

# Rapport Projet PAP : Equation de Black et Scholes

Mathieu Truc, Loïc Mary

## Table des matières

<b>1</b>	<b>EDP Complète : Théorie</b>	<b>2</b>
1.1	Représentation par les différences finies . . . . .	2
1.2	Schéma de Cranck-Nicholson . . . . .	2
<b>2</b>	<b>EDP Réduite : Théorie</b>	<b>4</b>
2.1	Changement de variable . . . . .	4
2.2	Représentation par les différences finies . . . . .	6
2.3	Schéma implicite . . . . .	6
<b>3</b>	<b>Implémentation en C++ : Structure des classes, Problèmes &amp; Solutions</b>	<b>8</b>
3.1	Compilation . . . . .	8
3.2	Conception des classes . . . . .	8
3.3	Problèmes & Solutions . . . . .	10
3.3.1	Problème de discrétisation . . . . .	10
3.3.2	Problème avec la résolution du système linéaire . . . . .	11
<b>4</b>	<b>Résultats : Affichage des courbes</b>	<b>12</b>
4.1	Cas d'un Put . . . . .	13
4.2	Cas d'un Call . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>15</b>

# 1 EDP Complète : Théorie

Le schéma de Cranck-Nicholson est un algorithme se basant sur une représentation par les différences finies permettant d'approcher une fonction vérifiant une équation aux dérivées partielles.

## 1.1 Représentation par les différences finies

Commençons par représenter l'équation de Black-Scholes par les différences finies. On effectue un maillage de l'espace de départ de la fonction  $C$ . C'est à dire que nous discrétisons  $[0, T]$  et  $[0, L]$  en, respectivement,  $N_t$  et  $N_S$  intervalles réguliers.

Les conditions initiales du problème sont des conditions terminales (pour la variable  $t$ ), pour revenir à un problème standard nous effectuons un changement de variable  $\tau = T - t$ . L'équation de Black-Scholes se réécrit alors,

$$\frac{\partial C}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} + rS \frac{\partial C}{\partial S} - rC \quad (1)$$

On pose maintenant, pour tout  $n$  dans  $\llbracket 0, N_S \rrbracket$  et pour tout  $j$  dans  $\llbracket 0, N_t \rrbracket$ ,  $S_n = n\Delta S$  et  $\tau_j = j\Delta t$  où  $\Delta S = \frac{L}{N_S}$  et  $\Delta t = \frac{T}{N_t}$ . Ces points correspondent au maillage de l'espace, notre but est de calculer de proche en proche  $C(\tau_{N_t}, S_n)$  pour tout  $n$ . Dans la suite on notera pour tout  $n$  et pour tout  $j$ ,  $C_{n,j} = C(\tau_j, S_n)$ . Les valeurs initialement connues sont donc, pour tout  $j$ ,  $C_{0,j}$  et  $C_{N_S,j}$  ainsi que pour tout  $n$ ,  $C_{n,0}$ .

## 1.2 Schéma de Cranck-Nicholson

Notre but est d'exploiter l'équation différentielle évaluée en tous les points  $n\Delta S$  et  $j\Delta t$  pour trouver des relations entre les points du maillage de l'espace.

Le schéma de Cranck-Nicholson approche les dérivées partielles de la manière suivante :

$$\begin{aligned} \frac{\partial C}{\partial \tau}(n\Delta S, j\Delta t) &\approx \frac{C_{n,j+1} - C_{n,j}}{\Delta t} \\ \frac{\partial C}{\partial S}(n\Delta S, j\Delta t) &\approx \frac{1}{2} \left[ \frac{C_{n+1,j+1} - C_{n-1,j+1}}{2\Delta S} + \frac{C_{n+1,j+1} - C_{n-1,j}}{2\Delta S} \right] \\ \frac{\partial^2 C}{\partial S^2}(n\Delta S, j\Delta t) &\approx \frac{1}{2} \left[ \frac{C_{n+1,j+1} - 2C_{n,j+1} + C_{n-1,j+1}}{\Delta S^2} + \frac{C_{n+1,j} - 2C_{n,j} + C_{n-1,j}}{\Delta S^2} \right] \end{aligned}$$

On pose pour tout  $n$  et tout  $j$ ,  $S_{n,j} = n\Delta S$ ,  $a_n = \frac{1}{2}\sigma^2 S_n^2$ ,  $b_n = rS_n$ ,  $c = -r$ . Le schéma est alors, pour tout  $n$  dans  $\llbracket 1, N_S - 1 \rrbracket$  et tout  $j$  dans

$\llbracket 0, N_t - 1 \rrbracket$ ,

$$\begin{aligned}
\frac{C_{n,j+1} - C_{n,j}}{\Delta t} = & \frac{1}{2} \left( a_n \frac{C_{n+1,j+1} - 2C_{n,j+1} + C_{n-1,j+1}}{(\Delta S)^2} \right. \\
& + b_n \frac{C_{n+1,j+1} - C_{n-1,j+1}}{2\Delta S} \\
& + c C_{n,j+1} \Big) \\
& + \frac{1}{2} \left( a_n \frac{C_{n+1,j} - 2C_{n,j} + C_{n-1,j}}{(\Delta S)^2} \right. \\
& + b_n \frac{C_{n+1,j} - C_{n-1,j}}{2\Delta S} \\
& + c C_{n,j} \Big)
\end{aligned} \tag{2}$$

On pose maintenant,  $\nu_1 = \frac{\Delta t}{(\Delta S)^2}$  et  $\nu_2 = \frac{\Delta t}{\Delta S}$ ,  $\alpha_n = \frac{1}{4}\nu_2 b_n - \frac{1}{2}\nu_1 a_n$ ,  $\beta_n = \nu_1 a_n - \frac{1}{2}\Delta t c$  et  $\gamma_n = -\frac{a_n}{2}\nu_1 - \frac{b_n}{4}\nu_2$ . En passant tout les termes en  $j+1$  d'un côté et tout les termes en  $j$  de l'autre on obtient les  $N_s - 1$  équations suivantes pour  $j$  fixé :

$$\alpha_n C_{n-1,j+1} + (1 + \beta_n) C_{n,j+1} + \gamma_n C_{n+1,j+1} = -\alpha_n C_{n-1,j} + (1 - \beta_n) C_{n,j} - \gamma_n C_{n+1,j}$$

On peut représenter ces équations sous forme matricielle :

$$\begin{pmatrix}
\alpha_1 & 1 + \beta_1 & \gamma_1 & 0 & \dots & \dots & 0 \\
0 & \alpha_2 & 1 + \beta_2 & \gamma_2 & 0 & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \dots & \dots & 0 & \alpha_{N_s-1} & 1 + \beta_{N_s-1} & \gamma_{N_s-1}
\end{pmatrix}
\begin{pmatrix}
C_{0,j+1} \\
C_{1,j+1} \\
\vdots \\
C_{N_s-1,j+1} \\
C_{N_s,j+1}
\end{pmatrix} =
\begin{pmatrix}
-\alpha_1 & 1 - \beta_1 & -\gamma_1 & 0 & \dots & \dots & 0 \\
0 & -\alpha_2 & 1 - \beta_2 & -\gamma_2 & 0 & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \dots & \dots & 0 & -\alpha_{N_s-1} & 1 - \beta_{N_s-1} & -\gamma_{N_s-1}
\end{pmatrix}
\begin{pmatrix}
C_{0,j} \\
C_{1,j} \\
\vdots \\
C_{N_s-1,j} \\
C_{N_s,j}
\end{pmatrix}$$

Ce système possède  $N_s - 1$  équations et  $N_s + 1$  inconnues. Grâce aux conditions aux bords de l'équation on peut retirer deux inconnues du système en le transformant au système équivalent suivant :

$$\begin{pmatrix} 1+\beta_1 & \gamma_1 & 0 & \dots & \dots & 0 \\ \alpha_2 & 1+\beta_2 & \gamma_2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \gamma_{N_S-2} \\ 0 & \dots & \dots & 0 & \alpha_{N_S-1} & 1+\beta_{N_S-1} \end{pmatrix} \begin{pmatrix} C_{1,j+1} \\ \vdots \\ \vdots \\ \vdots \\ C_{N_S-1,j+1} \end{pmatrix} + \begin{pmatrix} \alpha_1 C_{0,j+1} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \gamma_1 C_{N_S,j+1} \end{pmatrix} = \\
\begin{pmatrix} 1-\beta_1 & -\gamma_1 & 0 & \dots & \dots & 0 \\ -\alpha_2 & 1-\beta_2 & -\gamma_2 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & -\gamma_{N_S-2} \\ 0 & \dots & \dots & 0 & -\alpha_{N_S-1} & 1-\beta_{N_S-1} \end{pmatrix} \begin{pmatrix} C_{1,j} \\ \vdots \\ \vdots \\ \vdots \\ C_{N_S-1,j} \end{pmatrix} + \begin{pmatrix} -\alpha_1 C_{0,j} \\ 0 \\ \vdots \\ \vdots \\ 0 \\ -\gamma_1 C_{N_S,j} \end{pmatrix}$$

Ce qui est donc de la forme :

$$M_L^j P^{j+1} + R_L = M_R^j P^j + R_R$$

Ce qui est équivalent à :

$$M_L^j P^{j+1} = M_R^j P^j + R_R - R_L$$

Où le membre de droite est un vecteur colonne totalement connu par l'étape de calcul du  $P^j$  précédent. De plus  $M_L^j$  est une matrice tridiagonale à coefficients réels on peut donc appliquer une résolution de système rapide grâce à l'algorithme de Thomas qui évite l'inversion de matrice. Cet algorithme est de plus stable si la matrice est à diagonale dominante ou symétrique définie positive.

## 2 EDP Réduite : Théorie

### 2.1 Changement de variable

Pour transformer l'équation de Black-Scholes nous effectuons le changement de variable sur  $\tau$  et  $x$  suivant :  $t = T - \frac{2\tau}{\sigma^2}$   $S = Ke^x$   $C(t, S) = Kc(\tau, x)$

On a alors les relations suivantes :

$$\frac{\partial C}{\partial t}(t, S) = \frac{-K\sigma^2}{2} \frac{\partial c(\tau, x)}{\partial \tau}$$

$$\frac{\partial C}{\partial S}(t, S) = \frac{K}{S} \frac{\partial c(\tau, x)}{\partial x}$$

$$\frac{\partial^2 C}{\partial S^2}(t, S) = \frac{-K}{S^2} \frac{\partial c(\tau, x)}{\partial x} + \frac{K}{S^2} \frac{\partial^2 c(\tau, x)}{\partial x^2}$$

$c$  vérifie donc après injection dans l'équation de black-scholes :

$$\frac{\partial c}{\partial \tau} = \frac{\partial^2 c}{\partial x^2} + (k-1) \frac{\partial c}{\partial x} - kc \quad (3)$$

avec  $k = \frac{2r}{\sigma^2}$

On pose maintenant  $\tilde{S} = x$ ,  $\tilde{t} = \tau$ ,  $\alpha = -\frac{k-1}{2}$ ,  $\beta = \frac{-(k+1)^2}{4}$ ,  $\phi(\tilde{t}, \tilde{S}) = \exp(\alpha\tilde{S} + \beta\tilde{t})$  et la fonction  $\tilde{C}$  tel que  $c(\tilde{t}, \tilde{S}) = \phi(\tilde{t}, \tilde{S})\tilde{C}(\tilde{t}, \tilde{S})$ .

On a alors les relations suivantes :

$$\frac{\partial c}{\partial t}(\tilde{t}, \tilde{S}) = \beta\phi(\tilde{t}, \tilde{S})\tilde{C}(\tilde{t}, \tilde{S}) + \phi(\tilde{t}, \tilde{S}) \frac{\partial \tilde{C}}{\partial \tilde{t}}(\tilde{t}, \tilde{S})$$

$$\frac{\partial c}{\partial S}(\tilde{t}, \tilde{S}) = \alpha\phi(\tilde{t}, \tilde{S})\tilde{C}(\tilde{t}, \tilde{S}) + \phi(\tilde{t}, \tilde{S}) \frac{\partial \tilde{C}}{\partial \tilde{S}}(\tilde{t}, \tilde{S})$$

$$\frac{\partial^2 c}{\partial \tilde{S}^2}(\tilde{t}, \tilde{S}) = \alpha^2\phi(\tilde{t}, \tilde{S})\tilde{C}(\tilde{t}, \tilde{S}) + 2\alpha\phi(\tilde{t}, \tilde{S}) \frac{\partial \tilde{C}}{\partial \tilde{S}}(\tilde{t}, \tilde{S}) + \phi(\tilde{t}, \tilde{S}) \frac{\partial^2 \tilde{C}}{\partial \tilde{S}^2}(\tilde{t}, \tilde{S})$$

En injectant dans (3) ces relations et en simplifiant on obtient :

$$\frac{\partial \tilde{C}}{\partial \tilde{t}}(\tilde{t}, \tilde{S}) = \frac{\partial^2 \tilde{C}}{\partial \tilde{S}^2}(\tilde{t}, \tilde{S}) \quad (4)$$

pour  $\tilde{t} \in [0, \frac{\sigma^2 T}{2}]$  et  $\tilde{S} \in ]-\infty, \ln(\frac{L}{K})]$

Ce qui nous donne la relation entre  $C$  et  $\tilde{C}$  suivante :

$$\tilde{C}(\tilde{t}, \tilde{S}) = \frac{1}{K\phi(\tilde{t}, \tilde{S})} C(T - \frac{2\tilde{t}}{\sigma^2}, \exp(\tilde{S}))$$

Les conditions aux limites sont alors pour  $\tilde{t} \in [0, \frac{\sigma^2 T}{2}]$  et  $\tilde{S} \in ]-\infty, \ln(\frac{L}{K})]$  :

$$\tilde{C}(0, \tilde{S}) = \frac{1}{K} \exp(-\alpha\tilde{S}) C(T, \exp(\tilde{S}))$$

$$\tilde{C}(\tilde{t}, \ln(\frac{L}{K})) = \frac{1}{K} \exp(-\alpha \ln(\frac{L}{K}) - \beta\tilde{t}) C(T - \frac{2\tilde{t}}{\sigma^2}, L)$$

Nous n'avons ici plus un intervalle fini pour les valeurs de  $\tilde{S}$ , pour appliquer une méthode des différences finies il faut se placer sur un intervalle fini en choisissant une valeur  $\tilde{S}^-$  tel que  $\tilde{C}(\tilde{t}, -\infty) \approx \tilde{C}(\tilde{t}, \tilde{S}^-)$  et considérer la condition au bord en  $-\infty$  au point  $\tilde{S}^-$ . On prendra  $\tilde{S}^- = -5$ , en effet  $\exp(-5) \approx 0.007$  ce qui est relativement proche de 0 pour des valeurs usuelles de strike d'options de plus expérimentalement nous avons obtenus de bon résultats avec cette valeur.

La condition aux limites manquante est alors :

$$\tilde{C}(\tilde{t}, \tilde{S}^-) = \frac{1}{K} \exp(-\alpha\tilde{S}^- - \beta\tilde{t}) C(T - \frac{2\tilde{t}}{\sigma^2}, 0)$$

## 2.2 Représentation par les différences finies

Pour notre schéma implicite on discrétise l'espace  $[S^-, \ln(\frac{L}{K})]$  en  $N_s$  intervalles réguliers et  $[0, \frac{\sigma^2 T}{2}]$  en  $N_t$  intervalles réguliers aussi. On pose alors  $\Delta\tilde{S} = \frac{\ln(\frac{L}{K}) - S^-}{N_s}$  et  $\Delta\tilde{t} = \frac{\sigma^2 T}{2N_t}$ . On pose  $\tilde{S}_n = S^- + n\Delta S$  et  $\tilde{t}_j = j\Delta t$  pour  $n \in \llbracket 0, N_s \rrbracket$  et  $j \in \llbracket 0, N_t \rrbracket$ . On note  $\tilde{C}_{n,j} = \tilde{C}(\tilde{t}_j, \tilde{S}_n)$ . Avec le changement de variable on a pour tout  $S \in [0, L]$ ,

$$C(0, S) = K \exp(\alpha \ln(\frac{S}{K}) + \beta T \frac{\sigma^2}{2}) \tilde{C}(\frac{\sigma^2 T}{2}, \ln(\frac{S}{K}))$$

On cherche donc à calculer de proche en proche pour tout  $n \in \llbracket 0, N_s \rrbracket$ ,  $\tilde{C}_{n, N_t}$ . Les valeurs initialement connues sont  $\tilde{C}_{0,j}$ ,  $\tilde{C}_{N_s,j}$  et  $\tilde{C}_n, 0$ .

## 2.3 Schéma implicite

Le schéma aux différences finies implicites consiste à prendre les approximations suivantes pour les dérivées :

$$\begin{aligned} \frac{\partial \tilde{C}}{\partial \tilde{t}}(n\Delta S, j\Delta t) &\approx \frac{\tilde{C}_{n,j+1} - \tilde{C}_{n,j}}{\Delta t} \\ \frac{\partial^2 \tilde{C}}{\partial \tilde{C}^2}(n\Delta S, j\Delta t) &\approx \frac{\tilde{C}_{n+1,j+1} - 2\tilde{C}_{n,j+1} + \tilde{C}_{n-1,j+1}}{\Delta S^2} \end{aligned}$$

Pour tout  $n \in \llbracket 1, N_s - 1 \rrbracket$ . Ce qui nous donne le schéma implicite suivant :

$$\frac{\tilde{C}_{n,j+1} - \tilde{C}_{n,j}}{\Delta t} = \frac{\tilde{C}_{n+1,j+1} - 2\tilde{C}_{n,j+1} + \tilde{C}_{n-1,j+1}}{\Delta S^2}$$

On pose maintenant  $\lambda = \frac{\Delta t}{\Delta S^2}$  et en passant les termes en  $j+1$  du même côté de l'équation on obtient :

$$-\lambda \tilde{C}_{n+1,j+1} + (1 + 2\lambda) \tilde{C}_{n,j+1} - \lambda \tilde{C}_{n-1,j+1} = \tilde{C}_{n,j}$$

Ce qui donne sous forme matricielle :

$$\begin{pmatrix} -\lambda & 1 + 2\lambda & -\lambda & 0 & \dots & \dots & 0 \\ 0 & -\lambda & 1 + 2\lambda & -\lambda & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \dots & 0 \\ 0 & \dots & \dots & 0 & -\lambda & 1 + 2\lambda & -\lambda \end{pmatrix} \begin{pmatrix} \tilde{C}_{0,j+1} \\ \tilde{C}_{1,j+1} \\ \vdots \\ \tilde{C}_{N_s-1,j+1} \\ \tilde{C}_{N_s,j+1} \end{pmatrix} = \begin{pmatrix} \tilde{C}_{1,j} \\ \vdots \\ \vdots \\ \tilde{C}_{N_s-1,j} \end{pmatrix}$$

Encore une fois le système a plus d'inconnues que d'équations, il faut donc intégrer les conditions initiales avec le système équivalent suivant :

$$\begin{pmatrix} 1+2\lambda & -\lambda & 0 & \dots & 0 \\ -\lambda & 1+2\lambda & -\lambda & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\lambda \\ 0 & \dots & 0 & -\lambda & 1+2\lambda \end{pmatrix} \begin{pmatrix} \tilde{C}_{1,j+1} \\ \vdots \\ \vdots \\ \vdots \\ \tilde{C}_{N_s-1,j+1} \end{pmatrix} + \begin{pmatrix} -\lambda\tilde{C}_{0,j+1} \\ 0 \\ \vdots \\ 0 \\ -\lambda\tilde{C}_{N_s,j+1} \end{pmatrix} = \begin{pmatrix} \tilde{C}_{1,j} \\ \vdots \\ \vdots \\ \vdots \\ \tilde{C}_{N_s-1,j} \end{pmatrix}$$

Ce qui est donc de la forme :

$$M_L^j P^{j+1} + R_L = P^j$$

Ce qui est équivalent à :

$$M_L^j P^{j+1} = P^j - R_L$$

Où le membre de droite est un vecteur colonne totalement connu par l'étape de calcul du  $P^j$  précédent. De plus  $M_L^j$  est une matrice tridiagonale à coefficients réels on peut donc appliquer une résolution de système ici aussi avec l'algorithme de Thomas.

### 3 Implémentation en C++ : Structure des classes, Problèmes & Solutions

#### 3.1 Compilation

Le fichier qui utilise l'ensemble des classes que nous avons créé pour construire le vecteur  $C(0,.)$  est `main.cpp`. Nous avons donc créé un `Makefile` pour simplifier la compilation. Voici donc la procédure :

- 1) il faut taper `make` à la racine du dossier
- 2) aller dans le sous dossier `bin` et taper `./main`

#### 3.2 Conception des classes

Après notre travail de recherche sur la théorie mathématique autour de la discrétisation des équations à dérivées partielles, nous avons construit les différentes classes qui représenteront les différents objets qui interviendront dans la résolution de notre problème. Pour la résolution de ces équations différentielles, nous constatons qu'il y a des systèmes matricielles à résoudre mettant en jeu des matrices tri diagonales et des vecteurs colonnes. Nous avons donc créé une famille de classes pour les matrices, ainsi qu'une classe pour résoudre un système linéaire. Puis, comme indiqué dans l'énoncé, les conditions aux limites de la fonction  $C$  dépendent de la nature du Payoff. Nous avons donc créé des classes pour les deux types de Payoff à savoir le Call ou le Put. Enfin, nous avons créé une famille de classes pour les deux types d'équation différentielle que sont l'équation de Black Scholes et l'équation de la chaleur car même si elles ont la même structure, les matrices mises en jeu sont un peu différentes. De plus la méthode aux différences finies utilisée pour la résolution des deux équations est différentes. Nous avons pour cela créé une classe abstraite `DiffFinies` servant de base pour les deux méthodes Crank-nicholson et Implicit. Nous avons créé autant de classes afin de factoriser le code le plus possible.



Voici donc une description des différentes classes que nous avons implémenté :

- **BMatrix** : une base pour les matrices de taille  $n \times p$  avec toutes les opérations élémentaires
- **Tri\_Diag\_Matrix** : hérite de Bmatrix et représente les matrices tridiagonales
- **Col\_Vector** : hérite de Bmatrix et représente les vecteurs colonnes
- **Linear\_System** : permet de résoudre un système linéaire du type  $AX = B$  où  $A$  est une matrice tridiagonale et  $B$ , un vecteur colonne
- **Payoff** : représente les conditions initiale et de au bord d'un payoff
- **Call** : hérite de Payoff et représente un call
- **Put** : hérite de Payoff et représente un put
- **Diff\_finies** : représente la discrétisation en différences finies des équations aux dérivées partielles
- **Implicit** : hérite de Diff\_finies et représente la discrétisation pour l'équation de la chaleur
- **Cranck\_Nicholson** : hérite de Diff\_finies et représente la discrétisation pour l'équation de Black-Scholes

Voici donc le diagramme UML des différentes classes :

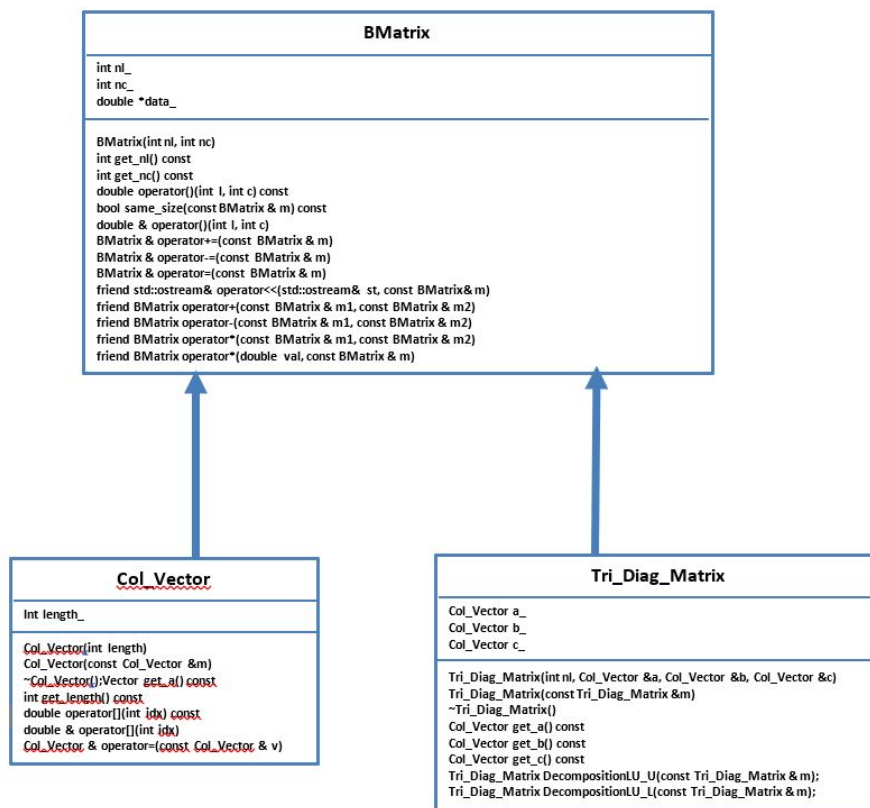


FIGURE 1 – Classes relatives aux matrices

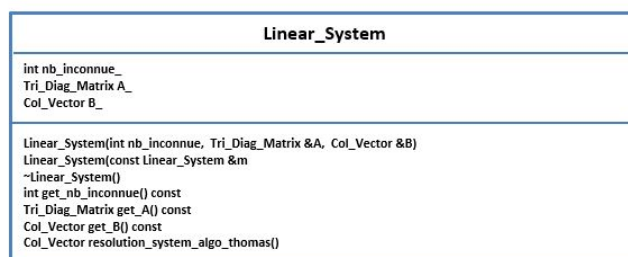


FIGURE 2 – Classes relatives au système linéaire

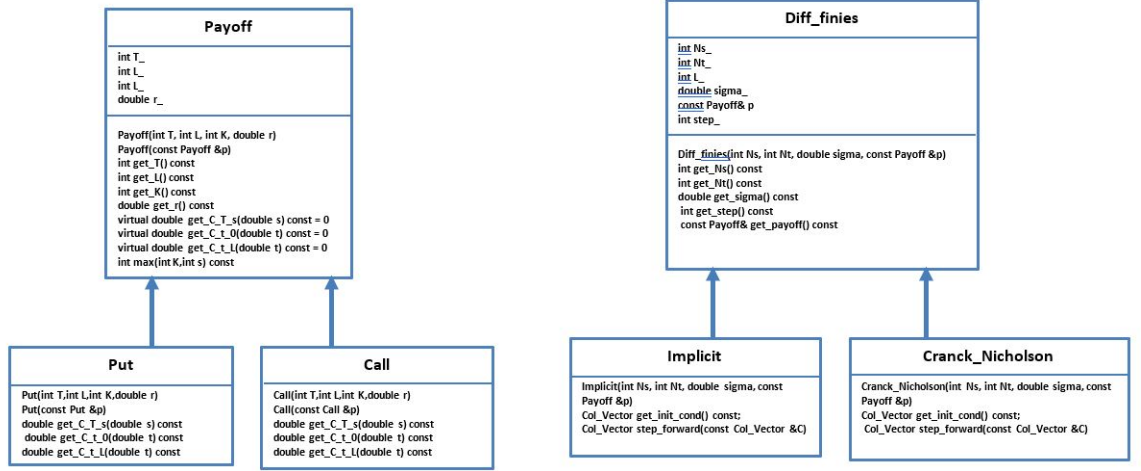


FIGURE 3 – Classes relatives aux Payoffs et à la discrétisation en différences finies

### 3.3 Problèmes & Solutions

#### 3.3.1 Problème de discrétisation

Le problème majeur de discrétisation concernait la discrétisation de l'espace  $]-\infty, \ln(\frac{L}{K})]$  car il n'est pas possible de discrétiser rigoureusement un tel intervalle. Nous avons donc du choisir une borne inférieure  $S^-$  pour simuler  $-\infty$ . Nous avons donc dû choisir une valeur de  $S^-$  telle que  $C$  ait des valeurs bien étalées sur tout l'intervalle après transformation. En effet, pour certains valeurs de  $S^-$  trop négative, le passage dans l'espace initial donne lieu à un calcul des valeurs de  $C$  en des points très concentrées autour de 0. Nous avons finalement expérimentalement choisit -5 comme valeur.

#### 3.3.2 Problème avec la résolution du système linéaire

Pour la résolution d'un système de type  $AX = B$ , nous avons d'abord choisi de résoudre ce système par une méthode LU. La matrice tridiagonale  $A$  respectait toutes les conditions pour la décomposer en la matrice LU. On aurait ainsi à résoudre un système triangulaire inférieur puis supérieur. Cette méthode est théoriquement assez efficace pour ce genre de problème. Néanmoins, nos matrices étaient de taille 1000 et il fallait résoudre 1000 systèmes linéaires. Or pour résoudre 1 système linéaire, il faut faire la décomposition LU de la matrice  $A$  et résoudre 2 sous systèmes. Ainsi avec cette technique les résolutions se font en  $O(n^3)$  et donc le calcul des valeurs de  $C$  se fait en  $O(n^4)$  avec  $n$  le nombre de segments de discrétisation (en supposant une discrétisation égale pour les 2 variables). Nous nous sommes rendus

compte de ce problème lorsque nous voulions afficher le vecteur  $C(0,s)$ . Il fallait attendre 4 minutes pour avoir un résultat de plus la méthode semblait donner expérimentalement des résultats instables. Nous avons donc cherché une méthode plus rapide et fiable pour résoudre un système tridiagonale. Nous avons donc choisi de nous tourner vers l'algorithme de Thomas qui est très adapté à ce genre de situation. Il nous a permis d'accélérer la résolution du système linéaire et surtout d'avoir une solution plus exacte , ce qui n'était pas le cas de la décomposition LU. Cette algorithme permet la résolution d'un système en  $O(n)$  et donc le calcul des valeurs de  $C$  se fait en  $O(n^2)$  ce qui reste plus rapide qu'une décomposition LU (même dans le cas de la résolution de plusieurs systèmes linéaires). Expérimentalement, avec l'algorithme de Thomas, nous obtenons le vecteur  $C(0,s)$  en 30 secondes , contre 4 minutes avec la décomposition LU. Cet algorithme est stable dans nos cas de figure, expérimentalement on observe que les matrices en jeu sont à diagonales dominantes.

## 4 Résultats : Affichage des courbes

Nous traçons pour les deux méthodes la courbe  $C(0,s)$  en fonction de  $s$ . Avec la méthode de Cranck-Nicholson , l'axe des abscisses est définie par  $X=\text{linspace}(0,L=300,1001)$  pour une discrétisation en 1000 de l'intervalle  $[0,300]$  Par contre, pour la méthode implicite nous avons discrétisé l'axe des abscisses d'une manière irrégulière car nous avons discrétiser de manière régulier les nouvelles variables après changement de variable. La transformation est non linéaire pour passer d'un espace à un autre ce qui à pour résultat de donner un maillage irrégulier. Nous avons défini l'axe des abscisses comme suit  $X'=\text{linspace}(S^-, \log(L/K),1001)$  et ensuite  $X = 100 * \exp(X')$  En effet, cela vient du fait que les deux vecteurs ne sont pas calculées avec exactement les mêmes points  $S_n$  De fait, nous n'avons pas trouvé un moyen pertinent de tracer le graphe des erreurs vu que les deux courbes n'ont pas été tracées exactement avec les mêmes valeurs de  $S_n$ .

## 4.1 Cas d'un Put

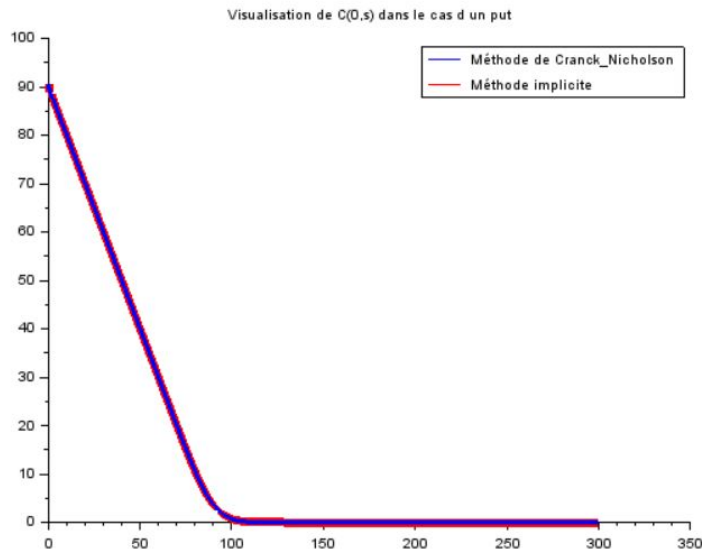


FIGURE 4 – Cas d'un put

Très clairement, la solution fournie par la méthode de Cranck-Nicholson (méthode 1) à partir de l'équation de Black-Scholes et la solution fournie par l'équation réduite en différences finies implicites (méthode 2) donnent des résultats très similaires. En effet, les courbes associées se superposent (fig.4). De plus, avec les deux méthodes, on a bien l'allure caractéristique d'un put et le "coude" aux alentours du strike (ici  $K=100$ ).

## 4.2 Cas d'un Call

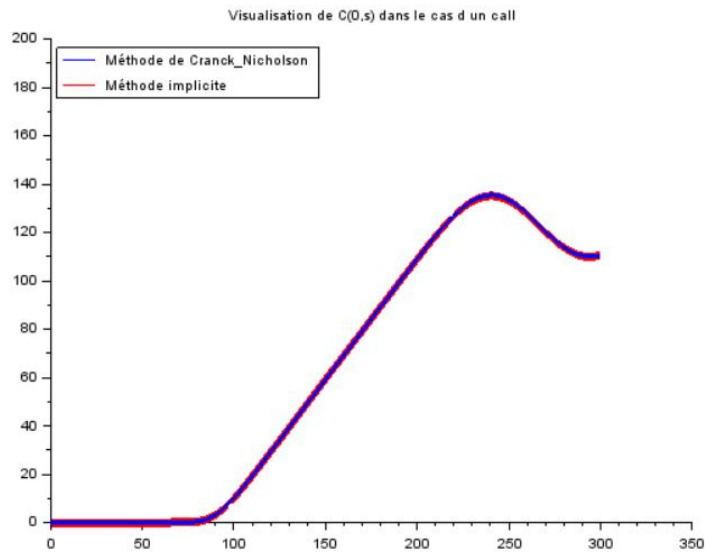


FIGURE 5 – Cas d'un call

Pour le call, on a la même situation que précédemment. Les solutions fournies par les deux méthodes concordent. On a bien l'allure générale d'un put et la présence du coude aux alentours de  $K$ .

## 5 Conclusion

Pour conclure, nos principaux problèmes d'implémentation concernent la discrétisation d'un intervalle de type  $] - \infty, \ln(L/K)]$  et la résolution d'un système matriciel de manière efficace. La deuxième difficulté a été surmontée grâce à l'algorithme de Thomas qui a donné des résultats plus que satisfaisants. Les résultats obtenus par la méthode de Cranck-Nicholson sur l'équation de Black Scholes et la méthode des différences finies sur l'équation de la chaleur sont très satisfaisants. On retrouve l'allure et les caractéristiques des différents payoffs. Expérimentalement on remarque que le calcul des valeurs par le passage de l'équation réduite est plus rapide que celui sur l'équation non transformée. Les résultats étant similaires pour les deux méthodes c'est cette méthode qui semble préférable.