ENSIIE

Rapport de projet Pap

Projet de PAP : Equation de Black et Scholes

Mame Diarra Toure

Imane Alla

Enseignants :

M. Torri

12/01/2020

Projet PAP 2019-2020

Mame Diarra Toure

Imane Alla

12 janvier 2020

Table des matières

1 Introduction

1.1 Contexte

En 1973, Fischer Black et Myron Scholes ont développé un modèle pour calculer le prix d'une option, dite européenne, liant le prix de cette option aux variations de l'actif sous-jacent (que l'on peut considérer comme une action). Leur modélisation du prix de l'actif sous-jacent S_t vérifie l'équation différentielle stochastique suivante :

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

où σ est la volatilité, μ une dérive, et W_t est un processus de Wiener.
En utilisant la formule d'Ito, on se ramène à un problème déterministe sous la forme d'une équation aux dérivées partielles (EDP) vérifiée par la fonction $C(t,S)$ définie sur $[0,T] \times [0,L]$:
$\forall t, S \in [0,T] \times [0,L]$:

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC \qquad (1)$$

Á l'aide d'un changement de variable, il est possible de transformer l'équation (1) en une forme réduite où la fonction $\tilde{C}$ satisfait

$$\frac{\partial \tilde{C}}{\partial \tilde{t}} = \mu \frac{\partial^2 \tilde{C}}{\partial x^2} \qquad (2)$$

1.2 Objectif

Le but de ce projet est de résoudre numériquement en utilisant respectivement la méthode des differences finies de Crank-nicholson et la méthode des differences finies implcites les EDP (1) et (2).
Les solutions obtenues ont été implémentées en C++.

2 Note importante

Lors de nos recherches sur l'équation de Blackscholes nous avons trouvé une relation liant le Call et le put nommé parité CALL-PUT. La parité put-call (Put-Call Parity) définit une relation entre le prix d'un call (option d'achat) et celui d'un put (option de vente), qui ont tous deux le même prix d'exercice(K) et la même échéance. La formule suppose que les options ne sont pas exercées avant échéance ce qui est le cas des options européennes. Elle s'enonce comme suit : soit C(t) valeur du call en un instant t , P(t) valeur du put en un instant t, S(t) la valeur de l'actif sous-jacent en un instant t , K le prix d'exercice ou strike et enfin r le taux d'interet. La parite call-put stipule que $C(t) - P(t) = S(t) - K\exp^{-rt}$. Cependant nous avons remarqué que les conditions en L de l'énoncé ne respectaient pas cette parité. **N'étant pas sur, nous avons procéder à la résolution en utlisant les données de l'énoncé.**

3 Diagramme des classes de notre résolution

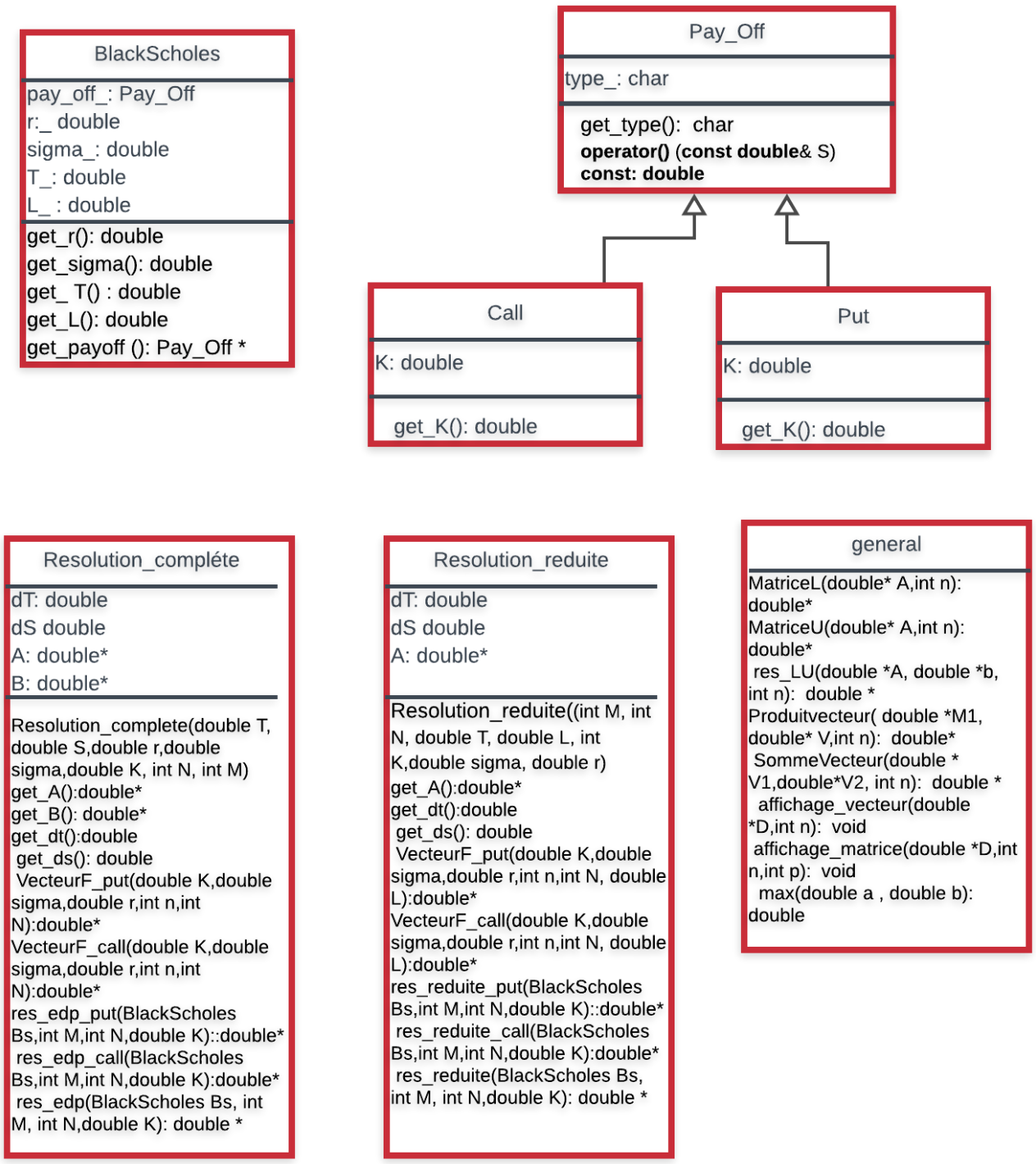

FIGURE 1 – diagramme UML adopté pour la résolution

Pour implémenter notre résolution en c++ nous avons créé 7 classes. Nous avons créé une classeBlack-Scholes permettant de créer une EDP de blackScholes avec les paramètres qui conviennent. La résolution de cette EDP dépend de la nature du payoff qui peut etre soit un call soit un put (d'où la classe payoff heritée par les classes call et put). Pour résoudre les deux équations (1) et (2) , nous avons implémenté la décomposition LU ainsi que la résolution des systémes triangulaires dans la classe général. Enfin les différentes meéthodes ont été mises en pratique pour résoudre l'équation dans la classe résolution$_{c}omplètetrésolution_{r}éduite$.

4 Résolution de l'EDP compléte de Black Scholes

Dans l'EDP complète de Black Scholes, le payoff C vérifie l'équation différentielle suivante :

$\forall t, \mathrm{S} \in [0,\mathrm{T}] \times [0,\mathrm{L}]$:

$$\frac{\partial \mathrm{C}}{\partial t} + r\mathrm{S}\frac{\partial \mathrm{C}}{\partial \mathrm{S}} + \frac{1}{2}\sigma^2\mathrm{S}^2\frac{\partial^2 \mathrm{C}}{\partial \mathrm{S}^2} = r\mathrm{C}$$

Les conditions de bords ainsi que la condition terminale nous donnent les valeurs de C en :

$$\begin{cases} (\mathrm{T}, s) \quad t.q \quad s \in [0,\mathrm{L}] \qquad (condition\ terminale\ en\ \mathrm{T}) \\ (t, 0) \quad t.q \quad t \in [0,\mathrm{T}] \qquad (condtion\ au\ bord\ en\ 0) \\ (t, \mathrm{L}) \quad t.q \quad t \in [0,\mathrm{T}] \qquad (condtion\ au\ bord\ en\ \mathrm{L}) \end{cases}$$

Ces conditions dependent de la nature du Payoff : Put ou Call.

Pour résoudre cette EDP avec la méthode des différences finies, nous avons opté pour la discrétisation suivante :

Soit $\mathrm{N}, \mathrm{M} \in \mathbb{N}^*$:

$$\begin{cases} \Delta\mathrm{T} = \frac{\mathrm{T}}{\mathrm{M}} \quad et \quad \forall n \in \{0,...,\mathrm{M}\}, \boxed{t_n = n\Delta\mathrm{T}} \\ \Delta\mathrm{S} = \frac{\mathrm{L}}{\mathrm{N}+1} \quad et \quad \forall i \in \{1,...,\mathrm{N}+1\}, \boxed{\mathrm{S}_i = i\Delta\mathrm{S}} \end{cases}$$

4.1 Grille schématisant la résolution

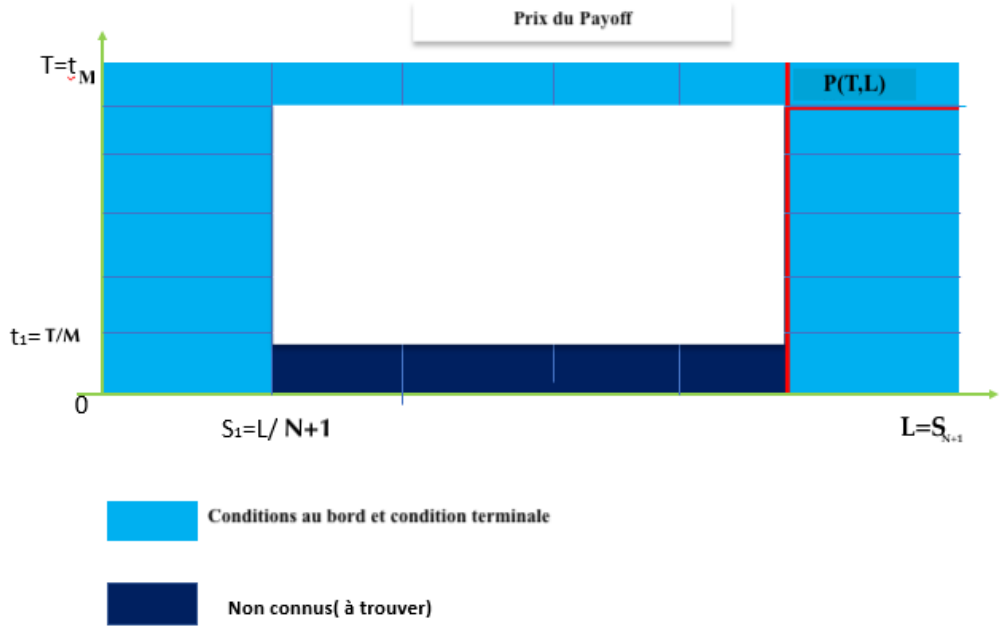

FIGURE 2 – Grille de points schématisant l'approche

4.2 Changement de variable et passage d'une condition terminale à une condition initiale

On veut une condition initiale et non terminale, posons donc $\mathbf{u}(\mathbf{t},\mathbf{S}) = \mathbf{C}(\mathbf{T}-\mathbf{t},\mathbf{S})$.

On obtient donc :

$$\frac{\partial u}{\partial t}(t,\mathrm{S}) = -\frac{\partial \mathrm{C}}{\partial t}(\mathrm{T}-t,\mathrm{S})$$

l'équation différentielle :

$$\frac{\partial \mathrm{C}}{\partial t} = -r\mathrm{S}\frac{\partial \mathrm{C}}{\partial \mathrm{S}} - \frac{1}{2}\sigma^2\mathrm{S}^2\frac{\partial^2 \mathrm{C}}{\partial \mathrm{S}^2} + r\mathrm{C}$$

devient :

$$\frac{\partial u}{\partial t}(t,\mathrm{S}) = r\mathrm{S}\frac{\partial u}{\partial \mathrm{S}}(t,\mathrm{S}) + \frac{1}{2}\sigma^2\mathrm{S}^2\frac{\partial^2 u}{\partial \mathrm{S}^2}(t,\mathrm{S}) - r\,u(t,\mathrm{S})$$

4.3 Approche numérique de l'EDP de Black-Scholes par différences finies : Crank Nicholson

Le Schéma est le suivant : $u_{n,i}$ sera l'approximation de $u(t_n, s_i)$

$\forall (n,i) \in \{1,...,\mathrm{M}-1\} \times \{1,...,\mathrm{N}\}$ on a :

$$\frac{\partial u}{\partial t}(t_n, s_i) \simeq \frac{1}{\Delta \mathrm{T}}\left(u_{n+1,i} - u_{n,i}\right)$$

$$\frac{\partial u}{\partial \mathrm{S}}(t_n, \mathrm{S}_i) \simeq \frac{\frac{1}{2}(u_{n+1,i+1} - u_{n+1,i}) + \frac{1}{2}(u_{n,i+1} - u_{n,i})}{\Delta \mathrm{S}}$$

$$\frac{\partial^2 u}{\partial \mathrm{S}^2}(t_n, \mathrm{S}_i) \simeq \frac{\frac{1}{2}(u_{n+1,i+1} - 2u_{n+1,i} + u_{n+1,i-1}) + \frac{1}{2}(u_{n,i+1} - 2u_{n,i} + u_{n,i-1})}{\Delta \mathrm{S}^2}$$

En approximant l'équation différentielle $\forall (n,i) \in \{0,...,\mathrm{M}-1\} \times \{1,...,\mathrm{N}\}$:

$$\boxed{\frac{\partial u}{\partial t}(t_n, \mathrm{S}_i) = r\mathrm{S}_i \frac{\partial u}{\partial \mathrm{S}}(t_n, \mathrm{S}_i) + \frac{1}{2}\sigma^2 \mathrm{S}_i^2 \frac{\partial^2 u}{\partial \mathrm{S}^2}(t_n, \mathrm{S}_i) - r\,u(t_n, \mathrm{S}_i)}$$

On remplace donc les derivées partielles par leur expression et on obtient

$$\frac{1}{\Delta \mathrm{T}}\left(u_{n+1,i} - u_{n,i}\right) = r\mathrm{S}_i \frac{\frac{1}{2}(u_{n+1,i+1} - u_{n+1,i}) + \frac{1}{2}(u_{n,i+1} - u_{n,i})}{\Delta \mathrm{S}} +$$

$$\frac{1}{2}\sigma^2 \mathrm{S}_i^2 \frac{\frac{1}{2}(u_{n+1,i+1} - 2u_{n+1,i} + u_{n+1,i-1}) + \frac{1}{2}(u_{n,i+1} - 2u_{n,i} + u_{n,i-1})}{\Delta \mathrm{S}^2} - r\,u_{n,i} \quad (3)$$

Afin de simplifier l'expression de (2), on pose :

$$a = \frac{1}{2} r \Delta \mathrm{T} i$$

$$b = \frac{1}{2}\sigma^2 \Delta \mathrm{T} i^2$$

$$c = \frac{1}{2} r \Delta \mathrm{T} i$$

$$d = \frac{1}{2}\sigma^2 \Delta T i^2$$

Alors (3) devient :

$$(1 + a + b)u_{n+1,i} - (a + \frac{1}{2}b)u_{n+1,i+1} - \frac{1}{2}bu_{n+1,i-1} =$$

$$(1 - c - r\Delta T - d)u_{n,i} + (\frac{1}{2}d + c)u_{n,i+1} + \frac{1}{2}d u_{n,i-i}$$

4.4 Traduction matricielle de (3)

Nous allons regrouper les termes en $u_{n+1,i}$ à gauche et $u_{n,i}$ à droite

Alors (3) devient :

$$(1 + a + b)u_{n+1,i} - (a + \frac{1}{2}b)u_{n+1,i+1} - \frac{1}{2}bu_{n+1,i-1} =$$

$$(1 - c - r\Delta T - d)u_{n,i} + (\frac{1}{2}d + c)u_{n,i+1} + \frac{1}{2}d u_{n,i-i}$$

On pose encore :

$$A_1 = 1 + a + b$$

$$A_2 = -(a + \frac{1}{2}b)$$

$$A_3 = -\frac{1}{2}b$$

Et

$$B_1 = 1 - c - d$$

$$B_2 = \frac{1}{2}d + c$$

$$B_3 = \frac{1}{2}d$$

(2) devient donc :

$$A_1 u_{n+1,i} + A_2 u_{n+1,i+1} + A_3 u_{n+1,i-1} = B_1 u_{n,i} + B_2 u_{n,i+1} + B_3 u_{n,i-1}$$

Ce qu'on peut représenter sous forme de :

$$\boxed{AU_{n+1} = BU_n + F_n}$$

Où Un est un vecteur colonne tel que :

$$U_n = \begin{pmatrix} u_{n,1} \\ \dots \\ u_{n,M} \end{pmatrix}$$

Et A et B sont deux matrices carrées tridiagonales de taille N :

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & A_{3,2} & A_{3,3} & A_{3,4} & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & A_{j,j-1} & A_{j,j} & A_{j,j+1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & A_{N-1,N} \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & A_{N,N-1} & A_{N,N} \end{pmatrix}$$

Et

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} & 0 & 0 & ... & ... & ... & ... & ... & 0 \\ B_{2,1} & B_{2,2} & B_{2,3} & 0 & ... & ... & ... & ... & ... & 0 \\ 0 & B_{3,2} & B_{3,3} & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & Bj, j-1 & B_{j,j} & B_{j,j+1} & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & B_{N-1,N} \\ 0 & 0 & ... & ... & ... & ... & ... & 0 & B_{N,N-1} & B_{N,N} \end{pmatrix}$$

avec :

$$A_{j,j} = 1 + a + b \quad \forall j \in \{1, ..., N\}$$

$$A_{j,j+1} = -(a + \frac{1}{2}b \quad \forall j \in \{1, ..., N-1\}$$

$$A_{j,j-1} = -\frac{1}{2}b \quad \forall j \in \{2, ..., N\}$$

et on a

$$B_{j,j} = 1 - c - d - r\Delta T \quad \forall j \in \{1, ..., N\}$$

$$B_{j,j+1} = \frac{1}{2}d + c \quad \forall j \in \{1, ..., N-1\}$$

$$B_{j,j-1} = \frac{1}{2}d \quad \forall j \in \{2, ..., N\}$$

et F_n est un vecteur colonne tel que :

$$F_n = \begin{pmatrix} F_{n,1} \\ ... \\ F_{n,N} \end{pmatrix}$$

$$F_n = \begin{pmatrix} F_g \\ 0 \\ \dots \\ 0 \\ F_d \end{pmatrix}$$

ou F_d et F_g représentent les conditions au bord à droite et à en gauche c'est a dire en 0 et en L.

4.5 Vecteur F_n dans le cas ou le payoff est un call

Dans le cas ou le payoff est un call les condtions aux bords sont :

$$\begin{cases} C(t,L) &= Ke^{-r(t-T)}, \quad \forall t \in [0,T] \quad F_d, \\ C(t,0) &= 0, \quad \forall t \in [0,T] \quad F_g. \end{cases}$$

Etant donné notre changement de variable $u(t,S) = C(T-t,S)$ les conditions au bord deviennent :

$$\begin{cases} u(t,L) &= Ke^{rt}, \quad \forall t \in [0,T] \quad F_d, \\ u(t,0) &= 0, \quad \forall t \in [0,T] \quad F_g. \end{cases}$$

d'où

$$F_n = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ Ke^{rn\Delta t} \end{pmatrix}$$

4.6 Vecteur F_n dans le cas ou le payoff est un put

Dans le cas ou le payoff est un call les conditions aux bords sont :

$$\begin{cases} \mathrm{C}(t,0) &= \mathrm{K}e^{r(t-\mathrm{T})}, \quad \forall t \in [0,\mathrm{T}] \quad \mathrm{F}_g, \\ \mathrm{C}(t,\mathrm{L}) &= 0, \quad \forall t \in [0,\mathrm{T}] \quad \mathrm{C}_\mathrm{L}. \end{cases}$$

Etant donné notre changement de variable $u(t,\mathrm{S}) = \mathrm{C}(\mathrm{T}-t,\mathrm{S})$ les conditions au bord deviennent :

$$\begin{cases} u(t,\mathrm{L}) &= \mathrm{K}e^{-rt}, \quad \forall t \in [0,\mathrm{T}] \quad \mathrm{F}_d, \\ u(t,0) &= 0, \quad \forall t \in [0,\mathrm{T}] \quad \mathrm{F}_g. \end{cases}$$

d'où

$$\mathrm{F}_n = \begin{pmatrix} \mathrm{K}e^{-rn\Delta t} \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix}$$

4.7 Résolution numérique de l'équation différentielle de Black-Scholes-Utlilisation de la decomposition LU

Pour determiner $\mathrm{C}(0,s)\forall s \in [0,\mathrm{L}]$, c'est a dire $\mathrm{U}(\mathrm{T},s)\forall s \in [0,\mathrm{L}]$, nous devons résoudre le système

$$\boxed{\mathrm{AU}_{n+1} = \mathrm{BU}_n + \mathrm{F}_n} \quad \forall n \in [0,\mathrm{M}]$$

Le dernier vecteur $\mathbf{U_{M+1}}$ est la solution cherchée.

Pour résoudre le sytème, nous allons procéder a la decomposition LU de la matrice A et ainsi resoudre 2 systemes triangulaires. **La matrice A est une matrice tridiagonale**

de ce fait sa decomposition LU est la suivante :

Soit la suite d_n definit comme suit :

$$\begin{cases} d_0 = 1 \\ d_1 = \mathrm{A}_{1,1} \\ d_k = \mathrm{A}_{k,k} d_{k-1} - \mathrm{A}_{k,k-1}\mathrm{A}_{k-1,k} d_{k-2} \quad \forall k \quad \in \quad [2, \mathrm{N}] \end{cases}$$

Si $\forall k \quad d_k \neq 0$ alors A peut s'écrire $\mathrm{A} = \mathrm{L}\hat{\mathrm{U}}$ avec

$$\mathrm{L} = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ \mathrm{A}_{2,1}\frac{d_0}{d_1} & 1 & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \mathrm{A}_{3,2}\frac{d_1}{d_2} & 1 & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \mathrm{A}_{j,j-1}\frac{d_{j-2}}{d_{j-1}} & 1 & 0 & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & \mathrm{A}_{\mathrm{N},\mathrm{N}-1}\frac{d_{\mathrm{N}-2}}{d_{\mathrm{N}-1}} & 1 \end{pmatrix}$$

et

$$\hat{\mathrm{U}} = \begin{pmatrix} \frac{d_1}{d_0} & \mathrm{A}_{1,2} & 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \frac{d_2}{d_1} & \mathrm{A}_{2,3} & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & \frac{d_3}{d_2} & \mathrm{A}_{3,4} & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & 0 & \frac{d_j}{d_{j-1}} & \mathrm{A}_{j,j+1} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \mathrm{A}_{\mathrm{N}-1,\mathrm{N}} \\ 0 & 0 & \dots & \dots & \dots & \dots & \dots & 0 & 0 & \frac{d_\mathrm{N}}{d_{\mathrm{N}-1}} \end{pmatrix}$$

Une fois la décompositiion LU faite, il ne reste plus qu'à résoudre les deux systéme triangulaire suivant :

$$\mathrm{L}y_n = \mathrm{BU}_n + \mathrm{F}_n$$

et

$$\hat{\mathrm{U}} \times \mathrm{U}_{n+1} = y_n$$

4.7.1 Resolution des systémes triangulaires

Triangulaire inférieur :

Posons : $b_n = \mathrm{BU}_n + \mathrm{F}_n$

$$\begin{cases} y_{n,1} = \frac{b_{n,1}}{\mathrm{L}_{1,1}} \\ y_{n,i} = \frac{1}{\mathrm{L}_{i,i}}(b_{n,i} - \sum_{j=1}^{i-1} \mathrm{L}_{i,j} y_{n,j}) \forall i \in [2, \mathrm{N}] \end{cases}$$

Triangulaire supérieur :

$$\begin{cases} \mathrm{U}_{n+1,\mathrm{N}} = \frac{y_{n,\mathrm{N}}}{\hat{\mathrm{U}}_{\mathrm{N},\mathrm{N}}} \\ \mathrm{U}_{n+1,i} = \frac{1}{\hat{\mathrm{U}}_{i,i}}(y_{n,i} - \sum_{j=i=1}^{\mathrm{N}} \hat{\mathrm{U}}_{i,j} y_{n,j}) \forall i \in [\mathrm{N}-1, 1] \end{cases}$$

4.8 Résolution numérique de l'équation différentielle de Black : Pseudo Code

Algorithm 1 Résolution numérique de l'équation différentielle de Black-Scholes

Require: $\mathrm{M}, \mathrm{N} \in \mathbb{N}, \mathrm{K}, r, \sigma \in \mathbb{R}$
Ensure: $\mathrm{U}_{\mathrm{M}+1} = \mathrm{A}^{-1}\mathrm{B}\mathrm{U}_{\mathrm{N}} + \mathrm{A}^{-1}\mathrm{C}_{\mathrm{N}}$
  $dt = \frac{\mathrm{T}}{\mathrm{M}} \quad ds = \frac{\mathrm{L}}{\mathrm{N}+1}$
  Initialiser A et B
  Calculer U_1 grâce aux conditions intiales (U_1 pas U_{N} car changement de variable)
  Décomposition LU de A
  for $n \quad in \quad 1 : \mathrm{M}$ **do**
    Calculer F_n
    Calculer $b_n = \mathrm{B}\mathrm{U}_n + \mathrm{F}_n$
    Résoudre $\mathrm{L}y_n = b_n$
    Résoudre $\hat{\mathrm{U}}\mathrm{U}_{n+1} = y_n$
  end for
  return $\mathrm{U}_{\mathrm{M}+1}$

5 Implémentation de la solution en C++

Comme indiqué sur le diagramme des classes (Figure 1), nous avons créé une classe Black-Scholes permettant de créer une EDP de blackScholes avec les paramètres qui conviennent. La résolution de cette EDP dépend de la nature du payoff qui peut etre soit un call soit un put (d'où la classe payoff heritée par les classes call et put). Pour résoudre les deux systèmes précédents, nous avons implémenté la décomposition LU ainsi que la résolution des systémes triangulaires dans la classe général. Enfin les

différentes meéthodes ont été mises en pratique pour résoudre l'équation dans la classe résolution_complète.

5.1 Problèmes rencontrés lors de l'implémentation

5.1.1 Représentation des matrices en C++

Lors de notre implémentation, nous avions initialement choisi de représenter les matrices par des tableaux en deux dimensions. Cependant cela augmentait considerablement la complexite de nos algorithmes. Nous avons donc décidé d'utiliser une représentation en une dimension ce qui est bien plus pratique. Ainsi le tableau représentant une matrice est constitué de la succession des lignes de la matrice.

5.1.2 Complexité des fonctions

Un autre problème que nous avons rencontré concerne la complexité de nos fonctions notamment au niveau de la mémoire. En effet, notre approche étant iterative nous avions choisi de conserver a chaque iteration la sortie de l'algorithme dans un tableau. Cependant ce n'etait pas nécessaire car pour obtenir la solution au temp n+1 nous n'avions besoin que du vecteur au temps n. En modifiant notre code en fonction, nous sommes parvenus à pouvoir exécuter notre programme pour une discretisation de 960 en temps et en espace sans soucis.

5.2 Résultat de la résolution de l'EDP de BlackScholes

Notre programmme nous a permis d'obtenir les valeurs suivantes pour le prix C(0, s).

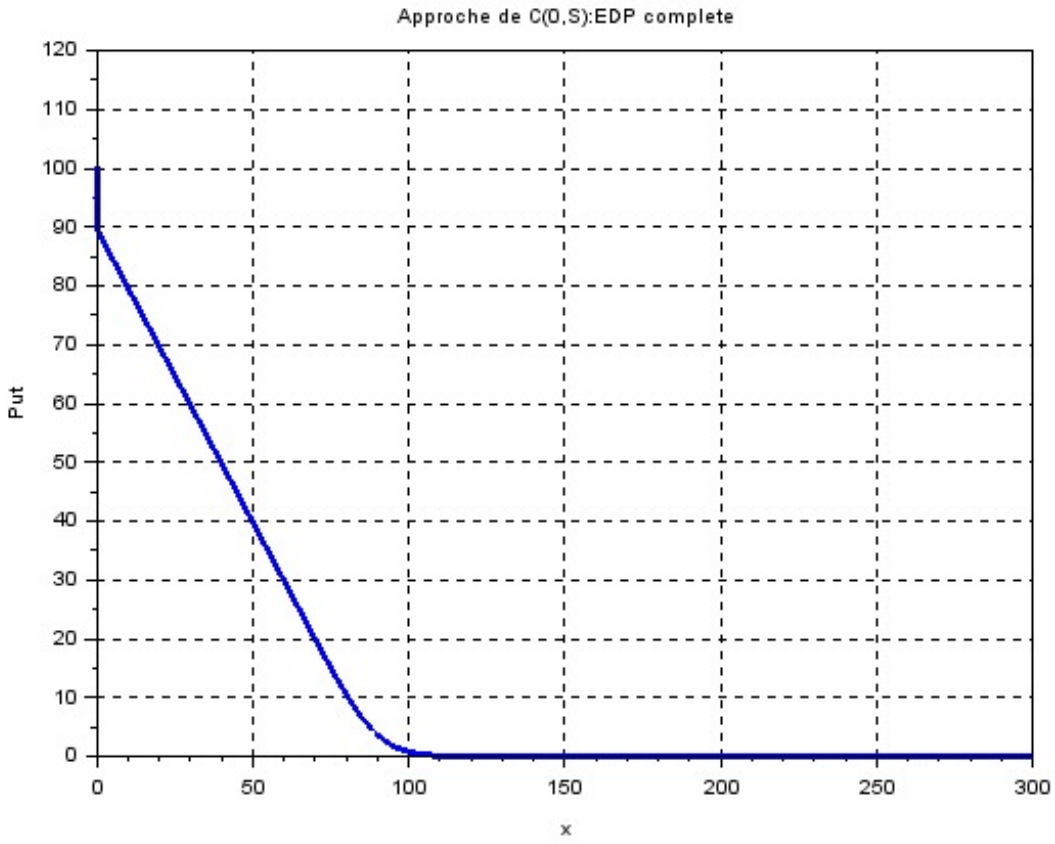

FIGURE 3 – Prix du Put en fonction de s, M=N=960

Le tracé ci-contre correspond bien à celui d'un put.La courbe est décroissante et la valeur du put s'annule au dela du prix d'exercice K = 100 .

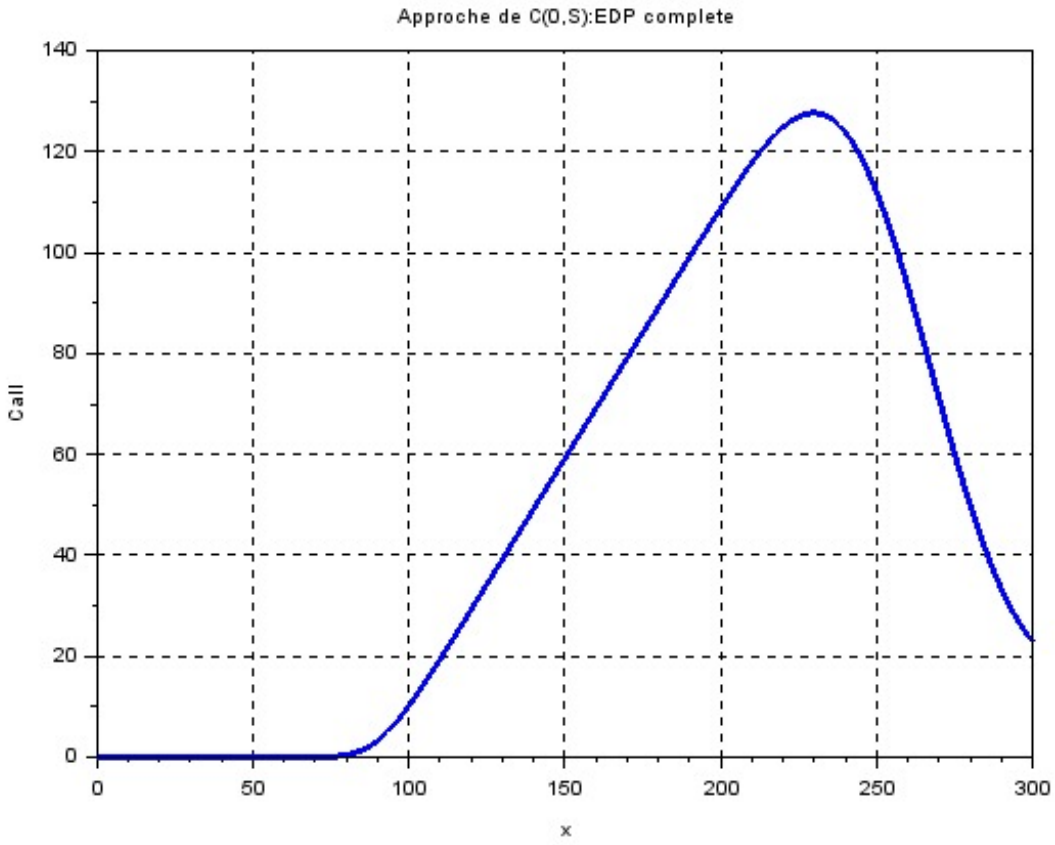

FIGURE 4 – Prix du call en fonction de s, M=N=960

Le tracé ci-contre ne correspond pas exactement à celui d'un call (qui doit etre croissante) . En accord avec la remarque faite sur le non-respect de la Parité Call-Put pour la condition en L , nous avons tracé le call en modifiant la condtion à cette borne et le résultat obtenu etait plus en adequation avec nos attentes.

6 Résolution de l'EDP réduite

Rappelons l'EDP de Black Scholes vérifiée par la fonction C(t,S) définie sur $[0,T] \times [0,L]$:

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

sur $[0,T] \times [0,L]$

6.1 Lien entre l'équation de Black-Scholes et l'équation de la chaleur

On considère le changement de variable suivant :

$$\begin{cases} S = K\exp(x) \\ t = T - \frac{2\tau}{\sigma^2} \\ C(t,S) = Ku(\tau,x) \end{cases}$$

On a donc :

$$u(\tau,x) = \frac{1}{K}C(T - \frac{2\tau}{\sigma^2}, K\exp(x))$$

On dérive une fois par rapport à x :

$\frac{\partial u}{\partial x} = \frac{1}{K}\frac{\partial C}{\partial S}\frac{\partial S}{\partial x} == \frac{S}{K}\frac{\partial C}{\partial S}$

On dérive une seconde fois par rapport à x :

$\frac{\partial^2 u}{\partial x^2} = \frac{S}{K}\frac{\partial C}{\partial S} + \frac{S^2}{K}\frac{\partial^2 C}{\partial S^2}$

On dérive par rapport à τ :

$\frac{\partial u}{\partial \tau} = \frac{1}{K}\frac{\partial C}{\partial t}\frac{\partial t}{\partial \tau} = \frac{1}{K}\frac{\partial C}{\partial t}(\frac{-2}{\sigma^2})$

Exprimons C en fonction de u :

$$\begin{cases} \frac{\partial C}{\partial t} = \frac{-K}{2}\sigma^2 \frac{\partial u}{\partial \tau} \\ \frac{\partial C}{\partial S} = \frac{K}{S}\frac{\partial u}{\partial x} \\ \frac{\partial^2 C}{\partial S^2} = \frac{K}{S^2}\left(\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial x}\right) \end{cases}$$

On revient à l'équation (1) de Black and Scholes :

$$\frac{-K}{2}\sigma^2 \frac{\partial u}{\partial \tau} + \left(rK - \frac{1}{2}K\sigma^2\right)\frac{\partial u}{\partial x} + \frac{1}{2}K\sigma^2 \frac{\partial^2 u}{\partial x^2} = rKu \qquad (3)$$

On divise par $-\frac{K}{2}\sigma^2$, donc :

$$\frac{\partial u}{\partial \tau} = \left(\frac{2r}{\sigma^2} - 1\right)\frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial x^2} - \frac{2r}{\sigma^2}u$$

En posant, $q = \frac{2r}{\sigma^2}$,

(3) devient : $\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} + (q-1)\frac{\partial u}{\partial x} - qu \qquad (4)$

Nous voulons une équation de la forme de (2) ce qui veut dire que nous devons nous débarasser des termes en u et en $\frac{\partial u}{\partial x}$.

Pour se faire posons : $u(\tau, x) = e^{\alpha x + \beta \tau}\tilde{C}(\tau, x)$

(4) devient donc en arrangeant les termes :

$$\frac{\partial \tilde{C}}{\partial \tau} = \frac{\partial^2 \tilde{C}}{\partial x^2} + (2\alpha + q - 1)\frac{\partial \tilde{C}}{\partial x} + (\alpha^2 - \beta - q + \alpha(q-1))\tilde{C} \qquad (5)$$

Pour retrouver la forme canonique de l'équation (2) qui est celle de l'equation de la chaleur , il suffit de choisir α et β suivant le système :

$$\begin{cases} 2\alpha + q - 1 = 0 \\ \\ \alpha^2 - \beta - q + \alpha(q-1) = 0 \end{cases}$$

$$\begin{cases} \alpha = -\frac{1}{2}(q-1) \\ \\ \beta = -\frac{1}{4}(q+1)^2 \end{cases}$$

Les derniers termes de l'équation (5) s'annulent, l'équation précédente se réduit à :

$$\frac{\partial \tilde{C}}{\partial \tau} = \frac{\partial^2 \tilde{C}}{\partial x^2}$$

6.2 Nouvelles conditions

6.2.1 Délimiter le domaine de x :

Nous allons restreindre le domaine à un intervalle borné. Le domaine est choisi de façon à capturer le comportement global de l'option et de maintenir l'exactitude du prix de l'option.

Puisque :$x = \ln(S/K)$, alors, posons $d = \ln(S/K)$. et $\delta x = d/N + 1$

Notre étude se fera sur l'intervalle Par conséquent x$\in [w, d]$. Cela nous amène à spécifier les conditions aux bords et la condition initiale.

Il est à noter que $w = \ln(\frac{0.001}{K})$ a été choisi pour faciliter les calculs des conditions aux bords par la suite. En effet le changement de vraible nous fait passer sur l'intervalle$[-\infty, d]$ mais pour pouvoir discretiser nous avons approximer la borne inferieur par $w = \ln(\frac{0.0001}{K})$ et nous avons approximer la condition en 0.0001 par la condition en 0 .

6.2.2 Réécriture de la condition initiale :

Notons que pour $t = T$ nous avons $\tau = 0$.A l'aide des nouvelles variables τ et x, la condition initiale devient :

— *Un put* : $\tilde{C}(0, x) = \max(0, e^{\frac{1}{2}(q-1)x} - e^{\frac{1}{2}(q+1)x})$ pour x $\in [w, d]$

— *Un call* : $\tilde{C}(0, x) = \max(0, e^{\frac{1}{2}(q+1)x} - e^{\frac{1}{2}(q-1)x})$ pour x $\in [w, d]$

6.2.3 Réécriture des conditions aux bords :

1. *Un put* :

— $\tilde{C}(\tau, w) = e^{-(\alpha w+\beta\tau)} e^{\frac{-2r\tau}{\sigma^2}}$ pour tout $\tau \in [0, \frac{\sigma^2}{2}T]$

— $\tilde{C}(\tau, d) = 0$ pour tout $\tau \in [0, \frac{\sigma^2}{2}T]$

2. *Un call* :

— $\tilde{C}(\tau, w) = 0$ pour tout $\tau \in [0, \frac{\sigma^2}{2}T]$

— $\tilde{C}(\tau, d) = e^{-\alpha \ln(\frac{L}{K})-\beta\tau} e^{\frac{2r\tau}{\sigma^2}}$ pour tout $\tau \in [0, \frac{\sigma^2}{2}T]$

6.3 Approche numérique de l'EDP réduite par différences finies implicites

6.3.1 Discrétisation de $[0, \frac{\sigma^2}{2}T] \times [w, d]$

Soient N,M $\in \mathbb{N}^*$.

Posons : $\begin{cases} \Delta T = \frac{1}{2M}\sigma^2 T, \quad \tau_n = n\Delta T \quad \forall n \in \{0, ..., M\} \\ \\ \Delta x = \frac{1}{N+1}(\ln(\frac{L}{K}) - \ln(\frac{0.001}{K})), \quad x_i = i\Delta x \quad \forall i \in \{0, ..., N+1\} \end{cases}$

6.3.2 Discrétisation de l'EDP réduite

On utilise un θ-schéma qui prend en compte la méthode des différences finies implicite (θ = 1).

$\tilde{C}_{n,i}$ sera l'approximation de $\tilde{C}(\tau_n, x_i)$ tel que :

$\forall (n, i) \in \{0, ..., M-1\} \times \{1, ..., N\}$, on a :

$$\frac{\partial \tilde{C}}{\partial \tau}(\tau_n, x_i) \simeq \frac{1}{\Delta T}\left(\tilde{C}_{n+1,i} - \tilde{C}_{n,i}\right)$$

$$\frac{\partial^2 \tilde{C}}{\partial x^2}(\tau_n, x_i) \simeq \frac{(\tilde{C}_{n+1,i+1} - 2\tilde{C}_{n+1,i} + \tilde{C}_{n+1,i-1})}{\Delta x^2}$$

En approximant l'équation différentielle $\forall (n,i) \in \{0,...,M-1\} \times \{1,...,N\}$:

$$\boxed{\frac{\partial \tilde{C}}{\partial \tau}(\tau_n, x_i) = \frac{\partial^2 \tilde{C}}{\partial x^2}(\tau_n, x_i)}$$

On remplace donc les dérivées partielles par leur expression et on obtient

$$\frac{1}{\Delta T}\left(\tilde{C}_{n+1,i} - \tilde{C}_{n+1,i}\right) = \frac{(\tilde{C}_{n+1,i+1} - 2\tilde{C}_{n+1,i} + \tilde{C}_{n,i-1})}{\Delta x^2} \quad (6)$$

6.4 Traduction matricielle de (6)

Comme précédemment,on regroupe les termes en $\tilde{C}_{n+1,i}$ à gauche et $\tilde{C}_{n,i}$ à droite. Afin de simplifier, on pose :

$$\begin{cases} a = 1 + \frac{2\Delta T}{\Delta x^2} \\ \\ b = \frac{\Delta T}{\Delta x^2} \end{cases}$$

Alors (6) devient :

$$a\tilde{C}_{n+1,i} + b\tilde{C}_{n+1,i+1} + b\tilde{C}_{n+1,i-1} = \tilde{C}_{n,i}$$

Ce qu'on peut représenter sous forme de

$$\boxed{AU_{n+1} = U_n + F_n} \qquad (7)$$

Où U_n est un vecteur colonne tel que :

$$U_n = \begin{pmatrix} \tilde{C}_{n,1} \\ ... \\ \tilde{C}_{n,M} \end{pmatrix}$$

Et A est une matrice carrée tridiagonale de taille N :

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & 0 & 0 & ... & ... & ... & ... & ... & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 & ... & ... & ... & ... & ... & 0 \\ 0 & A_{3,2} & A_{3,3} & A_{3,4} & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & A_{j,j-1} & A_{j,j} & A_{j,j+1} & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & A_{N-1,N} \\ 0 & 0 & ... & ... & ... & ... & ... & 0 & A_{N,N-1} & A_{N,N} \end{pmatrix}$$

$$\begin{cases} a = 1 + \frac{2\Delta T}{\Delta x^2} \\ \\ b = \frac{\Delta T}{\Delta x^2} \end{cases}$$

avec :

$$A_{j,j} = a \qquad \forall j \in \{1, ..., N\}$$

$$A_{j,j+1} = -b \qquad \forall j \in \{1, ..., N-1\}$$

$$A_{j,j-1} = -b \qquad \forall j \in \{2, ..., N\}$$

(7) peut s'écrire donc comme suit :

$$\boxed{\mathrm{AU}_{n+1} = \mathrm{U}_n + \mathrm{F}_n}$$

F_n est un vecteur colonne tel que :

$$\mathrm{F}_n = \begin{pmatrix} \mathrm{F}_{n,1} \\ \\ \dots \\ \\ \mathrm{F}_{n,\mathrm{N}} \end{pmatrix}$$

$$\mathrm{F}_n = \begin{pmatrix} \mathrm{F}_g \\ 0 \\ \dots \\ 0 \\ \mathrm{F}_d \end{pmatrix}$$

ou F_d et F_g représentent les conditions au bord à droite et à en gauche c'est a dire en w et en d.

6.5 Vecteur F_n dans le cas ou le payoff est un call

Dans le cas ou le payoff est un call les condtions aux bords sont :

— $\tilde{\mathrm{C}}(\tau, w) = 0$ pour tout $\tau \in [0, \frac{\sigma^2}{2}\mathrm{T}]$

— $\tilde{\mathrm{C}}(\tau, d) = e^{-\alpha d - \beta\tau} e^{\frac{2r\tau}{\sigma^2}}$ pour tout $\tau \in [0, \frac{\sigma^2}{2}\mathrm{T}]$

d'où

$$\mathrm{F}_n = b \times \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \\ e^{-\alpha \ln(\frac{\mathrm{L}}{\mathrm{K}}) - \beta\tau} e^{\frac{2r\tau}{\sigma^2}} \end{pmatrix}$$

6.6 Vecteur F_n dans le cas ou le payoff est un put

Dans le cas ou le payoff est un call les conditions aux bords sont :

— $\tilde{\mathrm{C}}(\tau, w) = e^{\alpha \ln(\frac{\mathrm{L}}{\mathrm{K}}) - \beta\tau} e^{\frac{-2r\tau}{\sigma^2}}$ pour tout $\tau \in [0, \frac{\sigma^2}{2}\mathrm{T}]$

— $\tilde{\mathrm{C}}(\tau, d)) = 0$ pour tout $\tau \in [0, \frac{\sigma^2}{2}\mathrm{T}]$

d'où

$$\mathrm{F}_n = b \times \begin{pmatrix} e^{-(\alpha \ln(\frac{0.001}{\mathrm{K}}) + \beta\tau)} e^{\frac{-2r\tau}{\sigma^2}} \\ 0 \\ \dots \\ 0 \\ 0 \end{pmatrix}$$

6.7 Résolution numérique de l'EDP réduite : Pseudo Code

Algorithm 2 Résolution numérique de l'EDP réduite

Require: $M, N \in \mathbb{N}, K, r, \sigma \in \mathbb{R}$
Ensure: $U_{M+1} = A^{-1}U_N + A^{-1}C_N$
$dt = \frac{1}{2M}\sigma^2 T \quad dx = \frac{1}{N+1}(\ln(\frac{L}{K}) - \ln(\frac{0.001}{K}))$
Initialiser A
Calculer U_1 grâce aux conditions intiales
Décomposition LU de A
for $n \quad in \quad 1 : M$ **do**
Calculer F_n
Calculer $b_n = U_n + F_n$
Résoudre $Ly_n = b_n$
Résoudre $\hat{U}U_{n+1} = y_n$
end for
return U_{M+1}

7 Implementation de la solution en C++

Comme indiqué précédemment, nous utilisons la classe Black-Scholes pour initialiser les paramètres de l'EDP. La résolution de cette EDP dépend de la nature du payoff qui peut soit un call soit un put (d'où la classe payoff heritée par les classes call et put). Pour résoudre les deux systèmes précédents, nous avons implémenté la decomposition LU ainsi que la résolution des systèmes triangulaires dans la classe general. Enfin les differentes methodes ont été mises en pratique pour résoudre l'equation dans la classe résolution_réduite.

7.1 Problèmes rencontrés lors de l'implémentation

7.2 Discretisation de l'espace

Après avoir procéder au changement de variable pour obtenir l'équation réduite à partir de l'EDP de Black Scholes, on a rencontré un problème avec la discrétisation spatiale ainsi que la condition en S = 0 pour une option put car le nouveau espace s'étale sur $]-\infty, \ln\left(\frac{L}{K}\right)]$.On a pas réussi à choisir un intervalle borné de façon à capturer le comportement global de l'option.

7.2.1 Supperposition des solutions

Un autre problème que nous avons rencontré concerne la superposition des courbes approchant $C(0,s)$ et $\tilde{C}(0,s)$.Les résultats obtenus ne sont pas du meme ordre de grandeur. Il est à noter qu'on a appliqué le changement de variable "inverse" pour retourner de $\tilde{C}(0,s)$ à $C(0,s)$ avant de les comparer.

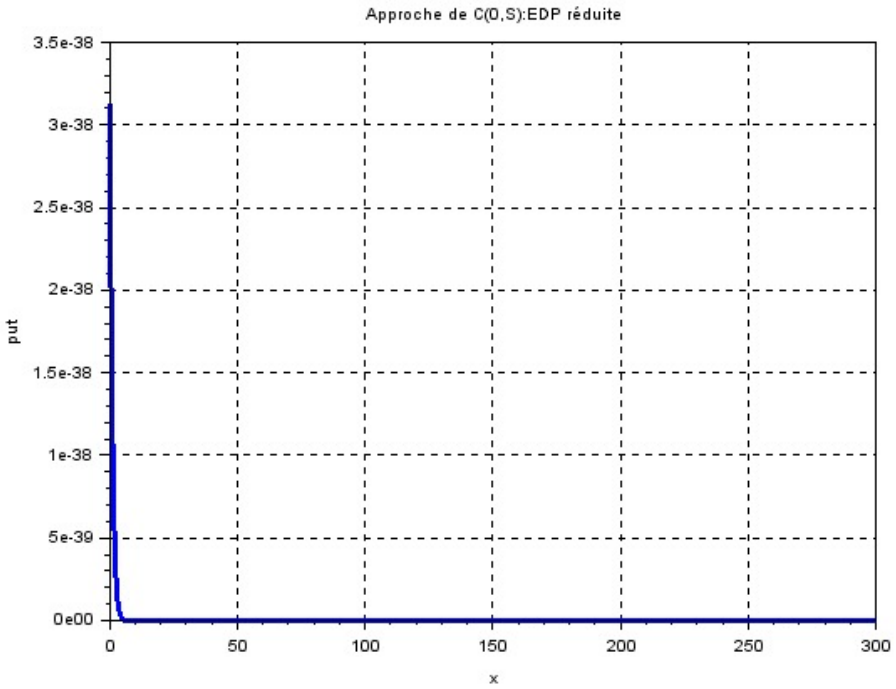

FIGURE 5 – Prix du put en fonction de s, M=N=960

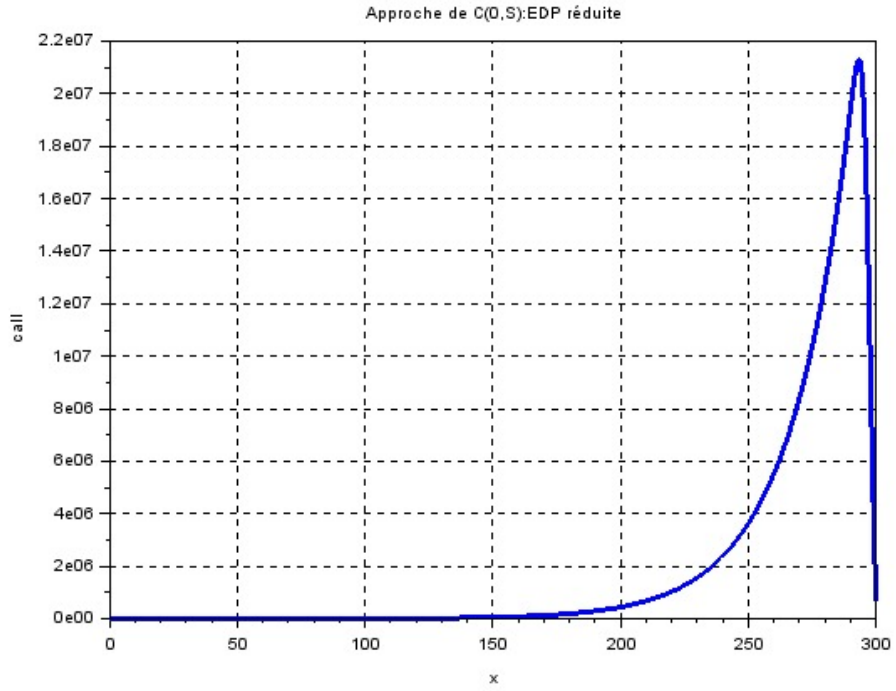

FIGURE 6 – Prix du call en fonction de s, M=N=960

7.3 Tracé de l'erreur relative entre la solution de l'equation reduite et la solution de l'equation de blackscholes

Nous avons utilisé la formule suivante pour determiner l'erreur relative entre la solution de l'equation de Blackscholes et celle obtenue avec l'équation réduite.

$$erreur\_relative(\mathrm{A},\mathrm{B}) = \frac{||\mathrm{A}-\mathrm{B}||}{||\mathrm{B}||}$$

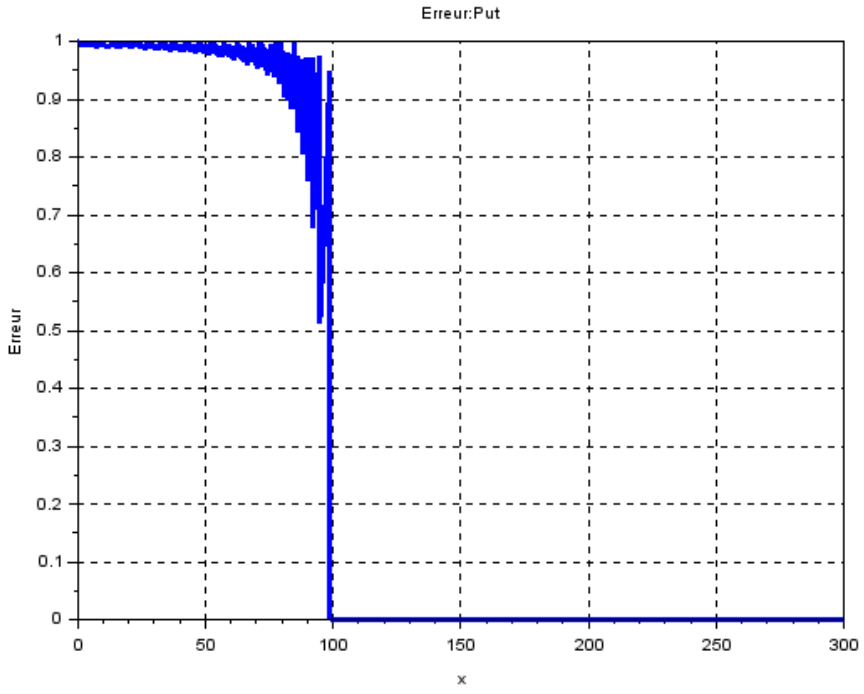

FIGURE 7 – Erreur entre les deux solutions dans le cas d'un ou le payoff est un put, M=N=960

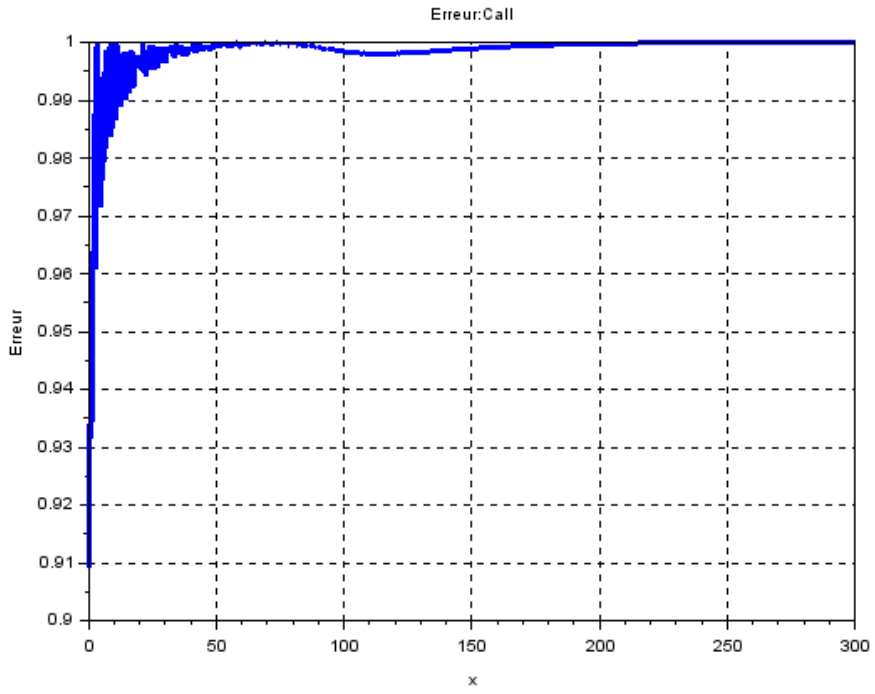

FIGURE 8 – Erreur entre les deux solutions dans le cas d'un ou le payoff est un call, M=N=960

8 Conclusion

Au cours de ce projet, nous avons formulé, résolu et présenté les résultats numériques obtenus pour l'évaluation d'option Européenne par la resolution de l'equation de blackScholes et de l'équation reduite qui en découle.Nous sommes confiants par rapport aux résultats obtenus pour la l'EDP de black scholes, cependant les valeurs obtenues par la résolution de l'equation réduite nous semblent aberrantes. Nous pensons que le probléme vient de la discrétisation en espace étant donné que nos différents changements de variables nous projette dans un espace non borné.