



Pricing par EDP

dans le modèle de Black-Scholes

Yacine Falaki Aeson Feehan

Table des matières

1	Le modèle de Black-Scholes	3
1.1	L'équation de Black-Scholes	3
1.2	Discrétisation de l'EDP	4
1.2.1	Notations	4
1.2.2	Expression de l'équation discrétisée	5
1.2.3	Expression matricielle	5
1.3	Résolution	6
1.4	Méthode alternative (M2)	6
2	L'implémentation	7
2.1	Architecture générale	7
2.2	Cas de la différence finie	7
2.3	Affichage avec Qt	9
2.3.1	Choix de la librairie	9
2.3.2	Méthodes d'affichage	9
3	Bilan	11
3.1	Les limites de l'implémentation	11
3.1.1	Différences finies et inversion approchée	11
3.1.2	Solutions	12
3.2	Extensions possibles	12
4	Bibliographie	14

Introduction

Lorsqu'on échange sur un marché financier, il est important de connaître la vraie valeur des produits échangés. Or la valeur peut différer du prix proposé : il faut créer des modèles de *pricing*, les plus sophistiqués possibles, pour renseigner chaque décision et sortir du marché en ayant fait un profit.

Depuis sa mise-au-point dans les années 1970, le modèle de Black-Scholes a servi de référence de la finance de marché moderne. Avec le modèle binomial, il constitue le fondement des modèles de *pricing* des options sur actions. Deux exemples fondamentaux sont les **calls** et les **puts**, ie. les droits d'acheter (*call*) ou de vendre (*put*) un **produit financier sous-jacent**, comme une action. Le modèle s'appuie sur six paramètres et permet de déterminer un unique prix pour un produit donné, éliminant le risque théorique de ceux-ci via une révision continue d'un **portefeuille** théorique de produits bien choisis.

Or le modèle de Black-Scholes est déduit d'une équation aux dérivées partielles (EDP) appelée **Équation de Black-Scholes** (voir 1.1). Comme la plupart des EDP, l'équation de Black-Scholes est difficile à résoudre explicitement. On verra qu'on peut parfois en trouver une forme explicite, mais on ne le peut pas toujours et probablement pas aussi rapidement qu'un ordinateur. *Comment instrumentaliser le modèle de Black-Scholes pour établir une stratégie de trading?* La réponse à cette question a un intérêt à la fois pratique (obtenir de l'argent) et théorique : Black et Scholes eux-mêmes ont implémenté leur propre modèle sur un vrai marché financier, mais ils ont fait des pertes importantes. Après avoir présenté le modèle en détail (1.), nous implémentons un algorithme en langage C++ pour approximer le prix d'un dérivé quelconque et afficher les résultats (2.). Enfin, on distingue certaines pistes d'amélioration de l'implémentation.

1 Le modèle de Black-Scholes

1.1 L'équation de Black-Scholes

L'équation de Black-Scholes est le fondement de tout ce qui suivra. Elle a été établie à la fin des années 1960 et est présente dans beaucoup de stratégies de *trading*, souvent avec quelques modifications. Elle décrit l'évolution du prix d'un produit **dérivé**, c'est-à-dire un produit financier dont la valeur dérive de la valeur d'un **sous-jacent**, comme la valeur de l'or ou d'une action chez une entreprise.

Considérons les quantités

$$\theta := \frac{\partial V}{\partial t} \quad \gamma := \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}$$

Notations :

1. V est la valeur de l'option dont la valeur nous intéresse,
2. S est le prix du sous-jacent,
3. $t \geq 0$ désigne le temps,
4. $\sigma \geq 0$ est la volatilité du prix du sous-jacent,
5. $r \in \mathbb{R}$ est la tendance.

Dans le cas des options, $\theta < 0$: moins il reste du temps pour exercer l'option, moins elle peut prendre de la valeur avant qu'on ne s'en serve et donc moins elle est intéressante. Au contraire, $\gamma > 0$: la valeur du dérivé est convexe en la valeur du sous-jacent. C'est cette caractéristique qui rend la détention d'une option intéressante.

D'autre part, considérons le portefeuille P composé des deux produits suivants :

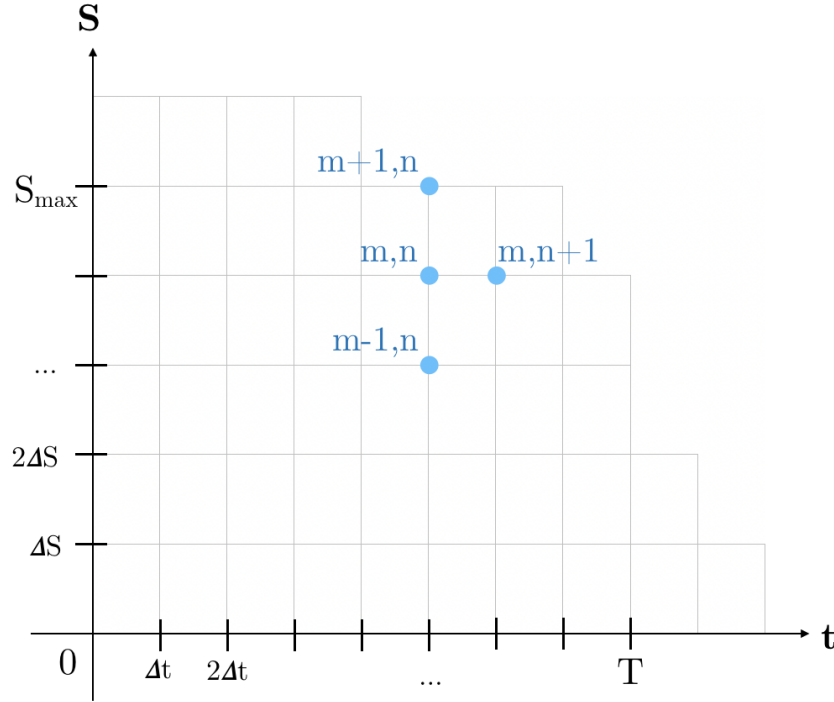
1. Une quantité r du dérivé, où r est la tendance de l'évolution de la valeur du dérivé.
2. Une quantité $-r \times \frac{\partial V}{\partial S}$ du sous-jacent.

L'intuition de Black-Scholes est que ce portefeuille sans risque permet de compenser parfaitement les variations dans la valeur de la dérivée en question. Cela s'écrit

$$\begin{aligned} \theta + \gamma &= P \\ \theta + \gamma &= rV - rS \frac{\partial V}{\partial S} \\ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} &= -rV + rS \frac{\partial V}{\partial S} \end{aligned}$$

d'où la forme usuelle de l'équation

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

FIGURE 1 – discrétisation du plan (t, S)

C'est cette forme de l'équation qu'on utilisera pour déterminer l'évolution du prix d'un produit dérivé dans la suite de cette étude.

Remarque : on a fait l'hypothèse que la volatilité σ est constante dans le temps. En réalité, on se doute que celle-ci varie à long terme, suivant l'information et les variations historiques des prix de marché des différents produits concernés (sous-jacent et dérivé), par exemple.

1.2 Discrétisation de l'EDP

1.2.1 Notations

On applique ici la méthode dite des **différences finies** (méthode **M1**) : le temps t et le prix du sous-jacent S sont discrétisés avec

1. $N + 1$ points régulièrement espacés $(t_n)_{0 \leq n \leq N}$ avec $t_0 = 0$ et $t_N = T$ (maturité du produit dérivé) ;
2. $M + 1$ points régulièrement espacés $(S_m)_{0 \leq m \leq M}$ avec $S_0 = 0$ et $S_M = S_{\max}$ (valeur du sous-jacent à maturité du dérivé). Dans l'implémentation, on fixe $S_{\max} = 1$ sans perdre de généralité.

La valorisation du dérivé en chaque point (t_n, S_m) du plan (t, S) est donnée par

$$V_{n,m} = V(n\Delta t, m\Delta S) \quad \Delta t = \frac{T}{N} \quad \Delta S = \frac{S}{M}$$

1.2.2 Expression de l'équation discrétisée

Tout d'abord, pour approximer $\frac{\partial V}{\partial S}$, il y a deux principales méthodes : celles d'Euler *implicite* et *explicite*. Ces approximations sont respectivement données par

$$\frac{V_{n,m} - V_{n,m-1}}{\Delta S} \quad \text{et} \quad \frac{V_{n,m+1} - V_{n,m}}{\Delta S}$$

Or l'implémentation sera la plus stable (on l'admet ici) si on fait la moyenne de ces deux approximations. Cela donne

$$\frac{\partial V}{\partial S} \simeq \frac{V_{n,m+1} - V_{n,m-1}}{2\Delta S}$$

et l'EDP discrétisée s'écrit

$$\frac{V_{n+1,m} - V_{n,m}}{\Delta t} + \frac{1}{2}\sigma^2 m^2 \Delta S^2 \frac{V_{n,m+1} - 2V_{n,m} + V_{n,m-1}}{\Delta S^2} + rm\Delta S \frac{V_{n,m+1} - V_{n,m-1}}{2\Delta S} = rV_{n,m}$$

ce qui peut se réécrire

$$V_{n+1,m} = \alpha_m V_{n,m-1} + \beta_m V_{n,m} + \gamma_m V_{n,m+1}$$

où on pose

$$\begin{aligned} \alpha_m &= \frac{1}{2}rm\Delta t - \frac{1}{2}\sigma^2 m^2 \Delta t \\ \beta_m &= 1 + \sigma^2 m^2 \Delta t \\ \gamma_m &= -\frac{1}{2}rm\Delta t - \frac{1}{2}\sigma^2 m^2 \Delta t \end{aligned}$$

pour tout $0 \leq n \leq N-1$ et $1 \leq m \leq M-1$.

1.2.3 Expression matricielle

Avec ce qui précède, on peut déterminer toutes les valeurs $(V_{n,m})_{n,m}$ discrétisées du produit dérivé, ayant simplement les paramètres σ , r et le vecteur initial $(S_m)_{0 \leq m \leq M}$ des valeurs du sous-jacent. En pratique, on passera d'un instant $n\Delta t$ au suivant, $(n+1)\Delta t$, avec l'équation matricielle

$$\overrightarrow{V_{n+1}} = A \times \overrightarrow{V_n}$$

où on a posé

$$A = \begin{bmatrix} \beta_0 & \gamma_0 & & & \\ \alpha_1 & \beta_1 & \gamma_1 & & \\ & \alpha_2 & \beta_2 & \gamma_2 & \\ & & \ddots & \ddots & \ddots \\ (0) & & & \alpha_{M-1} & \beta_{M-1} & \gamma_{M-1} \\ & & & & \alpha_M & \beta_M \end{bmatrix} \quad \overrightarrow{V_n} = \begin{bmatrix} V_{n,0} \\ V_{n,1} \\ V_{n,2} \\ \vdots \\ V_{n,M-1} \\ V_{n,M} \end{bmatrix}$$

Pour la suite, on considérera le cas où A est carrée ($N = M$), sauf mention contraire.

1.3 Résolution

Pour rappel, ce qui nous intéresse est de connaître la "vraie" valeur de notre dérivé à l'instant présent ($t = 0$), ou en tous cas celle que prédit le modèle de Black-Scholes. Celle-ci est peut être différente du prix de marché : si le produit est actuellement vendu à prix moins cher qu'il devrait l'être selon le modèle, alors on en achète.

On veut donc approximer \vec{V}_0 avec la condition au bord \vec{V}_M donnée (ou calculée) dans le contrat définissant le dérivé. C'est en fait le contraire de ce qu'on a fait jusqu'ici. En effet, on utilisait \vec{V}_0 pour calculer \vec{V}_M . En pratique, on va quand même pouvoir calculer \vec{V}_0 , en faisant l'opération

$$(A^{-1})^N \times \vec{V}_M = \vec{V}_0$$

Remarque : il est impossible de réaliser ces opérations si A n'est pas carrée, car on perd les notions d'inverse et de puissance.

1.4 Méthode alternative (M2)

Comme on le verra par la suite, l'inversion matricielle ne pose pas de problème mathématique (on montre assez facilement que la matrice A est inversible, sauf dans certains cas irréalistes), mais les méthodes numériques d'inversion sont complexes et souvent divergentes de la réalité. C'est le cas surtout lorsqu'on augmente les nombres de subdivisions N et M .

On considère donc une méthode alternative des différences finies qui ne nécessite pas d'inverser une matrice. En effet, on peut discrétiser l'EDP comme précédemment, pour obtenir (à chaque instant n)

$$\vec{V}_n = B \times \vec{V}_{n+1}$$

où on a posé

$$B = \begin{bmatrix} \beta'_0 & \gamma'_0 & & & & \\ \alpha'_1 & \beta'_1 & \gamma'_1 & & & \\ & \alpha'_2 & \beta'_2 & \gamma'_2 & & \\ & & \ddots & \ddots & \ddots & \\ (0) & & & \alpha'_{M-1} & \beta'_{M-1} & \gamma'_{M-1} \\ & & & & \alpha'_M & \beta'_M \end{bmatrix} \quad (0)$$

avec

$$\begin{aligned}\alpha'_m &= \frac{m\Delta t}{2}(\sigma^2 - \mu) \\ \beta'_m &= 1 - \Delta t(\mu - \sigma^2 m) \\ \gamma'_m &= \frac{m\Delta t}{2}(\sigma^2 + \mu)\end{aligned}$$

En faisant quelques tests, on constate que cette méthode est qualitativement comparable à la méthode des différences finies en termes de divergence si on augmente trop les paramètres. Il reste à quantifier la différence de qualité entre les deux méthodes.

2 L'implémentation

2.1 Architecture générale

Le programme sans capacité d'affichage s'organise en cinq fichiers principaux :

1. `main.cpp`
2. `matrix.cpp` : c'est le code associé aux matrices comme objets mathématiques. Ce fichier pourrait en théorie être réutilisé dans un autre contexte que celui de ce projet.
3. `eqres.cpp` : c'est le code propre à la résolution du modèle de Black-Scholes.
4. `display.cpp` : c'est le code propre à l'affichage des résultats sous forme de graphiques. Il emprunte beaucoup d'objets à la librairie Qt.
5. `mainwindow.cpp` : c'est le code permettant de générer une fenêtre.

L'architecture plus détaillée de ce code est donnée dans la figure 2. Le code contient aussi quelques autres fichiers, notamment les *headers* et un fichier `window.ui` (*user interface*). Le fichier `window.ui` n'est pas appelé.

2.2 Cas de la différence finie

Rappelons que le but premier de notre algorithme est de déterminer \vec{V}_0 grâce aux paramètres du modèle et du vecteur des valorisations à maturité \vec{V}_M :

$$(A^{-1})^N \times \vec{V}_M = \vec{V}_0$$

Pour ce faire, il nous faut d'abord définir une matrice (la classe `Matrix`), le produit matriciel et l'inverse matriciel.

Nous inversons la matrice A par la méthode du pivot de Gauss. Au vu de la forme particulière de A (tri-diagonale), on envisage aussi des méthodes d'inversion moins complexes (décomposition LU par exemple).

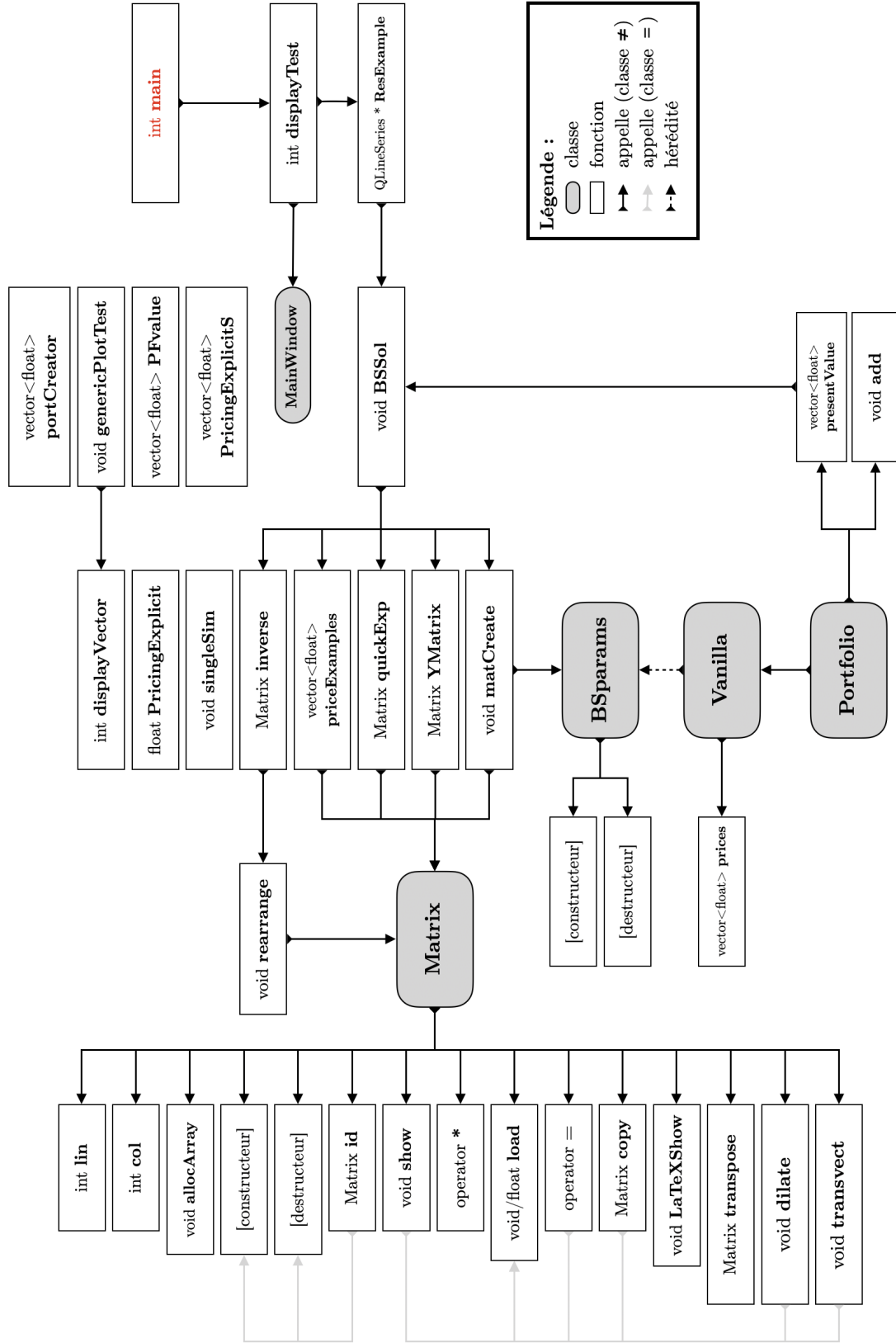


FIGURE 2 – architecture générale du programme

Par ailleurs, il est ici particulièrement judicieux d'utiliser la méthode d'exponentiation rapide pour calculer la N -ième puissance de A^{-1} , puisque les produits matriciels sont déjà de complexité $O(N^3)$.

2.3 Affichage avec Qt

2.3.1 Choix de la librairie

Pour résoudre le modèle de Black-Scholes, on approxime numériquement des fonctions réelles. Or l'évaluation et la lecture des résultats sont facilitées par un affichage graphique : on se propose donc d'ajouter une telle fonctionnalité à l'algorithme.

La librairie Qt – plus spécifiquement la sous-librairie *QtCharts* – contient des fonctions pour afficher des graphiques. Par ailleurs, la littérature associée à cette librairie est très complète et les références y sont nombreuses sur internet. C'est donc le package qui est utilisé.

Remarque : la partie du programme dédiée à l'affichage est contenue dans le fichier `display.cpp`. On y trouve notamment la fonction chargée de créer la fenêtre (une classe spécifique à Qt) et de quoi changer le format des matrices pour pouvoir en faire des graphiques.

2.3.2 Méthodes d'affichage

La fonction V (la valeur du produit dérivé) applique de \mathbb{R}^2 dans \mathbb{R} . Pour l'afficher, on peut notamment

1. afficher un filet en trois dimensions ;
2. afficher une “tranche” de la fonction, en gardant un paramètre fixé.

On implémente ici la seconde méthode, par souci de simplicité de l'implémentation et de la lecture. Dans le cas où on fixe le paramètre temporel, on lit la valeur du dérivé en fonction de la valeur du sous-jacent. Par exemple, dans la figure 3, on observe la valeur d'un **call vanille**¹ de strike 0.5 en fonction de S .

On évalue ainsi qualitativement l'efficacité des deux méthodes implémentées : la méthode **M1** (différences finies classique) a divergé de manière significative, tandis que **M2** a donné de très bons résultats jusqu'à l'approche du bord, où ceux-ci sont inutilisables. Cette différence dans la qualité de l'approximation vient de l'inversion (ou pas) de la matrice permettant de passer d'un instant n au suivant, ce qui a rajouté des erreurs.

Définition : un call vanille est une option relativement simple : elle confère le droit d'acheter un sous-jacent à prix donné, appelé le **strike** et noté K en général. Si $S \leq K$, alors le dérivé ne vaut rien : il n'est pas intéressant d'acheter à prix plus cher que le prix de marché. En revanche, la valeur du call augmente linéairement en fonction du gain obtenu lorsque K est inférieur au prix de marché. En effet, le call permet d'acheter à prix K , puis de revendre à profit sur le marché, en se faisant une marge $K - S$.

1. dans l'exemple, $N = M = 100$, $T = 0.5$, $\mu = 1\%$, $\sigma = 30\%$

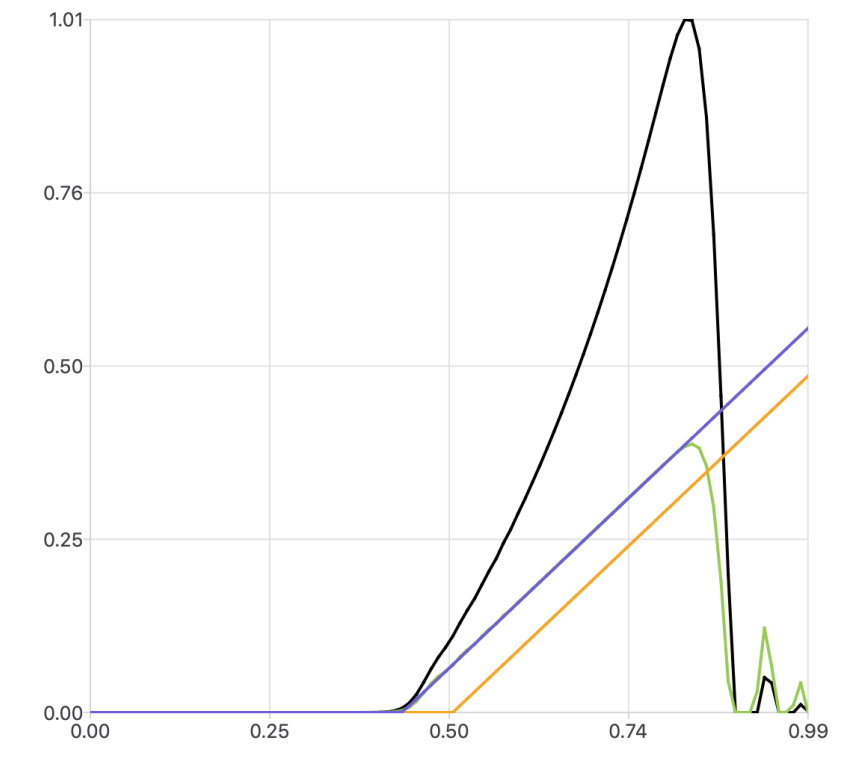


FIGURE 3 – valorisation d'un call vanille à maturité (orange), à l'instant présent (bleu) et les approximations **M1** (noir) et **M2** (vert) de ces valeurs (noir)

3 Bilan

3.1 Les limites de l'implémentation

3.1.1 Différences finies et inversion approchée

Tout comme la méthode relativement simple qui a été choisie pour résoudre le modèle (les différences finies), le code implémenté peut encore être amélioré.

D'abord, l'inversion matricielle se fait ici par la méthode du pivot de Gauss, alors qu'elle peut théoriquement être faite en complexité linéaire. D'autre part, la méthode du pivot diverge assez rapidement. Prenons l'exemple de la matrice 5×5 suivante :

$$A = \begin{bmatrix} 1.07 & 0 & 0 & 0 & 0 \\ 0 & 1.07 & 0 & 0 & 0 \\ 0 & 0 & 1.07 & 0 & 0 \\ 0 & 0 & 0 & 1.07 & 0 \\ 0 & 0 & 0 & 0 & 1.07 \end{bmatrix}$$

Et considérons $\phi(A)$ l'approximation numérique de l'inverse de A que l'on obtient par la méthode du pivot. Ici,

$$\phi(A) = \begin{bmatrix} 1 & 0.07 & 0 & 0 & 0 \\ -0.02 & 0.99 & 0.07 & 0 & 0 \\ 0 & -0.02 & 0.99 & 0.07 & 0 \\ 0 & 0 & -0.02 & 0.99 & 0.07 \\ 0 & 0 & 0 & -0.02 & 0.99 \end{bmatrix}$$

Alors on constate que

$$A * \phi(A) = \begin{bmatrix} 1.07 & 0.07 & 0 & 0 & 0 \\ -0.02 & 1.07 & 0.07 & 0 & 0 \\ 0 & -0.02 & 1.07 & 0.07 & 0 \\ 0 & 0 & -0.02 & 1.07 & 0.07 \\ 0 & 0 & 0 & -0.02 & 1.06 \end{bmatrix}$$

Si on considère la distance matricielle infinie d_∞ , alors

$$d_\infty(A * \phi(A), I_5) = 0.07$$

ce qui est déjà important, relativement à norme infinie de A (qui vaut 1). Cela permet de quantifier la divergence dans l'approximation de l'inverse par la méthode du pivot.

Ensuite, lorsqu'on élève l'inverse de A à une puissance élevée dans la méthode des différences finies, l'erreur sur certaines valeurs est conséquent et tend à exploser lorsqu'on augmente la maturité ou la tendance μ d'un dérivé. Par exemple, dans le cas de la figure 3, l'approximation devient fausse à l'approche des bords de l'ensemble de résolution : la courbe noire est censée rester parallèle à la courbe verte au voisinage de 1.

3.1.2 Solutions

Il s'agit de limiter la divergence de la méthodes des différences finies de la vraie valeur prédite par le modèle de Black-Scholes.

Première solution : On peut interdire les valorisations négatives, car elles n'existent pas en réalité (au pire, un produit financier ne vaut rien). Ainsi, lorsqu'on passe d'un instant n à l'instant $n + 1$, si une valeur $V_{n+1,m}$ est négative pour un certain m , on la remplace par 0 avant de poursuivre les calculs.

Seconde solution : On implémente aussi une variante de la méthode des différences finies. On discrétise l'EDP, puis on la met sous une forme qui est telle qu'on n'a pas besoin d'inverser une matrice pour résoudre le modèle. C'est une méthode plus stable, parce qu'on élimine les erreurs accumulées lors de la méthode du pivot.

3.2 Extensions possibles

Les implémentations du modèle de Black-Scholes sont classiques et constituent un très grand répertoire d'algorithmes. En ce qui concerne l'implémentation décrite ici, on peut encore améliorer la stabilité et la précision des résultats. On peut aussi changer le modèle mathématique de Black-Scholes, ce qui est souvent le cas en finance. Parmi les pistes concrètes à explorer, on distingue notamment :

1. la stabilisation de la méthode des différences finies, par exemple en élevant d'abord la matrice A à la puissance N , puis en approchant l'inverse de cette dernière à l'aide de la série de Neumann :

$$A^{-N} \simeq \left(\frac{1}{s} I_N + A^N \right)^{-1} = \sum_{k=0}^{\infty} (-1)^k s^{k+1} A^{Nk}$$

Il suffit alors de choisir s très grand pour obtenir le résultat, mais il reste à voir si les calculs des puissances de A sont problématiques.

2. la résolution de l'EDP par une autre méthode, comme la méthode dite *des éléments finis*. Celle-ci s'appuie sur des concepts mathématiques plus sophistiqués.
3. la subdivision du plan réel (S, t) en un maillage mieux adapté à notre problème (voir figure 4).
4. la création de portefeuilles de dérivés sur différents sous-jacents. Par exemple, on pourrait valoriser un call sur le pétrole, un put sur le blé et un autre put sur les actions d'une fintech : on approxime alors une partie de \mathbb{R}^4 , (S_1, S_2, S_3, t) .

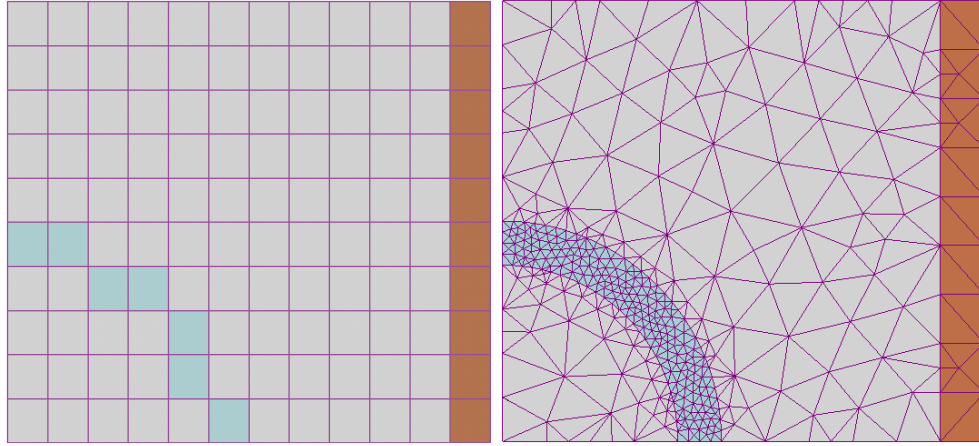


FIGURE 4 – notre maillage (à gauche) et un maillage plus sophistiqué (à droite)

Conclusion

Le modèle de Black-Scholes est un fondement de la finance de marché moderne, cependant il est souvent impossible à résoudre explicitement. Après avoir établi l'équation du modèle dans sa forme générale, nous avons présenté deux méthodes de discrétisation pour la résolution numérique de cette équation : la méthode *des différences finies* et une variante de celle-ci. Nous avons ensuite implémenté ces deux méthodes, ainsi que certains outils (portefeuilles, options vanille) courants en finance. Le seul langage utilisé est le C++ et la visualisation des résultats s'est faite à l'aide de la librairie Qt.

4 Bibliographie

- Nebra et Schaller, "Programmez avec le langage C++" (deuxième édition).
- Fanache et Smeureanu, "A Linear Algorithm for Black Scholes Economic Model", 2008. Accédé via pdfs.semanticscholar.org
- WIKIPEDIA
- Stack Overflow
- Documentation officielle Qt

Table des figures

1	discrétisation du plan (t, S)	4
2	architecture générale du programme	8
3	valorisation d'un call vanille à maturité (orange), à l'instant présent (bleu) et les approximations M1 (noir) et M2 (vert) de ces valeurs (noir)	10
4	notre maillage (à gauche) et un maillage plus sophistiqué (à droite)	13