

Projet

L'objectif de ce projet est d'appréhender par la pratique l'intérêt de la programmation orientée objet et générique par l'exemple. Ici, on s'intéressera à la résolution des équations de Black-Scholes par une méthode d'éléments finis en dimension 2 d'espace.

1 Équations de Black-Scholes

Définition 1 (Option) *En finance, une option est un produit dérivé qui donne le droit mais pas l'obligation d'acheter ou de vendre un actif financier (action, obligation, ...) à un prix fixé à l'avance (strike).*

On peut modéliser une option u sur deux actifs à l'aide des équations de Black-Scholes (voir §6 ci-dessous) :

$$\partial_t u - \nabla \cdot \mathbf{A} \nabla u - \nabla \cdot (\mathbf{a}u) + bu = 0 \quad \text{dans } \Omega \times]0, T[, \quad (1)$$

où

$$\mathbf{A} = \begin{pmatrix} \frac{(\sigma_1 x)^2}{2} & \rho \frac{\sigma_1 \sigma_2 xy}{2} \\ \rho \frac{\sigma_1 \sigma_2 xy}{2} & \frac{(\sigma_2 y)^2}{2} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_1(x, y) \\ a_2(x, y) \end{pmatrix} \text{ et } b = b(x, y).$$

On se donne également $u_0(\mathbf{x}) = u(\mathbf{x}, t = 0)$ pour tout $\mathbf{x} = (x, y)$ dans Ω .

Ce problème est en fait un problème de convection-diffusion. Pour pouvoir utiliser une méthode d'éléments finis pour le résoudre, il faut que la convection soit dominée par la diffusion. Dans la suite, on se placera donc dans ce cas.

Afin de résoudre ce problème, on va commencer par traiter le problème plus simple de la diffusion. Ensuite on traitera l'équation de la chaleur, puis on remplacera l'opérateur de Laplace par $\nabla \cdot \mathbf{A} \nabla$ et enfin on discrétisera l'opérateur du premier ordre.

1.1 Le Laplacien

Dans un premier temps, on cherchera à résoudre le problème suivant :

$$\begin{cases} -\Delta u_g = f \text{ dans } \Omega, \\ u_g = g \text{ sur } \partial\Omega, \end{cases}$$

où Ω est un ouvert polygonal connexe de \mathbb{R}^2 , $f \in L^2(\Omega)$ et $g \in H^{\frac{1}{2}}(\partial\Omega)$. Puisque l'on souhaite utiliser une méthode d'éléments finis, il faut écrire la formulation variationnelle de ce problème comme pour toute méthode de Galerkin. Pour cela, on se ramène par exemple au cas Dirichlet homogène en considérant un relèvement harmonique de g dans Ω . On cherche donc la solution $u = u_g - Rg$ qui satisfait donc le problème

$$\begin{cases} -\Delta u = f \text{ dans } \Omega, \\ u = 0 \text{ sur } \partial\Omega. \end{cases}$$

La formulation variationnelle de ce problème est : trouver $u \in H_0^1(\Omega)$ telle que

$$\forall v \in H_0^1(\Omega), \quad \int_{\Omega} \nabla u \cdot \nabla v = \int_{\Omega} f v.$$

Une méthode de Galerkin consiste à choisir un sous-espace de dimension finie de $H^1(\Omega)$. En particulier, pour la méthode des éléments finis P^1 , on choisit une triangulation \mathcal{T}_h conforme de Ω sur laquelle on construit l'espace des fonctions continues sur Ω et affines sur chaque élément $K \in \mathcal{T}_h$. On notera cet espace V_h . On peut définir alors

$$V_{h,0} = \{v \in V_h / v|_{\partial\Omega} = 0\} \subset H_0^1(\Omega).$$

Notons N_s le nombre de sommets \mathbf{x}_i de la triangulation \mathcal{T}_h . V_h est engendré par la base $(w_j)_{1 \leq j \leq N_s}$ suivante

$$\begin{aligned} \forall j, \quad w_j &\in V_h, \\ w_j(\mathbf{x}_i) &= \delta_{ij}. \end{aligned}$$

En fait, d'un point de vue numérique, nous allons considérer d'abord le problème de Neumann (sans les conditions de Dirichlet) que nous modifierons par la suite pour résoudre le problème qui nous intéresse. On cherche donc $u_h \in V_h$ telle que

$$\forall v_h \in V_h, \quad \int_{\Omega} \nabla u_h \cdot \nabla v_h = \int_{\Omega} f v_h,$$

soit encore

$$\forall 1 \leq i \leq N_s, \quad \int_{\Omega} \nabla u_h \cdot \nabla w_i = \int_{\Omega} f w_i. \quad (2)$$

Comme $u_h \in V_h$, il existe un unique vecteur $(U_j)_{1 \leq j \leq N_s} \in \mathbb{R}^{N_s}$ tel que

$$u_h = \sum_{j=1}^{N_s} U_j w_j.$$

Le système d'équations (2) s'écrit donc

$$\forall 1 \leq i \leq N_s, \quad \int_{\Omega} \nabla \left(\sum_{j=1}^{N_s} U_j w_j \right) \cdot \nabla w_i = \int_{\Omega} f w_i,$$

soit donc

$$\forall 1 \leq i \leq N_s, \quad \sum_{j=1}^{N_s} \left(\int_{\Omega} \nabla w_j \cdot \nabla w_i \right) U_j = \int_{\Omega} f w_i.$$

En notant $A_{ij} = \int_{\Omega} \nabla w_j \cdot \nabla w_i$ et $F_i = \int_{\Omega} f w_i$, résoudre ce problème revient à résoudre le système linéaire suivant :

$$AU = F$$

Cependant ce système ne correspond pas au problème de Dirichlet. Il faut donc le modifier pour imposer $u_g = g$ sur $\partial\Omega$

Une méthode facile à mettre en œuvre est une pénalisation de la matrice A et du vecteur F . Ainsi, on choisit $P \in \mathbb{R}$ très grand (10^{15} par exemple) et on définit

$$A_{dij} = \begin{cases} A_{ij} + P & \text{si } i = j \text{ et } \mathbf{x}_i \in \partial\Omega, \\ A_{ij} & \text{sinon,} \end{cases} \quad \text{et } F_{di} = \begin{cases} F_i + g(\mathbf{x}_i)P & \text{si } \mathbf{x}_i \in \partial\Omega, \\ F_i & \text{sinon.} \end{cases}$$

1.2 Équation de la chaleur

L'équation de la chaleur est :

$$\begin{cases} \partial_t u - \Delta u = f & \text{dans } \Omega, \\ u(\cdot, t = 0) = u_0, \end{cases}$$

où u_0 est la répartition initiale de la température et f une source de chaleur.

Sachant déjà résoudre le problème de Laplace, on va simplement utiliser un algorithme d'Euler implicite pour résoudre l'équation de la chaleur : connaissant u^n , la température au temps t^n , on trouve la température u^{n+1} au temps $t^{n+1} > t^n$ en résolvant

$$\frac{1}{\delta t}u^{n+1} - \Delta u^{n+1} = f + \frac{1}{\delta t}u^n \text{ dans } \Omega.$$

Du point de vue de la méthode des éléments finis, on devra résoudre le système linéaire suivant :

$$\forall 1 \leq i \leq N_s, \quad \sum_{j=1}^{N_s} \left(\int_{\Omega} \frac{1}{\delta t} w_j w_i + \nabla w_j \cdot \nabla w_i \right) U_j^{n+1} = \int_{\Omega} \left(\frac{1}{\delta t} u^n + f \right) w_i,$$

qui sera modifié pour prendre en compte les conditions aux limites de Dirichlet.

1.3 Diffusion anisotrope

Le terme de diffusion de l'équation de Black-Scholes n'est pas isotrope : on ne diffuse pas de la même manière dans chaque direction. La modification à apporter est simple puisqu'il s'agit simplement de remplacer le terme

$$\int_{\Omega} \nabla w_j \cdot \nabla w_i$$

de la formulation variationnelle par

$$\int_{\Omega} (\mathbf{A} \nabla w_j) \cdot \nabla w_i.$$

1.4 Black-Scholes

Finalement pour résoudre le problème (1), il reste deux termes à traiter :

- le terme de masse bu , mais sa discrétisation est identique à celle de $\frac{1}{\delta t}u$ et a donc déjà été traitée,
- et le terme d'ordre 1 : $\nabla \cdot (\mathbf{a}u)$.

La formulation variationnelle deviendra donc

$$1 \leq \forall i \leq N_s,$$

$$\begin{aligned} \sum_{j=0}^{N_s} \left(\int_{\Omega} \left(\frac{1}{\delta t} + b \right) w_j w_i + (\mathbf{A} \nabla w_j) \cdot \nabla w_i + (\mathbf{a} w_j) \cdot \nabla w_i \right) U_j^{n+1} \\ = \int_{\Omega} \frac{1}{\delta t} u^n w_i. \end{aligned}$$

2 Implémentation

La mise en œuvre d'un programme permettant la résolution de ce problème suivra les étapes suivantes.

1. Écrire une classe `R2` complète (opérateurs utiles surchargés). Une meilleure approche consistera à utiliser la classe `TinyVector<2,double>` présentée en cours qui servira de conteneur uniquement. La classe `R2` héritera alors de cette classe et ajoutera les opérateurs mathématiques souhaitables pour une classe manipulant les éléments de \mathbb{R}^2 .
2. Écrire une classe `Sommet` héritant de la classe `R2` qui lui ajoutera la notion de référence (`int`).
3. Écrire une classe `template <typename T> Tableau` (conteneur générique) surchargeant l'opérateur `[]` pour l'accès aux données.
4. Écrire une classe `Rn` héritant de la classe `Tableau<double>` et y ajouter l'algèbre de \mathbb{R}^n par surcharge d'opérateurs.
5. Écrire une classe `Triangle` stockant 3 pointeurs sur des `Sommet`s et une référence (`int`). Créer
 - un opérateur `[]` d'accès aux `Sommet`s (et non aux pointeurs),
 - une fonction d'accès à la référence,
 - une fonction de calcul de l'aire du triangle,
 - un opérateur `=` qui permettra de copier des `Triangles`.On pourra utiliser la classe `TinyVector<3,const Sommet*>` pour stocker les sommets.
6. Écrire une classe `Maillage`.
 - Stocker les `Sommet`s ainsi que les `Triangles` à l'aide de la classe `template <typename T> Tableau`.
 - Surcharger l'opérateur `[]` pour accéder aux `Triangles` et l'opérateur `()` pour accéder aux sommets.
 - Surcharger l'opérateur `>>` pour lire un `Maillage` dans un fichier au format `freefem++` voir paragraphe 5.1.
7. Écrire une classe `FonctionFEM` qui décrit une fonction élément fini en associant un `Rn` à un `Maillage`. Surcharger l'opérateur `<<` pour sauver la fonction dans un fichier au format `gnuplot` (voir paragraphe 5.2).
8. Écrire une classe `Matrice` utilisant le stockage de votre choix. Cette classe devra hériter d'une classe d'interface `IMatrice` qui ne contiendra qu'une fonction virtuelle pure : l'opérateur `*` du produit par un `Rn`.
9. On écrira ensuite une classe `ElementFinis` qui contiendra les fonctions nécessaires à l'assemblage de la matrice et du second membre associés

au problème de Neumann. Et la fonction permettant leur modification pour la prise en compte des conditions de Dirichlet.

10. L'algorithme de gradient conjugué utilisé pour la résolution du système linéaire sera fourni. Pour la résolution de Black-Scholes, on utilisera l'algorithme du bi-gradient conjugué qui sera lui aussi fourni.

3 Application numérique

Les données peuvent être celles fournies au paragraphe §6.4.

4 Évaluation des fonctions de base et de leur gradients

L'utilisation de fonctions de base P^1 permet de simplifier considérablement la mise en œuvre de la méthode des éléments finis. Les fonctions étant affines sur chaque triangle, on peut les calculer sur place sans avoir besoin de revenir à l'élément de référence. Ainsi, dans l'élément K , s'appuyant sur les sommets x_i , $1 \leq i \leq 3$ et d'aire $|K|$,

$$\nabla w_i = -\frac{(\mathbf{x}_{i+2} - \mathbf{x}_{i+1})^\perp}{2|K|}, \quad w_i(\mathbf{x}) = \frac{\begin{vmatrix} 1 & 1 & 1 \\ x & x_{i+1} & x_{i+2} \\ y & y_{i+1} & y_{i+2} \end{vmatrix}}{2|K|}$$

où $\mathbf{x} = (x, y)$ et où $\mathbf{x}_i = (x_i, y_i)$ sont les coordonnées du sommet i du triangle K ; $i+1$ et $i+2$ étant définis modulo 3 (i.e. assimilés à leur reste de division euclidienne par 3).

5 Format de maillages

5.1 Entrée de maillages générés par freefem++

Le format est le suivant :

```

nv nt ne
x1 y1 refv1
x2 y2 refv2
...
xnv ynv refvnv
```

```

s11 s12 s13 reft1
s21 s22 s23 reft2
...
snt1 snt2 snt3 reftnt
e11 e12 refe1
e21 e22 refe2
...
ene1 ene2 refene

```

nv est le nombre de sommets, nt le nombre de triangles et ne le nombre d'arêtes de bord. Suit la liste des sommets : coordonnées+référence. Ensuite les triangles : on donne les 3 numéros des sommets et la référence. Enfin, les arêtes, on donne cette fois les 2 numéros des sommets puis la référence. **Attention, la numérotation des sommets commence à 1 !**

5.2 Sorties au format gnuplot

Le format de sortie gnuplot est très simple. Il s'agit de dessiner les triangles en 3D, un par un :

```

x11 y11 z11
x12 y12 z12
x13 y13 z13
x11 y11 z11

```

```

x21 y21 z21
x22 y22 z22
x23 y23 z23
x21 y21 z21

```

...

On donne les trois sommets de chaque triangle en revenant au sommet initial, puis on passe au triangle suivant en laissant deux lignes blanches.

6 Notes sur les équations de Black-Scholes (B&S)

6.1 Description et réduction des équations de Black-Scholes (B&S)

En désignant par S_1, S_2 , les deux actifs, ceux-ci satisfont chacun l'équation différentielle stochastique

$$dS_{it} = S_{it}(r dt + \sigma_i dW_{it}) \quad i = 1, 2,$$

avec $\mathbb{E}(W_1 W_2) \neq 0$, c'est-à-dire que les mouvements browniens W_1 et W_2 sont corrélés. Cette corrélation induit sur S_1, S_2 une corrélation de coefficient ρ .

Le prix de l'option C_t est alors défini par l'espérance mathématique suivante :

$$\mathbb{E}(\phi(S_1, S_2)e^{-rt}),$$

où ϕ est une fonction donnée, appelée pay-off. Pour déterminer C_t , outre l'approche par une méthode du type Monte-Carlo (voir TP), il en existe une faisant recours aux formules d'Îto, conduisant ainsi à une équation aux dérivées partielles :

$$\begin{aligned} \frac{\partial C}{\partial t} + \frac{(\sigma_1 S_1)^2}{2} \frac{\partial^2 C}{\partial S_1^2} + 2\rho \frac{\sigma_1 \sigma_2 S_1 S_2}{2} \frac{\partial^2 C}{\partial S_1 \partial S_2} \\ + \frac{(\sigma_2 S_2)^2}{2} \frac{\partial^2 C}{\partial S_2^2} + r S_1 \frac{\partial C}{\partial S_1} + r S_2 \frac{\partial C}{\partial S_2} - r C = 0, \end{aligned} \quad (3)$$

définie sur un domaine $\Omega \times]0, T[$. C'est une équation aux dérivées partielles rétrograde, puisqu'on cherche $C(S_1, S_2, 0)$ sous une condition finale donnée par :

$$C(S_1, S_2, T) = \phi(S_1, S_2). \quad (4)$$

6.2 Conditions aux limites et Pay-off

6.2.1 Conditions aux limites

Dans la pratique, le domaine de résolution Ω est le carré

$$\Omega = [0, L] \times [0, L],$$

où L est l'actif maximal. Il faut alors associer, à l'équation précédente, des conditions aux limites. Ces conditions aux limites sont :

- Sur les frontières $S_1 = L$ et $S_2 = L$ et pour le call,

$$C(S_1, S_2, t) = (S_1 - K_1 e^{-r(T-t)})^+ + (S_2 - K_2 e^{-r(T-t)})^+. \quad (5)$$

Et on utilise la parité Put-Call pour déduire la condition aux limites pour le put.

- Sur la frontière $S_2 = 0$,

$$\frac{\partial C}{\partial t} + \frac{\sigma_1^2}{2} S_1^2 \frac{\partial^2 C}{\partial S_1^2} + r S_1 \frac{\partial C}{\partial S_1} - rC = 0. \quad (6)$$

- Sur la frontière $S_1 = 0$,

$$\frac{\partial C}{\partial t} + \frac{\sigma_2^2}{2} S_2^2 \frac{\partial^2 C}{\partial S_2^2} + r S_2 \frac{\partial C}{\partial S_2} - rC = 0. \quad (7)$$

Où par définition, $z^+ = \max(z, 0) \quad \forall z \in \mathbb{R}$.

On remarque ici que sur les bords $S_1 = 0, S_2 = 0$ il faut résoudre une équation de BES monodimensionnelle. On peut aussi imposer sur ces frontières une condition de sortie libre, après réécriture de l'équation sous forme conservative : voir (8) ci-dessous où on a désigné ces frontières par Γ_N .

6.2.2 Pay-off

Les choix suivants, non exhaustifs, peuvent être adoptés pour le Pay-off :
Pour le call on prend :

$$\phi(S_1, S_2) = (S_1 - K_1)^+ + (S_2 - K_2)^+.$$

Pour le put on peut prendre :

$$\phi(S_1, S_2) = ((\max(K_1, K_2) - \max(S_1, S_2))^+).$$

6.3 Réduction des équations

L'équation ci-dessus est une équation aux dérivées partielles **rétrograde** en temps. Pour modifier le sens de l'évolution en temps, on fait le changement de variable temporelle suivant :

$$\tau = T - t.$$

On pose aussi :

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \rho\mathbf{A}_{12} \\ \rho\mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \quad \mathbf{A}_{ij} = \frac{\sigma_i S_i \sigma_j S_j}{2}.$$

On introduit alors $S = (S_1, S_2)$ et on désigne par $g(S, \tau)$ la condition de Dirichlet sur la partie du bord Γ_D , incluant le cas échéant les solutions des équations (6) et(7).

Le problème est alors ramené à sa forme conservative pour faciliter son traitement par éléments finis.

$$\left\{ \begin{array}{ll} \frac{\partial C}{\partial \tau} - \nabla \cdot (\mathbf{A} \nabla C) - \nabla \cdot (\mathbf{a} C) + bC = 0 & \text{dans } \Omega \times]0, T[, \\ C(S, 0) = \phi(S) & \text{dans } \bar{\Omega}, \\ C(S, \tau) = g(S, \tau) & \text{sur } \Gamma_D \times [0, T[, \\ (K \nabla C + aC) \cdot n = 0 & \text{sur } \Gamma_N \times [0, T[, \end{array} \right. \quad (8)$$

avec

$$\Omega =]0, L[\times]0, L[,$$

$$\mathbf{a} = rS - \nabla \cdot (\mathbf{A}^T), \quad (9)$$

$$b = r - \nabla \cdot \mathbf{a}. \quad (10)$$

6.4 Quelques données pour l'évaluation numérique

Pour l'évaluation numérique de ces équations, on fournit les données suivantes :

Données pour les problèmes 2D		
	Premier actif	deuxième actif
actif max (L)	200	200
volatilité (σ)	10 %	30 %
taux-intérêt (r)	10 %	le même
coefficient de corrélation (ρ)	0.66	
Strike (K)	100	90
maturité (T)	1	le même

TAB. 1 – Données pour les tests pour B&S 2D

6.5 Solution exacte de l'équation de B&S 1D

Comme expliqué précédemment, si l'on désire utiliser une condition aux limites de type Dirichlet sur les frontières ($S_1 = 0$, et $S_2 = 0$), il faut y résoudre des équations de B&S en une dimension d'espace. Pour simplifier cela on fournit ici la solution analytique d'une telle équation. On rappelle que le problème considéré est donné, après modification du sens d'évolution en temps par :

$$\frac{\partial C}{\partial t} - \frac{\sigma^2}{2} S^2 \frac{\partial^2 C}{\partial S^2} - rS \frac{\partial C}{\partial S} + rC = 0 \quad]0, L[\times]0, T[, \quad (11)$$

avec la condition initiale

$$C(S, 0) = \phi(s) = \begin{cases} \max(S - K, 0) & \text{pour le call,} \\ \max(K - S, 0) & \text{pour le put.} \end{cases}$$

Attention ! : la valeur de K est déduite de la projection du *pay-off* bidimensionnel sur les axes $S_1 = 0$ et $S_2 = 0$.

La solution du problème est alors donnée pour le call par :

$$C(x, t) = xN(d_1(S, t)) - Ke^{-rt}N(d_2(x, t)),$$

et pour le put par :

$$C(x, t) = Ke^{-rt}N(-d_2(x, t)) - xN(-d_1(x, t)),$$

où on a posé

$$d_1(x, t) = \frac{\log(x/K) + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}, \quad d_2(x, t) = d_1(x, t) - \sigma\sqrt{t},$$

et définit

$$N(\xi) = \begin{cases} f(\xi) & \text{si } \xi > 0, \\ 1 - f(\xi) & \text{si } \xi < 0, \\ 0.5 & \text{sinon,} \end{cases}$$

avec

$$f(\xi) = 1 - (b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5) \frac{e^{-\frac{\xi^2}{2}}}{\sqrt{2\pi}}, \quad t = \frac{1}{1 + p \xi},$$

$$\begin{aligned} b_1 &= 0.319381530, & b_2 &= -0.356563782, & b_3 &= 1.781477937, \\ b_4 &= -1.821255978, & b_5 &= 1.330274429, & p &= 0.2316419. \end{aligned}$$