Rapport Programmation Avancée & Projet

Équation de Black et Scholes

Adib HABBOU - Alae KHIDOUR

Table des matières

1 Étude théorique

1.1 Équation de Black et Scholes

Le modèle de Black et Scholes est basé sur l'hypothèse que les marchés financiers sont efficients et que les prix des actifs reflètent toutes les informations disponibles sur ces actifs. Selon cette hypothèse, il est impossible pour un investisseur de réaliser des profits à long terme en agissant sur le marché, à moins de prendre des risques supplémentaires.

Le modèle de Black et Scholes utilise également l'hypothèse que les taux d'intérêt sans risque sont constants et connus à l'avance. Il suppose également que la volatilité de l'actif sous-jacent est constante et connue à l'avance (la volatilité mesure la fluctuation des prix de l'actif sous-jacent).

En résumé, le modèle de Black et Scholes est un modèle mathématique utilisé pour déterminer le prix d'une option en prenant en compte les taux d'intérêt sans risque, la volatilité de l'actif sous-jacent et la date d'expiration de l'option. Il a été largement utilisé dans l'industrie financière pour la tarification des options depuis sa création en 1973 par Fischer Black et Myron Scholes.

L'équation complète de Black et Scholes permet de déterminer le prix d'une option en prenant en compte tous les paramètres qui peuvent influencer son prix. L'équation réduite de Black et Scholes, quant à elle, est une approximation de l'équation complète qui permet de simplifier les calculs en négligeant certains paramètres moins importants.

1.1.1 EDP Complète

La fonction $C(t, S)$ définie sur $[0, T] \times [0, L]$ est une approximation du prix d'une option qui dépend du temps et du prix de l'actif. Elle vérifie l'EDP suivante :

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

Sachant que :

- T : temps terminal
- L : valeur terminal
- r : taux d'intérêt du marché
- σ : volatilité de l'actif

1.1.2 EDP Réduite

L'équation aux dérivées partielles complète de Black et Scholes peut se transformer à l'aide de changements de variables en une équation plus simple dîtes EDP réduite :

$$\frac{\partial \widetilde{C}}{\partial \widetilde{t}} = \mu \frac{\partial^2 \widetilde{C}}{\partial \widetilde{S}^2} \tag{2}$$

La solution de l'EDP réduite est alors une approximation de C avec $\mu \in \mathbb{R}$.

1.1.3 Détermination du facteur μ

On considère l'EDP complète de Black et Scholes :

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

Elle peut également s'écrire sous la forme :

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

On effectue les changements de variables suivants :

$$S = exp(\widetilde{x}) \quad \text{et} \quad t = T - \widetilde{t}$$

On obtient alors l'EDP suivante :

$$-\frac{\partial C}{\partial \widetilde{t}} + \frac{1}{2}\sigma^2 \frac{\partial^2 C}{\partial \widetilde{x}^2} + (r - \frac{1}{2}\sigma^2)\frac{\partial C}{\partial \widetilde{x}} - rC = 0$$

On pose ensuite :

$$\widetilde{C} = exp(r\widetilde{t})C$$

On obtient alors l'équation :

$$\frac{\partial \widetilde{C}}{\partial \widetilde{t}} - \frac{1}{2}\sigma^2 \frac{\partial^2 \widetilde{C}}{\partial \widetilde{x}^2} - (r - \frac{1}{2}\sigma^2)\frac{\partial \widetilde{C}}{\partial \widetilde{x}} = 0$$

On pose finalement :

$$\widetilde{S} = \widetilde{x} + (r - \frac{1}{2}\sigma^2)\widetilde{t}$$

On trouve l'équation :

$$\frac{\partial \widetilde{C}}{\partial \widetilde{t}} + (r - \frac{1}{2}\sigma^2)\frac{\partial \widetilde{C}}{\partial \widetilde{S}} - \frac{1}{2}\sigma^2 \frac{\partial^2 \widetilde{C}}{\partial \widetilde{S}^2} - (r - \frac{1}{2}\sigma^2)\frac{\partial \widetilde{C}}{\partial \widetilde{S}} = 0$$

On conclut que l'EDP réduite s'écrit :

$$\frac{\partial \widetilde{C}}{\partial \widetilde{t}} = \frac{1}{2}\sigma^2 \frac{\partial^2 \widetilde{C}}{\partial \widetilde{S}^2}$$

On conclut donc que :

$$\boxed{\mu = \frac{1}{2}\sigma^2} \tag{3}$$

Il faut toutefois garder en tête que cette valeur dépend des changements de variables effectués pour trouver l'EDP réduite à partir de l'EDP complète.

1.2 Méthodes de résolution

1.2.1 Discrétisation temporelle et spatiale

Pour réaliser la résolution numérique des EDP on procède tout d'abord à la discrétisation temporelle et spatiale :

— $[0, T]$ en M intervalles $[t_i, t_{i+1}]$ tel que $\Delta t = t_{i+1} - t_i$

— $[0, L]$ en N intervalles $[j, s_{j+1}]$ tel que $\Delta S = s_{j+1} - s_j$

On note choisit comme notation :

$$C(t_i, s_j) = C_j^i$$

On procède ensuite à la discrétisation des dérivées partielles :

- Dérivée première par rapport au temps :

$$\boxed{\frac{\partial C(t_i, j)}{\partial t} \approx \frac{1}{\Delta t}(C_j^{i+1} - C_j^i)} \tag{4}$$

- Dérivée première par rapport à la valeur de l'actif :

$$\boxed{\frac{\partial C(t_i, j)}{\partial S} \approx \frac{1}{2\Delta S}(C_{j+1}^i - C_{j-1}^i)} \tag{5}$$

- Dérivée seconde par rapport à la valeur de l'actif :

$$\frac{\partial^2 C(t_i, j)}{\partial S^2} \approx \frac{\frac{C_{j+1}^i - C_j^i}{\Delta S} - \frac{C_j^i - C_{j-1}^i}{\Delta S}}{\Delta S} = \frac{C_{j+1}^i - 2C_j^i + C_{j-1}^i}{(\Delta S)^2}$$

$$\boxed{\frac{\partial^2 C(t_i, j)}{\partial S^2} \approx \frac{1}{(\Delta S)^2}(C_{j+1}^i - 2C_j^i + C_{j-1}^i)} \tag{6}$$

1.2.2 Résolution d'un système tridiagonale

Pour résoudre un système matriciel $Ax = b$ telle que A est une matrice tridiagonale, il est judicieux d'effectuer une décomposition LU pour résoudre le système.

En décomposant notre matrice : $A = LU$ avec L matrice triangulaire inférieure à diagonale unitaire et U matrice triangulaire supérieure. Lorsqu'on souhaite résoudre le système linéaire :

$$LUx = b$$

Il suffit de résoudre d'abord le système suivant :

$$LK = b$$

Et ensuite de résoudre le système :

$$Ux = K$$

Par soucis d'efficacité on va utiliser l'**algorithme de Thomas** qui est basé sur une décomposition LU, et qui permet de résoudre des systèmes tridiagonaux de manière efficace.

1.2.3 Crank-Nicholson pour l'EDP Complète

On a d'après ce qui précède l'EDP complète suivante :

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

On remplace dans l'EDP complète :

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

$$\Leftrightarrow \frac{1}{\Delta t}C_j^{i+1} + [-\frac{1}{\Delta t} - \frac{\sigma^2 j^2}{\Delta S^2} - r]C_j^i + [\frac{rj}{2\Delta S} + \frac{\sigma^2 j}{2\Delta S^2}]C_{j+1}^i + [-\frac{rj}{2\Delta S} + \frac{\sigma^2 j}{2\Delta S^2}]C_{j-1}^i = 0$$

$$\Leftrightarrow C_j^{i+1} = \Delta t\,[\underbrace{(\frac{1}{\Delta t} + \frac{\sigma^2 j^2}{\Delta S^2} + r)}_{y_j} C_j^i - \underbrace{(\frac{rj}{2\Delta S} + \frac{\sigma^2 j}{2\Delta S^2})}_{z_j} C_{j+1}^i + \underbrace{(\frac{rj}{2\Delta S} - \frac{\sigma^2 j}{2\Delta S^2})}_{x_j} C_{j-1}^i]$$

On pose les coefficients suivants :

$$y_j = \frac{1}{\Delta t} + \frac{\sigma^2 j^2}{\Delta S^2} + r \tag{7}$$

$$z_j = -(\frac{rj}{2\Delta S} + \frac{\sigma^2 j}{2\Delta S^2}) = -\frac{j}{2\Delta S}(r + \frac{\sigma^2}{\Delta S}) \tag{8}$$

$$x_j = \frac{rj}{2\Delta S} - \frac{\sigma^2 j}{2\Delta S^2} = \frac{j}{2\Delta S}(r - \frac{\sigma^2}{\Delta S}) \tag{9}$$

On obtient donc l'expression simplifiée suivante :

$$C_j^{i+1} = \Delta t(y_j C_j^i + z_j C_{j+1}^i + x_j C_{j-1}^i)$$

On peut transformer cette écriture en notation matricielle :

$$\mathbf{C}^{i+1} = P_1 \mathbf{C}^i \quad \text{avec} \quad P_1 = \Delta t \begin{bmatrix} y_1 & z_1 & 0 & \dots & \dots & 0 \\ x_2 & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & z_{N-2} \\ 0 & \dots & \dots & 0 & x_{N-1} & y_{N-1} \end{bmatrix}$$

1.2.4 Différences finies implicites pour l'EDP Réduite

On a d'après ce qui précède l'EDP réduite suivante :

$$\frac{\partial \widetilde{C}}{\partial \widetilde{t}} = \frac{1}{2}\sigma^2 \frac{\partial^2 \widetilde{C}}{\partial \widetilde{S}^2}$$

On remplace dans l'EDP réduite :

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

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

$$\Leftrightarrow C_j^{i+1} = \underbrace{\frac{\sigma^2 \Delta t}{2\Delta S^2}}_{\lambda} C_{j+1}^i + (1 - \underbrace{\frac{\sigma^2 \Delta t}{\Delta S^2}}_{2\lambda}) C_j^i + \underbrace{\frac{\sigma^2 \Delta t}{2\Delta S^2}}_{\lambda} C_{j-1}^i$$

Pour simplifier l'écriture en pose :

$$\lambda = \frac{\sigma^2 \Delta t}{2\Delta S^2} \tag{10}$$

On obtient donc l'expression simplifiée suivante :

$$C_j^{i+1} = \lambda C_j^i + (1 - 2\lambda) C_{j+1}^i + \lambda C_{j-1}^i$$

On peut transformer l'écriture en notation matricielle :

$$\mathbf{C}^{i+1} = P_2 \mathbf{C}^i \quad \text{avec} \quad P_2 = \begin{bmatrix} (1-2\lambda) & \lambda & 0 & \dots & \dots & 0 \\ \lambda & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \lambda \\ 0 & \dots & \dots & 0 & \lambda & (1-2\lambda) \end{bmatrix}$$

2 Implémentation : choix, difficultés et améliorations

2.1 Présentation des classes

2.1.1 Classes pour les options

La classe `Option` est une classe abstraite qui sert de base pour les classes concrètes `Put` et `Call`. La classe `Option` contient les attributs suivants :

- `K_` : Strike de l'option
- `T_` : Temps terminal de l'option
- `L_` : Valeur terminal de l'option
- `r_` : Taux d'intérêt du marché
- `sigma_` : Volatilité de l'actif

Elle possède également des getters pour chacun de ses attributs et une méthode virtuelle pure `payoff` qui retourne le payoff de l'option pour une valeur donnée de l'actif et du temps, elle va servir à modéliser les conditions initiales et les conditions aux bords. Les classes `Put` et `Call` héritent de `Option` et implémentent la méthode virtuelle pure `payoff`.

2.1.2 Classes pour les équations aux dérivées partielles

La classe EDP est une classe abstraite qui sert de base pour les classes concrètes EDPComplete et EDPReduite.

La classe `EDP` est une classe abstraite qui représente une équation différentielle aux dérivées partielles pour une option. Elle possède un seul attribut : `option_`, qui est une référence constante vers l'option associée à l'équation aux dérivées partielles. Elle possède également un constructeur qui initialise cet attribut avec la valeur passée en paramètre, ainsi qu'un getter qui permet de récupérer la valeur de l'attribut.

Les classes `EDPComplete` et `EDPReduite` héritent de `EDP` et possèdent chacune un constructeur qui appelle le constructeur de la classe `EDP` en passant une option en paramètre. Ces classes concrètes permettent de représenter deux types différents d'équation aux dérivées partielles, qui sont l'équation aux dérivées partielles complète et l'équation aux dérivées partielles réduite de Black et Scholes.

2.1.3 Classes pour les Différences Finies

La classe `DifferencesFinies` est une classe abstraite qui sert de base pour les classes concrètes `CrankNicholson` et `Implicite`. On retrouve également la déclaration de la fonction `algoThomas`, qui permet de résoudre un système linéaire de la forme $Ax = b$ en utilisant une décomposition LU de la matrice tridiagonale A.

La classe `DifferencesFinies` est une classe abstraite qui représente une méthode des différences finies pour résoudre une équation différentielle aux dérivées partielles. Elle possède plusieurs attributs :

- `edp_` : Référence vers l'objet EDP à résoudre
- `M_` : Nombre de pas de temps
- `N_` : Nombre de pas d'espace
- `dt_` : Pas de temps
- `dS_` : Pas d'espace
- `t_` : Valeurs de temps t pour lesquelles on calcule la solution
- `S_` : Valeurs de l'actif S pour lesquelles on calcule la solution

Elle possède également un constructeur qui initialise ses attributs et trois getters qui permettent de récupérer la valeur de ces attributs.

Les classes `CrankNicholson` et `Implicite` héritent de `DifferencesFinies` et implémente la méthode de Crank-Nicholson pour résoudre une équation aux dérivées partielles complète et la méthode Implicite pour résoudre une équation aux dérivées partielles réduite.

2.1.4 Classes pour l'affichage

La classe `Sdl` contient des méthodes permettant de dessiner des courbes dans une fenêtre SDL *(Simple DirectMedia Layer)*, qui est une bibliothèque logicielle permettant de faire de la programmation multimédia.

La classe `Sdl` contient deux attributs : `renderer_` et `window_`, qui sont respectivement un pointeur vers un renderer SDL et une fenêtre SDL. Elle possède également deux constructeurs : un constructeur par défaut et un constructeur avec deux paramètres en plus d'un destructeur.

La classe contient également deux méthodes publiques : `draw_curve` et `show`. La première permet de dessiner une courbe dans la fenêtre en utilisant le renderer SDL, et la seconde affiche la fenêtre.

On définit aussi la fonction `init_window` qui permet d'initialiser une fenêtre SDL, la fonction `init_renderer` qui permet d'initialiser un renderer SDL et la fonction `cleanup` qui permet de détruire un window et un renderer.

2.2 Problèmes rencontrés

De nombreux problèmes et difficultés ont été rencontrés lors de la réalisation du projet parmi lesquelles :

- *Résolution d'un système linéaire avec la décomposition LU :* au départ on effectuait une décomposition LU classique et on stockait donc tous les éléments même nuls des matrices $P1$ et $P2$, au final on a décidé d'utiliser l'algorithme de Thomas pour ne stocker que les éléments non nuls des matrices et gagner en temps d'exécution ;

- *Méthodes solve de Crank Nicholson et Implicite :* réussir à implémenter les calculs théoriques précédents en définissant correctement les vecteurs de coefficients des matrices $P1$ et $P2$. Puis réussir également à appliquer correctement la résolution avec les bons indices et les bonnes conditions aux bords et terminales au sein de la matrice C pour éviter d'avoir des `-inf`, des `nan` ou encore des valeurs aberrantes ;

- *Affichage des résultats avec SDL :* réussir à jongler entre les différentes fenêtres et les différentes courbes de manière à pouvoir afficher deux courbes sur une fenêtre, ajouter un cadre blanc et également pouvoir fermer les 4 fenêtres.

2.3 Améliorations possibles

Le projet pourrait être amélioré en effectuant les modifications suivantes :

- *Améliorer la précision des solutions :* utiliser des discrétisations plus précises pour les dérivées spatiales premières et secondes :

$$\frac{\partial C(t_i, j)}{\partial S} \approx \frac{1}{4\Delta S}(C_{j+1}^{i+1} - C_{j-1}^{i+1} + C_{j+1}^{i} - C_{j-1}^{i})$$

$$\frac{\partial^2 C(t_i, j)}{\partial S^2} \approx \frac{1}{2(\Delta S)^2}(C_{j+1}^{i+1} - 2C_{j}^{i+1} + C_{j-1}^{i+1} + C_{j+1}^{i} - 2C_{j}^{i} + C_{j-1}^{i})$$

- *Améliorer l'affichage SDL :* ajouter une légende qui associe le nom de chaque courbe à sa couleur pour pouvoir les différencier et ajouter également les axes des abscisses et des ordonnées à la place du cadre blanc.

3 Conclusion : résultats et interprétation

3.1 Résultats pour un Put

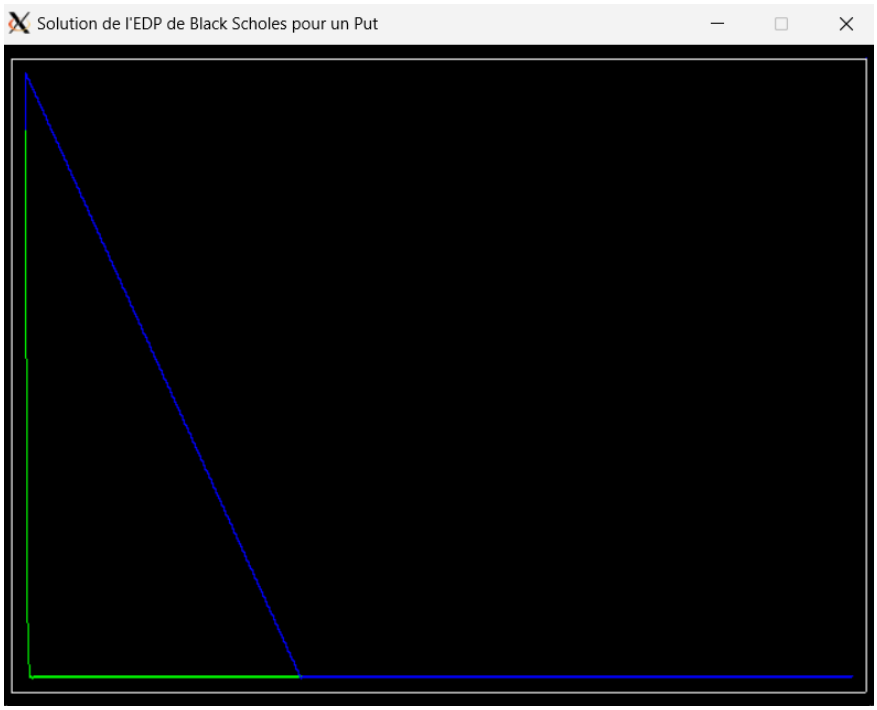

FIGURE 1 – Solution de l'EDP de Black Scholes pour un Put

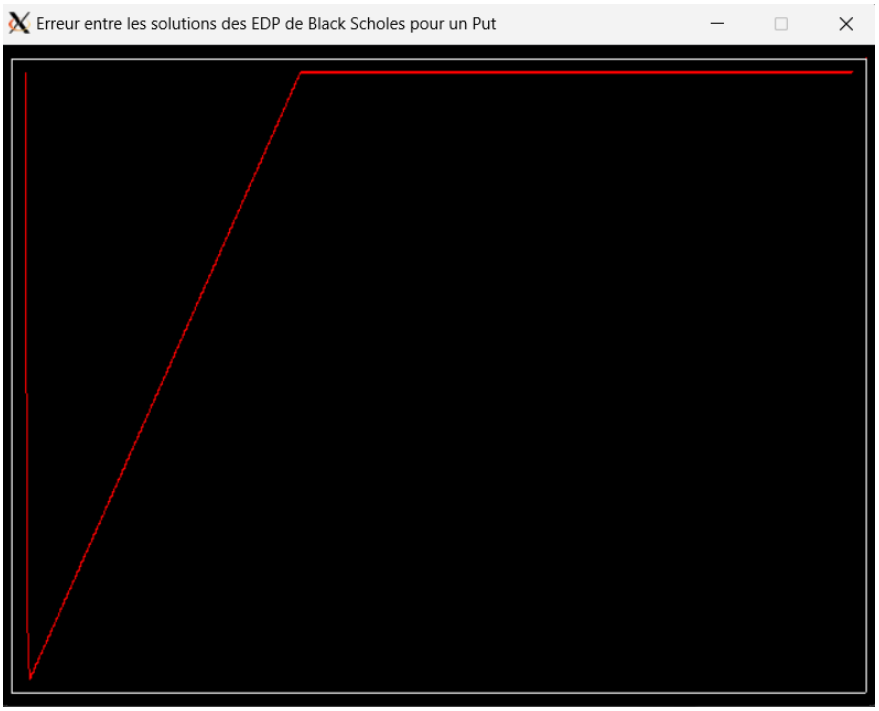

FIGURE 2 – Erreur entre les solutions des EDP de Black Scholes pour un Put

3.2 Résultats pour un Call

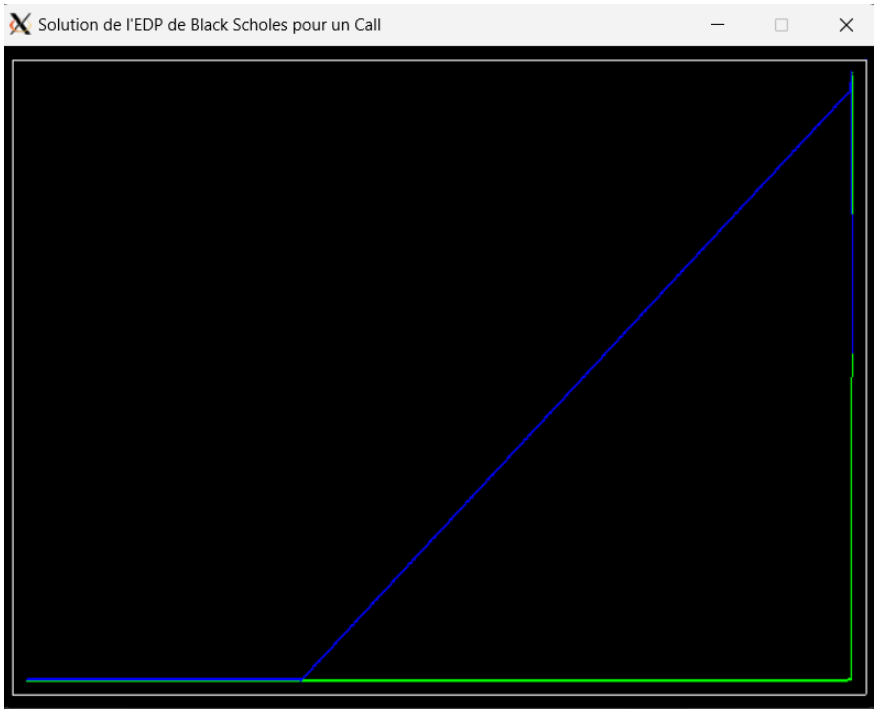

FIGURE 3 – Solution de l'EDP de Black Scholes pour un Call

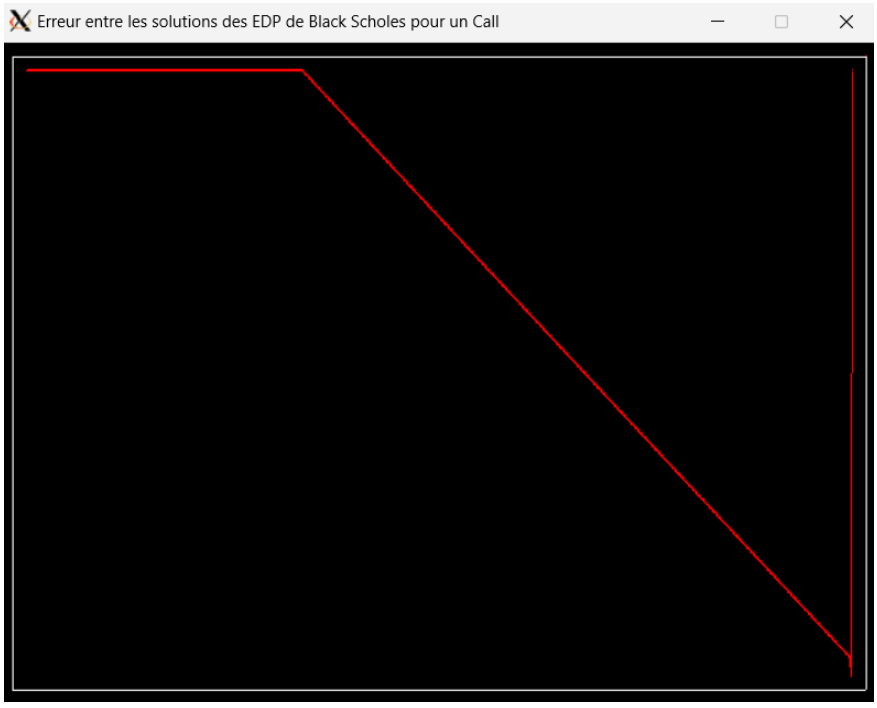

FIGURE 4 – Erreur entre les solutions des EDP de Black Scholes pour un Call

3.3 Interprétation des résultats

Il est important de garder en tête que les conditions aux bords modélisent le fait que :

- Pour les options de vente (Put), une option a une valeur intrinsèque positive si le prix de l'actif sous-jacent est inférieur au prix d'exercice de l'option (Strike). Dans ce cas, l'acheteur de l'option peut décider d'exercer son option et de vendre l'actif sous-jacent au prix d'exercice, ce qui lui permet de réaliser un bénéfice. Si le prix de l'actif sous-jacent est supérieur au prix d'exercice de l'option, l'option a une valeur intrinsèque négative et ne peut pas être utilisée pour réaliser un bénéfice ;

- Pour les options d'achat (Call), une option a une valeur intrinsèque positive si le prix de l'actif sous-jacent est supérieur au prix d'exercice de l'option (Strike). Dans ce cas, l'acheteur de l'option peut décider d'exercer son option et d'acheter l'actif sous-jacent au prix d'exercice, ce qui lui permet de réaliser un bénéfice. Si le prix de l'actif sous-jacent est inférieur au prix d'exercice de l'option, l'option a une valeur intrinsèque négative et ne peut pas être utilisée pour réaliser un bénéfice.

Les deux solutions obtenues pour l'EDP Complète et l'EDP Réduite pour un Put décroît vers 0 à mesure que le temps de maturité de l'option se rapproche. Cela se produit parce que, à mesure que le temps de maturité de l'option se rapproche, l'incertitude sur le prix de l'actif sous-jacent diminue, ce qui réduit la valeur de l'option. Si l'option n'est pas exercée avant son échéance, elle expirera sans valeur. C'est pourquoi les solutions pour un Put décrivent une courbe qui décroît rapidement vers zéro à mesure que le temps de maturité de l'option se rapproche.

Les deux solutions obtenues pour l'EDP Complète et l'EDP Réduite pour un Call sont nulles au début, car la plupart du temps, les options ont une valeur intrinsèque négative. Cependant, à mesure que le temps de maturité de l'option se rapproche, la valeur intrinsèque de l'option peut devenir positive si le prix de l'actif sous-jacent augmente au-dessus du prix d'exercice de l'option (Strike). Dans ce cas, la valeur de l'option commence à croître rapidement à mesure que le temps de maturité de l'option se rapproche, car elle reflète la valeur de l'actif sous-jacent au-dessus du prix d'exercice de l'option.

L'erreur relative entre la solution complète et la solution réduite de l'équation de Black-Scholes pour un Put est nulle au début et tend vers une valeur élevée à mesure que le temps de maturité de l'option se rapproche.

L'erreur relative entre la solution complète et la solution réduite de l'équation de Black et Scholes pour un Call est assez élevée au début et tend vers 0 à mesure que le temps de maturité de l'option se rapproche.

Cela se produit parce que la solution réduite de l'équation de Black et Scholes est une approximation de la solution complète et que cette approximation devient de plus en plus précise à mesure que le temps de maturité de l'option se rapproche ou s'éloigne suivant si l'option est un Put ou un Call.

4 Annexes

4.1 Diagramme des classes

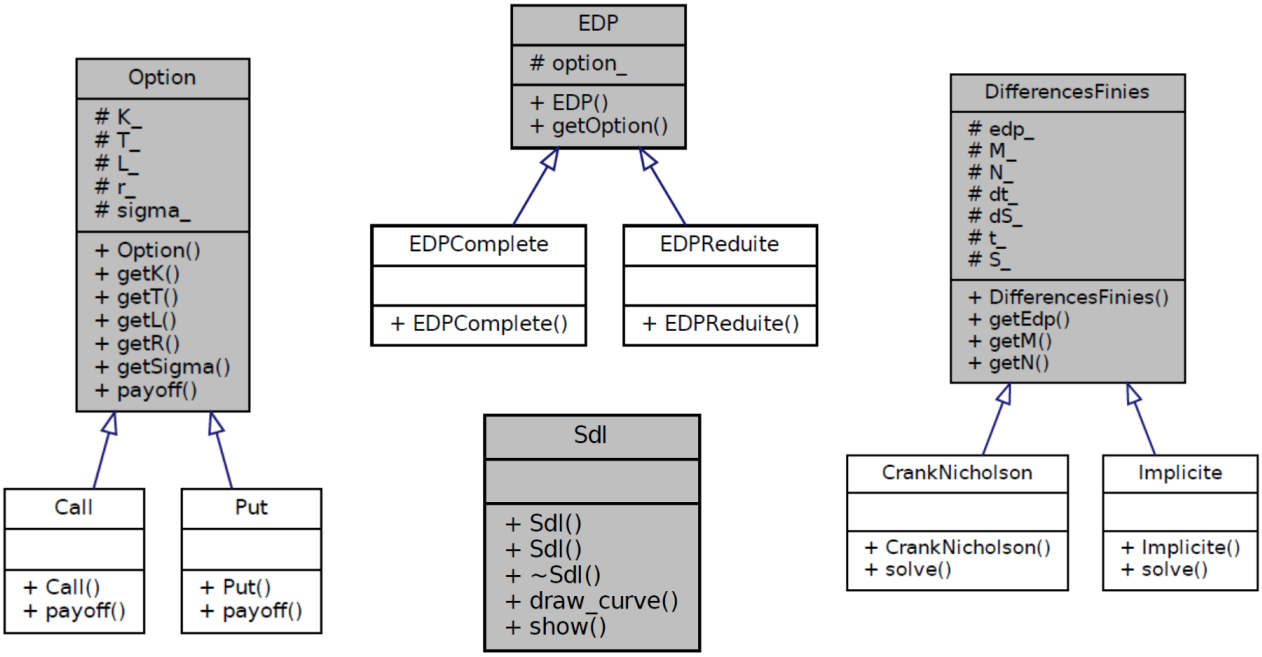

FIGURE 5 – Diagramme des classes

4.2 Diagramme des inclusions

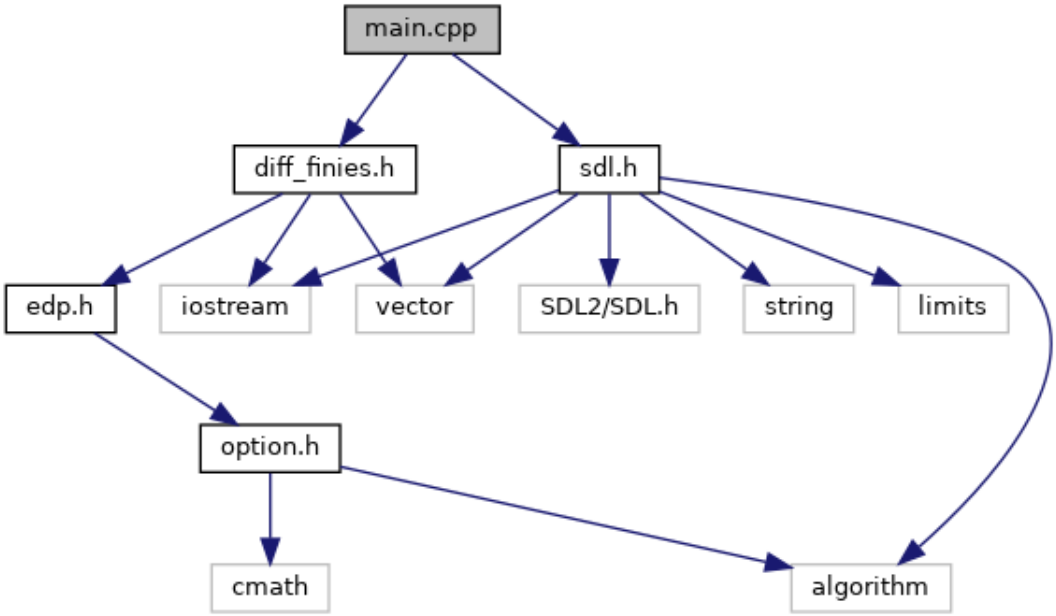

FIGURE 6 – Diagramme des inclusions