + \sigma B_t - \frac{|\sigma|^2}{2} t + cte$$

Pour  $t=0$ , nous avons  $S_0 = s$  et  $\log(S_0) = \sigma B_0 + cte$ , donc  $cte = \log(s) - \sigma B_0$ . On remplace :

$$\log(S_t) = r t + \sigma B_t - \frac{|\sigma|^2}{2} t + \log(S_0) - \sigma B_0$$

Par passage à l'exponentielle, nous obtenons :

$$S_t = s \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \sigma(B_t - B_0)\right)$$

Sachant  $B_0 = 0$ ,

$$S_t = s \exp\left(\left(r - \frac{\sigma^2}{2}\right)t + \sigma B_t\right)$$

### 2.2 Question 12 - Fonction pricer\_MC

Nous stockons les éléments de la suite de variables aléatoires  $(\xi_i)_{1 \leq i \leq n}$  dans une `array list` en utilisant la fonction `random.normal` de la librairie `numpy`.

Sachant que les valeurs appliquées peuvent être très grandes, nous utilisons les fonctions mathématiques de la librairie `mpmath` pour gérer ces ordres de grandeur.

### 2.3 Question 13 - Graphique de $Prix_{MC}^{(n)}$

On code la fonction  $f : x \mapsto \max(x - 110, 0)$ .

Comme  $n = 10^5 k$  avec  $1 \leq k \leq 10$ , nous créons une liste **N** qui contient l'ensemble des valeurs prises par  $n$ . Puis, pour chacune de ces valeurs, nous ajoutons à une liste **price\_MC** le  $Prix_{MC}^{(n)}$  correspondant, pour  $r=0.01$ ,  $\sigma=0.1$ ,  $s=100$  et  $T=1$ .

Enfin, grâce à la librairie `matplotlib.pyplot`, nous affichons le graphique de la variation du pricer selon le modèle de Monte-Carlo.

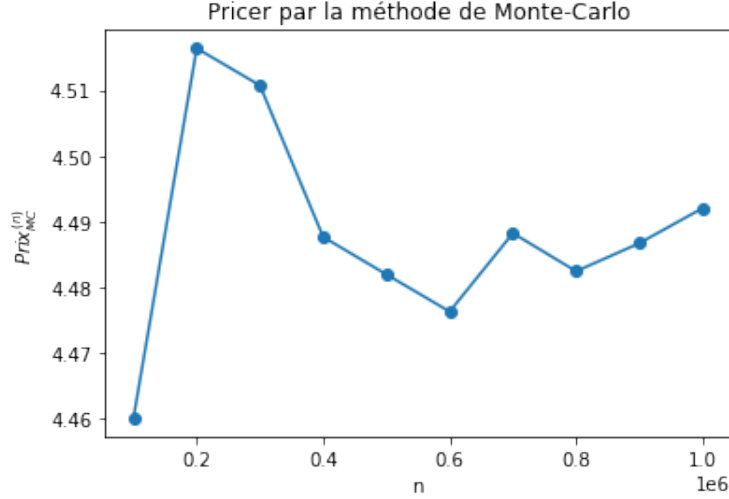


FIGURE 1 – Pricer par le modèle de Monte-Carlo

### 2.4 Question 14 - Convergence presque-sûre de $(Prix_{MC}^{(n)})_{n \in \mathbb{N}}$

On a  $P = \mathbb{E}[e^{-rT} f(S_T)]$ .

Donc,  $P = \mathbb{E}[e^{-rT} f(s \exp((r - \frac{\sigma^2}{2})T + \sigma B_T))]$ .

Or,  $\forall t \geq 0, B_t - B_s \sim \mathcal{N}(0, t - s)$ ,

donc,  $B_t = B_t - B_0 \sim \mathcal{N}(0, t)$ .

On a alors, d'après l'estimation analytique de l'espérance d'une loi normale,

$$\mathbb{E}[e^{-rT} f(s \exp((r - \frac{\sigma^2}{2})T + \sigma B_T))] = \frac{1}{n} \sum_{i=1}^n e^{-rt_i} f(s \exp((r - \frac{\sigma^2}{2})t_i + \sigma B_{t_i}))$$

Donc,

$$P = \frac{1}{n} \sum_{i=1}^n e^{-rt_i} f(s \exp((r - \frac{\sigma^2}{2})t_i + \sigma B_{t_i}))$$

Or, nous avons  $\xi_i \sim \mathcal{N}(0, 1)$ , donc on en déduit  $\sqrt{T}\xi_i \sim \mathcal{N}(0, T)$ .

Nous pouvons en conclure que :

$$\boxed{Prix_{MC}^{(n)} \xrightarrow[n \rightarrow +\infty]{p.s.} P}$$

Donc, la suite  $(Prix_{MC}^{(n)})_{n \in \mathbb{N}}$  converge presque sûrement vers  $P$ .

## 2.5 Question 15 - Fonction put\_BS

Nous implémentons d'abord `F`, la fonction de répartition de la loi normale centrée réduite, qui renvoie pour tout  $x \in \mathbb{R}$ ,  $\mathbb{P}(Y \leq x)$  où  $Y \sim \mathcal{N}(0, 1)$ . Pour ceci, nous nous servons de la fonction `norm.cdf` de la librairie `scipy.stats`. Nous ne précisons pas les valeurs de l'espérance et de l'écart-type, car par défaut on a  $\mu=0$  et  $\sigma=1$ . Dans la fonction `put_BS`, nous commençons par calculer  $d1$  et  $d2$ , pour ensuite appliquer la formule de  $Prix_{BS}$ .

## 2.6 Question 16 - Test put\_BS

Pour  $r=0.01$ ,  $\sigma=0.1$ ,  $s=100$ ,  $T=1$  et  $K=90$ , la fonction `put_BS` renvoie :

$$Prix_{BS} = 0.5815000751362422 \text{ €}$$

## 2.7 Question 17 - Comparaison pricer\_MC et put\_BS

On code la fonction `f : x ↦ max(90 - x, 0)`.

Pour chacune des valeurs prises par  $n$ , nous ajoutons à une nouvelle liste `price_MC` le  $Prix_{MC}^{(n)}$  correspondant pour cette nouvelle fonction `f`, pour  $r=0.01$ ,  $\sigma=0.1$ ,  $s=100$  et  $T=1$ . Puis, nous superposons au graphe de la variation de  $Prix_{MC}^{(n)}$  la droite de  $Prix_{BS}$ .

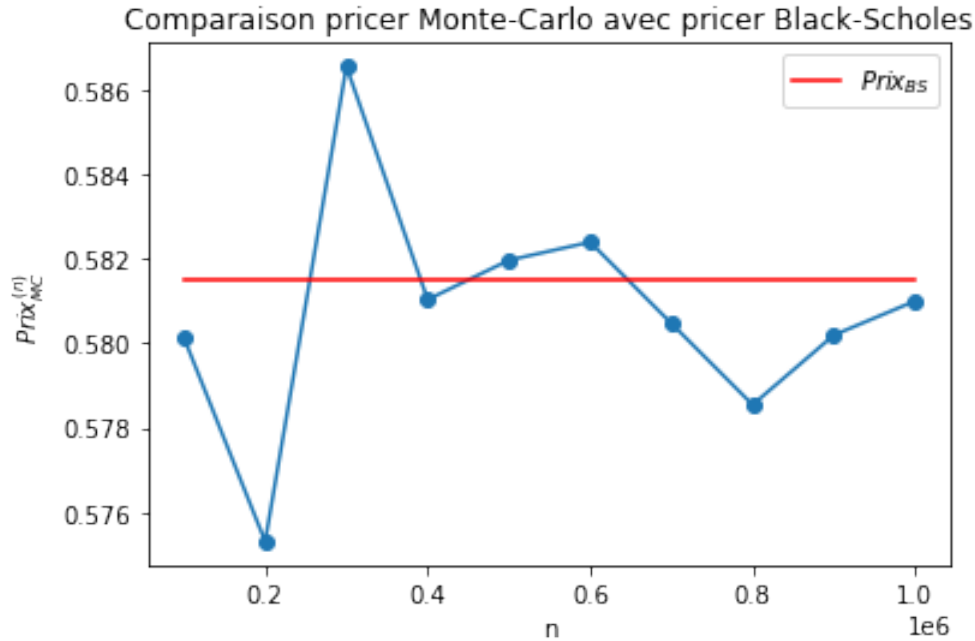


FIGURE 2 – Comparaison entre le pricer par le modèle de Monte-Carlo avec le pricer par la formule de Black-Scholes

Nous remarquons que plus la valeur de  $n$  est grande, plus  $Prix_{MC}^{(n)}$  s'approche du vrai prix trouvé par la formule de Black-Scholes : à terme, le modèle va converger vers le prix réel. On peut donc considérer que la méthode de Monte-Carlo n'est pas assez précise pour des valeurs de l'ordre de  $n = 10^5$ , mais qu'elle se rapproche du vrai prix pour des valeurs de l'ordre de  $10^6$ .

## 2.8 Question 18 - Graphique 3D de put\_BS

On trace sur un graphique en 3 dimensions le prix réel de l'option donné par la fonction `put_BS` en faisant varier à la fois  $s$ , la valeur initiale à  $t = 0$ , et  $T$ . Pour ce faire, on utilise la fonction `mplot3d` de la librairie `mpl_toolkits`. Quant à elle, la fonction `meshgrid` de la librairie `numpy` renvoie une matrice de coordonnées à partir des vecteurs  $s$  et  $t$ , nous permettant de faire varier à la fois le paramètre  $s$  et  $T$  pour la fonction `put_BS`. Nous avons choisi de superposer à notre surface une représentation en *wireframe* pour pouvoir mieux visualiser le tracé et identifier chaque droite qui le compose.

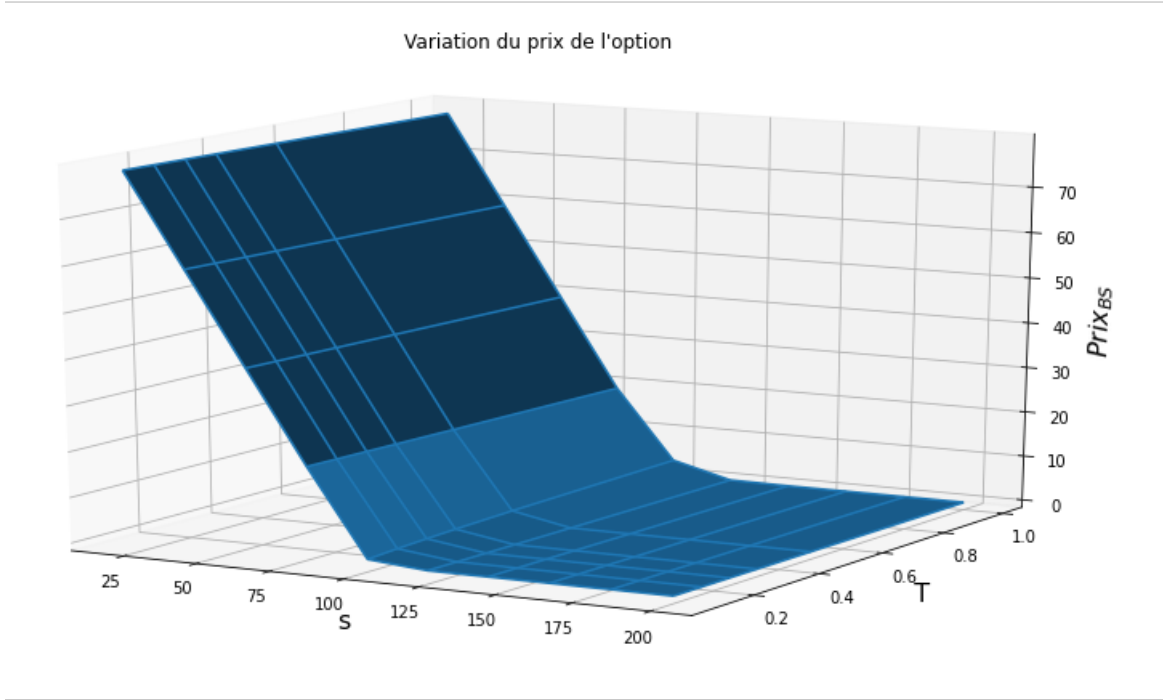


FIGURE 3 – Variation du prix de l'option en fonction du prix initial et du temps

Nous remarquons que  $Prix_{BS}$  varie uniquement en fonction de la valeur de  $s$ , sans tenir compte de  $T$  : le pricer de Black-Scholes dépend uniquement du prix initial de l'option, l'instant auquel on choisit de l'observer n'ayant aucune influence. De plus,  $Prix_{BS}$  décroît linéairement pour  $s$  compris entre 20 et 100, pour ensuite s'annuler au delà d'un certain seuil.

Ce seuil équivaut au **strike price** de notre option : lorsque le prix initial de l'option devient supérieur ou égal au **strike price**, noté ici  $K$ , l'achat de l'option perd tout son intérêt. Dans notre modèle, l'option n'a plus de valeur et donc son prix s'annule.

## 2.9 Question 19 - Comparaison pricer\_2 et put\_BS

On code la fonction  $f : x \mapsto \max(100 - x, 0)$ .

Nous avons repris la fonction `pricer_2` de la question 5 et nous avons superposé sa représentation à la droite du vrai prix obtenu par la fonction `put_BS`.

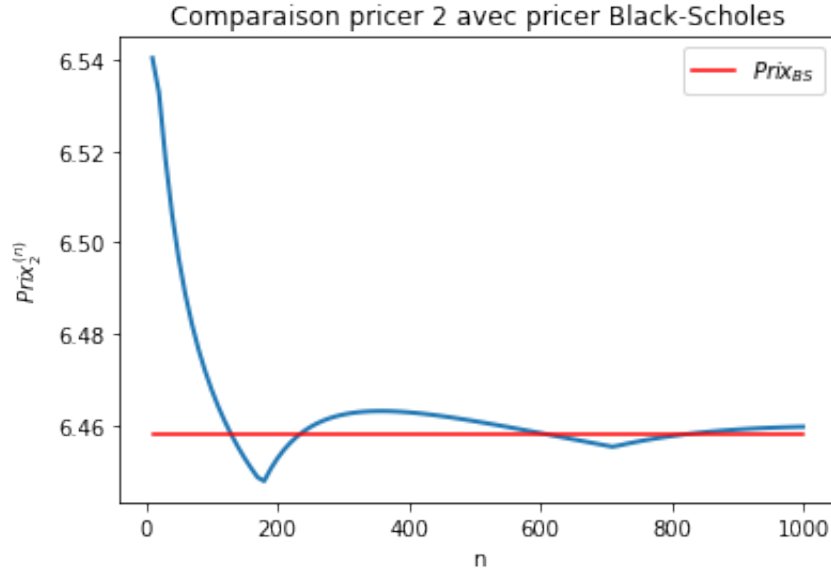


FIGURE 4 – Comparaison du `pricer_2` avec celui de Black-Scholes

On remarque que le prix obtenu par la fonction `pricer_2` converge vers le vrai prix, obtenu par le pricer de Black-Scholes. Comme le deuxième pricer correspond au modèle de Cox-Ross-Rubinstein, on en déduit que pour des très grandes valeurs de  $n$ , de l'ordre de  $10^3$ , le modèle binomial converge vers celui de Black-Scholes. On peut donc dire que le prix d'une option dans le modèle CRR converge vers le prix dans le modèle BS.

## 2.10 Question 20 - Comparaison des méthodes d'approche numérique de l'EDP de Black-Scholes

Une approximation du prix du put est donnée par une fonction  $p(t, x)$  qui dépend du temps jusqu'à maturité  $t$  et du log prix de l'actif risqué  $x = \log(S)$ . Elle vérifie l'équation aux dérivées partielles suivante :

$$\forall (t, x) \in (0, T] \times (x_{\min}, x_{\max}), \quad \frac{\partial p}{\partial t}(t, x) - \frac{1}{2}\sigma^2 \frac{\partial^2 p}{\partial t^2}(t, x) - \left(r - \frac{\sigma^2}{2}\right) \frac{\partial p}{\partial x}(t, x) + rp(t, x) = 0$$

La condition initiale est le payoff, qui correspond à la valeur du prix pour un temps jusqu'à maturité de 0. Nous définissons ainsi les conditions aux limites :

$$\begin{cases} \forall x \in [x_{\min}, x_{\max}], p(0, x) = \max(K - e^x, 0) \\ \forall t \in [0, T], p(t, x_{\min}) = Ke^{-rt} - e^{x_{\min}} \\ \forall t \in [0, T], p(t, x_{\max}) = 0 \end{cases}$$

### 2.10.1 Différences finies explicites

Adoptons la notation  $p(t_m, x_j) = p_{m,j}$ . On a donc le schéma aux différences finies suivant :

$$\begin{cases} \frac{\partial p}{\partial t}(t_m, x_j) \approx \frac{1}{\Delta t}(p(t_{m+1}, x_j) - p(t_m, x_j)) \\ \frac{\partial p}{\partial x}(t_m, x_j) \approx \frac{1}{2h}(p(t_m, x_{j+1}) - p(t_m, x_{j-1})) \\ \frac{\partial^2 p}{\partial x^2}(t_m, x_j) \approx \frac{1}{h^2}(p(t_m, x_{j+1}) - 2p(t_m, x_j) + p(t_m, x_{j-1})) \end{cases}$$

En utilisant ce schéma, l'équation de Black-Scholes donne alors :

$$\frac{1}{\Delta t} [p_{m+1,j} - p_{m,j}] - \frac{\sigma^2}{2h^2} [p_{m,j+1} - 2p_{m,j} + p_{m,j-1}] - \frac{1}{2h} \left( r - \frac{\sigma^2}{2} \right) [p_{m,j+1} - p_{m,j-1}] + r p_{m,j} = 0$$

$$\frac{1}{\Delta t} p_{m+1,j} + \left[ r + \frac{\sigma^2}{h^2} - \frac{1}{\Delta t} \right] p_{m,j} + \left[ \frac{1}{2h} \left( r - \frac{\sigma^2}{2} \right) - \frac{\sigma^2}{2h^2} \right] p_{m,j-1} - \left[ \frac{\sigma^2}{2h^2} + \frac{1}{2h} \left( r - \frac{\sigma^2}{2} \right) \right] p_{m,j+1} = 0$$

$$p_{m+1,j} = \Delta t \left[ \left( \frac{1}{\Delta t} - r - \frac{\sigma^2}{h^2} \right) p_{m,j} + \left( \frac{\sigma^2}{2h^2} - \frac{r}{2h} + \frac{\sigma^2}{4h} \right) p_{m,j-1} + \left( \frac{r}{2h} - \frac{\sigma^2}{4h} + \frac{\sigma^2}{2h^2} \right) p_{m,j+1} \right]$$

$$p_{m+1,j} = \Delta t \left[ \left( \frac{1}{\Delta t} - r - \frac{\sigma^2}{h^2} \right) p_{m,j} + \left( \frac{\sigma^2}{2h^2} + \frac{\sigma^2 - 2r}{4h} \right) p_{m,j-1} + \left( \frac{2r - \sigma^2}{4h} + \frac{\sigma^2}{2h^2} \right) p_{m,j+1} \right]$$

Simplifions l'écriture en posant les coefficients respectifs pour  $p_{m,j-1}$ ,  $p_{m,j}$  et  $p_{m,j+1}$  :

$$\begin{cases} c_{j,0} = \frac{\sigma^2}{2h^2} + \frac{\sigma^2 - 2r}{4h} \\ c_{j,1} = \frac{1}{\Delta t} - r - \frac{\sigma^2}{h^2} \\ c_{j,2} = \frac{2r - \sigma^2}{4h} + \frac{\sigma^2}{2h^2} \end{cases}$$

Nous obtenons donc l'équation suivante :

$$p_{m+1,j} = \Delta t (c_{j,1} p_{m,j} + c_{j,0} p_{m,j-1} + c_{j,2} p_{m,j+1})$$

Passons en calcul matriciel :  $P_{m+1} = A \times P_m$

$$\text{en posant la matrice des coefficients } A = \Delta t \begin{pmatrix} c_{j,1} & c_{j,2} & 0 & \dots & 0 \\ c_{j,0} & c_{j,1} & c_{j,2} & \ddots & \vdots \\ 0 & c_{j,0} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{j,2} \\ 0 & \dots & 0 & c_{j,0} & c_{j,1} \end{pmatrix}$$

Nous itérons les calculs directement sur le vecteur  $P_m$  mais nous prenons soin d'implémenter deux fonctions `pmin` et `pmax` pour calculer les conditions aux limites `xmin` et `max`, nous remplissons avec les éléments  $p_{m,1}$  et  $p_{m,N-1}$  de chaque vecteur  $P_m$ .



### 2.10.2 Différences finies implicites

Pour la méthode des différences finies implicite, on reprend le schéma explicite et on substitue dans les dérivés partielles en  $x$  les valeurs en  $t_m$  avec celles en  $t_{m+1}$  : ainsi nous exprimons le prix à l'instant  $m$  en fonction de celui à l'instant  $m+1$ . Par cette démarche, nous obtenons le schéma suivant :

$$\begin{cases} \frac{\partial p}{\partial t}(t_m, x_j) \approx \frac{1}{\Delta t}(p(t_{m+1}, x_j) - p(t_m, x_j)) \\ \frac{\partial p}{\partial x}(t_m, x_j) \approx \frac{1}{2h}(p(t_{m+1}, x_{j+1}) - p(t_{m+1}, x_{j-1})) \\ \frac{\partial^2 p}{\partial x^2}(t_m, x_j) \approx \frac{1}{h^2}(p(t_{m+1}, x_{j+1}) - 2p(t_{m+1}, x_j) + p(t_{m+1}, x_{j-1})) \end{cases}$$

En utilisant ce schéma, l'équation de Black-Scholes donne alors :

$$\frac{1}{\Delta t} [p_{m+1,j} - p_{m,j}] - \frac{\sigma^2}{2h^2} [p_{m+1,j+1} - 2p_{m+1,j} + p_{m+1,j-1}] - \frac{1}{2h} \left( r - \frac{\sigma^2}{2} \right) [p_{m+1,j+1} - p_{m+1,j-1}] + r p_{m+1,j} = 0$$

$$\left( \frac{1}{\Delta t} + r + \frac{\sigma^2}{h^2} \right) p_{m+1,j} - \frac{1}{\Delta t} p_{m,j} - \left( \frac{\sigma^2}{2h^2} + \frac{r}{2h} - \frac{\sigma^2}{4h} \right) p_{m+1,j+1} - \left( \frac{\sigma^2}{2h^2} + \frac{\sigma^2}{4h} - \frac{r}{2h} \right) p_{m+1,j-1} = 0$$

$$\left( \frac{1}{\Delta t} + r + \frac{\sigma^2}{h^2} \right) p_{m+1,j} + \left( \frac{\sigma^2 - 2r}{4h} - \frac{\sigma^2}{2h^2} \right) p_{m+1,j+1} + \left( \frac{2r - \sigma^2}{4h} - \frac{\sigma^2}{2h^2} \right) p_{m+1,j-1} = \frac{1}{\Delta t} p_{m,j}$$

Simplifions l'écriture en posant les coefficients respectifs pour  $p_{m+1,j-1}$ ,  $p_{m+1,j}$  et  $p_{m+1,j+1}$  :

$$\begin{cases} b_{j,0} = \frac{2r - \sigma^2}{4h} - \frac{\sigma^2}{2h^2} \\ b_{j,1} = \frac{1}{\Delta t} + r + \frac{\sigma^2}{h^2} \\ b_{j,2} = \frac{\sigma^2 - 2r}{4h} - \frac{\sigma^2}{2h^2} \end{cases}$$

En calcul matriciel, nous obtenons :  $D \times P_m = C \times P_{m+1}$

$$\text{avec } D = \frac{1}{\Delta t} I_{n-1,n-1} \text{ et } C = \begin{pmatrix} b_{j,1} & b_{j,2} & 0 & \dots & 0 \\ b_{j,0} & b_{j,1} & b_{j,2} & \ddots & \vdots \\ 0 & b_{j,0} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & b_{j,2} \\ 0 & \dots & 0 & b_{j,0} & b_{j,1} \end{pmatrix}$$

### 2.10.3 Méthode de Crank-Nicolson

La méthode de Crank-Nicolson utilise un  $\theta$ -schéma pour  $\theta = \frac{1}{2}$  : ceci revient à faire une moyenne entre les équations du schéma explicite et celui implicite. De plus, par la linéarité de la moyenne, travailler avec les équations différentielles revient au même que travailler avec les expressions matricielles. Donc, sommons les équations obtenues par les deux schémas avec un coefficient  $\frac{1}{2}$  pour chacune.

$$\text{Méthode explicite : } P_{m+1} = A \times P_m \quad (1)$$

$$\text{Méthode implicite : } P_{m+1} = C^{-1} \times D \times P_m \quad (2)$$

Finalement, pour la méthode de Crank-Nicolson nous obtenons :

$$\bar{P}_{m+1} = \frac{1}{2}(A \times P_m + C^{-1} \times D \times P_m)$$

$$\bar{P}_{m+1} = \frac{1}{2}(A + C^{-1} \times D) \times P_m$$

De même que pour les autres méthodes, nous pouvons itérer numériquement cette relation.

### 2.10.4 Erreurs relatives

Pour estimer les erreurs relatives, nous calculons la distance entre les vecteurs  $P_m$  obtenus par chacune des méthodes aux différences finies et le vrai prix obtenu par la méthode de Black-Scholes.

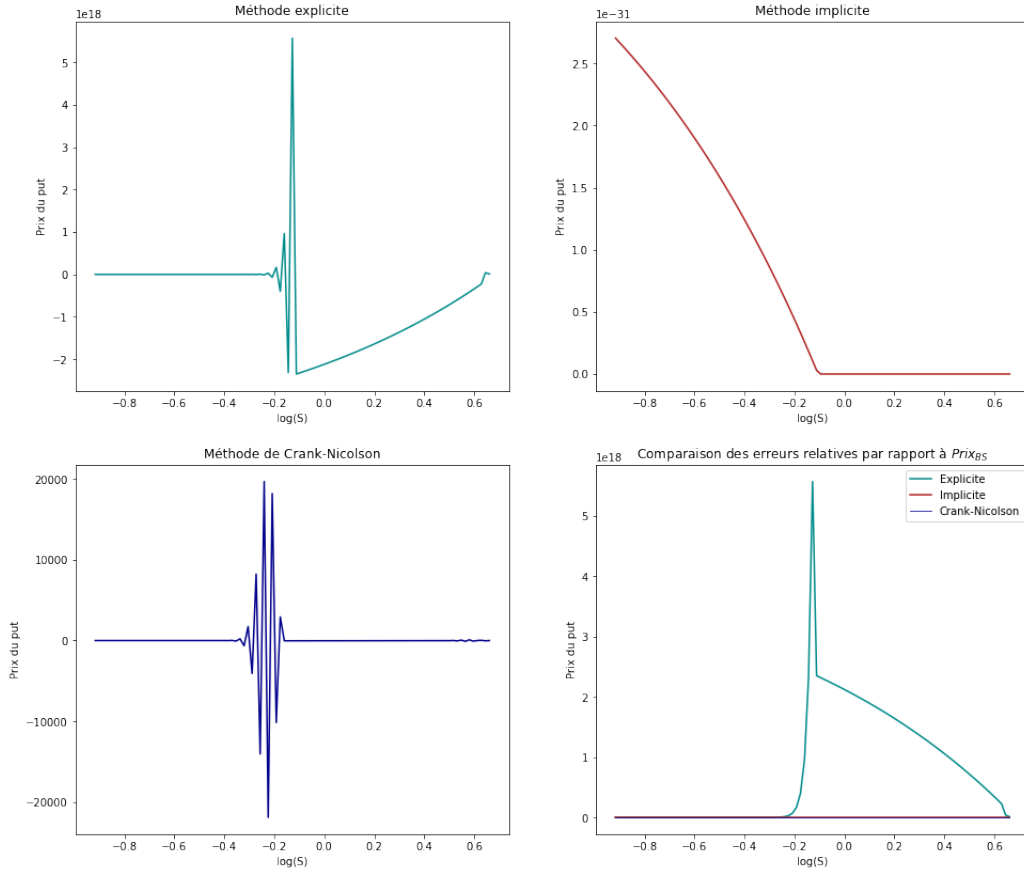


FIGURE 5 – Comparaison des méthodes aux différences finies à Black-Scholes

### 2.10.5 Comparaison et conclusions

Pour la méthode explicite, le principal problème est dû au fait que, lorsque on itère plusieurs fois le calcul, les valeurs propres de la matrice  $A$  font "exploser" les valeurs de ses éléments : on perd en stabilité. D'où tout l'intérêt de choisir une bonne valeur pour  $M$ , afin d'éviter cette instabilité qui engendrera des valeurs trop extrêmes pour les prix du put. Nous remarquons également que pour  $M = N$  notre méthode n'est pas du tout stable : en effet, les valeurs observées sont aberrantes.

Concernant la méthode implicite, on remarque qu'elle est stable quelque soit la valeur choisie pour  $M$ , même en cas d'égalité avec  $N$ . De plus, la méthode de Crank-Nicolson, qui n'est qu'un cas particulier de la méthode theta, a la même stabilité que la méthode implicite. Cependant elle converge plus rapidement vers le vrai prix du put.

On peut donc dire que pour déterminer le prix d'un put vérifiant l'équation aux dérivées partielles de Black-Scholes, on va préférer la méthode de Crank-Nicolson (ou plus généralement la méthode theta). En effet, sa stabilité et sa vitesse de convergence nous permettent, en sachant la maturité de notre put et le prix de son actif risqué, d'estimer au mieux son prix en se rapprochant le plus possible du vrai prix obtenu par la méthode de Black-Scholes.

La précision des trois méthodes est principalement affectée par les erreurs de troncature résultant de l'approximation aux différences finies dans les développements en série de Taylor. En particulier, l'équation différentielle du modèle d'évaluation des options Black-Scholes peut être transformée en équation de chaleur, et ainsi des solutions numériques pour l'évaluation des options peuvent être obtenues avec la méthode Crank-Nicolson.

## 3 Conclusion

L'importance des méthodes différences finies pour la finance est que les problèmes d'évaluation des options, lorsqu'ils sont étendus au-delà des hypothèses standard, ne peuvent pas être résolus par la formule fermée, mais peuvent être résolus en utilisant ces méthodes.

La tarification des produits dérivés a été facilitée par le développement du modèle Black-Scholes tel que discuté dans cette deuxième partie de projet. La mise en œuvre de la simulation de Monte-Carlo et la méthode Crank-Nicolson a facilité la comparaison des résultats obtenus par ces méthodes numériques avec la solution explicite par la formule de Black-Scholes.

Le modèle Black-Scholes suppose cependant que l'actif sous-jacent suit une distribution normale. En réalité, les rendements des actifs sont connus pour suivre des distributions à queue lourde. Ce qui nous conduit en réalité à l'utilisation de modèles qui incorporent la queue lourde comme par exemple la variance Gamma ou la loi de Pareto.