SIM202 : calcul d'une option européenne

sujet proposé par Nicolas KIELBASIEWICZ : nicolas.kielbasiewicz@ensta-paris.fr

18 janvier 2022

Dans le contexte des marchés financiers, il est courant de manipuler des produits financiers appelés *options*. Il s'agit de contrats d'assurance d'une durée T (maturité) émis par un organisme financier et visant à garantir à un client le prix K (appelé *strike*) d'un actif a (actions, matières premières, ...) d'une fluctuation aléatoire de sa valeur $x(t)$ (un cours sur un marché). Lorsqu'il s'agit d'un droit de vente, on parle de *put*. Lorsqu'il s'agit d'un droit d'achat, on parle de *call*. Suivant les droits de vente du titulaire du contrat, on distingue plusieurs types d'options :

— La plus simple, l'*option européenne*, stipule que le droit de vente ne peut s'exercer qu'à la fin du contrat (maturité).

— Lorsqu'on peut exercer son droit à vendre au prix K à tout instant $t < T$, on parle d'*option américaine*.

— Il existe bien évidemment d'autres types de règles.

On se place dans la suite dans le cadre d'une option européenne. On définit alors la « récompense » ou *payoff* $R(t) = (x(t) - K)^+$ qui traduit le fait que sile cours est supérieur au prix garanti, le détenteur de l'actif a intérêt à vendre sans faire jouer son contrat. Dans le cas contraire, le détenteur du contrat fera jouer son option et vendra donc au prix K. La problématique consiste donc à déterminer, dans un contexte fluctuant, le prix $p(x,t)$ d'un tel contrat à tout instant t en fonction de la valeur x de l'actif sous-jacent à l'option. Dans le cadre du modèle de Black-Scholes à risque neutre où la fluctuation aléatoire suit un processus stochastique particulier (*brownien*), on montre que le prix $p(x,t)$ est donné par l'équation de diffusion rétrograde :

$$\forall x > 0, \forall 0 < t < T \frac{\partial p}{\partial t}(x,t) + \frac{1}{2}\sigma^2 x^2 \frac{\partial^2 p}{\partial^2 x}(x,t) - rp(x,t) = 0$$

où σ^2 est ma variance de l'actif (*volabilité*) et r le taux d'intérêt. A l'échéance T du contrat, le prix $p(x,T) = (x-K)^+ = R(T)$

1 Modèle à 2 actifs

Ce modèle se généralise à plusieurs actifs. Par exemple, pour un modèle à 2 actifs, on considère le problème (x=(x_1, x_2)) :

$$
\begin{cases}
\dfrac{\partial p}{\partial t}(x,t) + \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{2} \Xi_{ij} x_i x_j \dfrac{\partial^2 p}{\partial x_i \partial x_j}(x,t) + r \displaystyle\sum_{i,j=1}^{2} x_i \dfrac{\partial p}{\partial x_i}(x,t) - rp(x,t) = 0 \quad \forall x > 0, \forall 0 < t < T \\
p(x,T) = (x_1 + x_2 - K)^+ \quad \forall x > 0
\end{cases} \tag{1}
$$

où Ξ est la matrice de covariance associée aux actifs (a_1, a_2) que l'on suppose, pour simplifier, indépendante de x et t. En effectuant le changement $t \to -t$ (renversement du temps), on aboutit à une équation de la chaleur à coefficients variables :

$$
\begin{cases}
\dfrac{\partial p}{\partial t}(x,t) - \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{2} \Xi_{ij} x_i x_j \dfrac{\partial^2 p}{\partial x_i \partial x_j}(x,t) - r \displaystyle\sum_{i,j=1}^{2} x_i \dfrac{\partial p}{\partial x_i}(x,t) + rp(x,t) = 0 \quad \forall x > 0, \forall 0 < t < T \\
p(x,0) = (x_1 + x_2 - K)^+ \quad \forall x > 0
\end{cases} \tag{2}
$$

Cette équation dégénère en $x_1 = 0$ ou $x_2 = 0$ car alors, des termes d'ordre 2 disparaissent. A cause de cette dégénérescence, il n'est pas nécessaire, et en fait pas possible, de préciser la condition aux limites en $x_1 = 0$ ou $x_2 = 0$! ! ! L'étude théorique de cette équation présente des difficultés qui sortent du cadre de ce projet. En utilisant des approches plus générales (cf Lions), on peut montrer qu'il existe une unique solution $p \in L^2(0,T,V) \cap C^0(0,T,L^2(\mathbb{R}^{+2}))$ avec :

$$
V = \left\{ v \in L^2(\mathbb{R}^{+2}), x_i \partial_{x_i} \in L^2(\mathbb{R}^{+2}), \forall i = 1,2 \right\}
$$

L'espace V muni de la norme :

$$
||v||_V^2 = ||v||^2_{L^2(\mathbb{R}^{+2})} + ||x_1 \partial_{x_1} v||^2_{L^2(\mathbb{R}^{+2})} + ||x_2 \partial_{x_2} v||^2_{L^2(\mathbb{R}^{+2})}
$$

est un espace de Hilbert dense dans $L^2(\mathbb{R}^{+2})$.

Notons que comme la solution $P \in C^0(0,T,L^2(\mathbb{R}^{+2}))$, $P(x,t)$ tend vers 0 lorsque $||x||$ tend vers 0 (les fonctions de L^2 tendent vers 0 à l'infini !). Dans la perspective de la résolution numérique de ce problème, il convient de borner l'espace, ce qui est raisonnable en pratique car x_1 ou x_2 n'ont pas de raison de tendre vers l'infini. On supposera donc dorénavant, que le problème est posé sur le carré :

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

et compte tenu de la remarque précédente, on impose la condition aux limites suivantes sur la frontière artificielle :

$$
\frac{\partial p}{\partial n}(x,t) = 0 en x_1 = a ou x_2 = a
$$

On aurait pu également imposer une condition de Dirichlet homogène $p = 0$.

Finalement, on cherche à résoudre le problème :

$$
\begin{cases}
\frac{\partial p}{\partial t} - \mathrm{div} A(x) \nabla p(x,t) + V(x) \cdot \nabla p(x,t) + rp(x,t) = 0 \quad \forall x > 0, \forall 0 < t < T \\
p(x,0) = (x_1 + x_2 - K)^+ \quad \forall x > 0 \\
\frac{\partial p}{\partial n}(x,t) = 0 en x_1 = a ou x_2 = a
\end{cases} \tag{3}
$$

où on a posé :

$$
A(x) = \frac{1}{2} \begin{bmatrix} \Xi_{11} x_1^2 & \Xi_{12} x_1 x_2 \\ \Xi_{21} x_1 x_2 & \Xi_{22} x_2^2 \end{bmatrix} \text{ et } V(x) = \begin{pmatrix} \left(\Xi_{11} + \frac{1}{2}\Xi_{21} - r\right) x_1 \\ \left(\Xi_{22} + \frac{1}{2}\Xi_{12} - r\right) x_2 \end{pmatrix}
$$

La formulation variationnelle de ce problème est :

$$
\left|
\begin{array}{l}
\text{Trouver } p \in L^2(0,T,V) \cap C^0(0,T,L^2(\Omega)) \text{ tel que } \forall q \in V : \\
\dfrac{d}{dt} \displaystyle\int_\Omega p(x,t) q(x) dx + \int_\Omega A(x) \nabla p(x,t) \cdot \nabla q(x) dx + \int_\Omega V(x) \cdot \nabla p(x,t) q(x) dx + r \int_\Omega p(x,t) q(x) = 0, \quad \forall 0 < t < T \\
p(x,0) = (x_1 + x_2 - K)^+ \quad \forall x > 0
\end{array}
\right. \tag{4}
$$

2 Eléments algorithmiques

2.1 Discrétisation par éléments finis

Comme l'espace $H^1(\Omega)$ est inclus dans l'espace V, les approximations par éléments finis de Lagrange sont conformes dans l'espace V. On peut donc appliquer à la formulation variationnelle (4) une approximation par éléments finis de Lagrange.

Soit $(w_i)_{i=1,N}$ la famille des fonctions de base attachées à un maillage du domaine Ω composé des triangles $(T_l)_{l=1,L}$ et des nœuds $(M_i)_{i=1,N}$ ($M_i = (x_{i1}, x_{i2})$). On introduit $V_h = \text{vect}(w_i)_{i=1,N}$ l'espace d'approximation interne ($V_h \subset V$). On note $p_i(t) \in V_h$ l'approximation au temps t de la solution au nœud M_i et $P(t)$ le vecteur de composantes $p_i(t)$ dans la base $(w_i)_{i=1,N}$. En considérant la formulation discrète dans V_h associée à la formulation continue (4), on obtient le système d'équations suivants :

$$\begin{cases} \mathbb{M}\dfrac{d}{dt}P(t) + (\mathbb{K} + \mathbb{B} + r\mathbb{M})\,P(t) = 0, \quad \forall 0 < t < T \\ P(0) = Q_0 \end{cases} \tag{5}$$

avec

$$\mathbb{M}_{ij} = \int_\Omega w_i w_j dx, \quad \mathbb{K}_{ij} = \int_\Omega A(x)\nabla w_i \cdot \nabla w_j dx, \quad \mathbb{B}_{ij} = \int_\Omega V(x) \cdot \nabla w_j \; w_i dx \text{ et } Q_{0_i} = (x_{i1} + x_{i2} - K)^+$$

On pose par la suite $\mathbb{D} = \mathbb{K} + \mathbb{B} + r\mathbb{M}$.

Rappelons que l'assemblage des matrices s'opère en réalisant une boucle sur tous les triangles du maillage et en utilisant une formule de quadrature numérique pour calculer les intégrales sur chacun des triangles. Attention, les matrices $\mathbb{K}$ et $\mathbb{B}$ font intervenir des données fonctions.

2.2 Discrétisation en temps

En utilisant par exemple le schéma d'Euler implicite ($\theta = 1$) avec un pas de temps constant ($t_k = k\Delta t, \quad 0 < k < N_k$), on est conduit au schéma itératif suivant pour lequel P^k est l'approximation du vecteur $P(t_k)$:

$$\left| \begin{array}{l} P^0 = Q_0 \\ (\mathbb{M} + \Delta t\mathbb{D})\,P^{k+1} = \mathbb{M}P^k, k = 0, N_k - 1 \end{array} \right. \tag{6}$$

On pourra dans un deuxième temps envisager d'utiliser le schéma de Crank-Nicholson d'ordre 2 :

$$\left| \begin{array}{l} P^0 = Q_0 \\ \left(\mathbb{M} + \frac{\Delta t}{2}\mathbb{D}\right) P^{k+1} = \left(\mathbb{M} - \frac{\Delta t}{2}\mathbb{D}\right) P^k, k = 0, N_k - 1 \end{array} \right. \tag{7}$$

2.3 Résolution du système linéaire

Les matrices ne sont pas toutes symétriques. En particulier, $\mathbb{B}$ ne l'est pas et donc $\mathbb{D}$ non plus !!!

Bien que non symétriques, ces matrices sont à profil symétrique. On pourra donc utiliser le stockage profil (cf [CiarLun14]).

Le système linéaire obtenu est de la forme $\mathbb{E}P^{k+1} = \mathbb{F}P^k$ avec $\mathbb{E}$ une matrice non symétrique. On implémentera la factorisation LU en stockage profil de la matrice $\mathbb{E}$, puis les algorithmes de descente/remontée permettant de résoudre le système $\mathbb{L}\mathbb{U}P^{k+1} = \mathbb{F}P^k$.

3 Eléments de conception

La résolution du problème nécessite le développement de tous les outils permettant d'implémenter les schémas itératifs (6) et (7). On aura donc essentiellement besoin d'une classe de vecteurs, de matrices creuses (stockage profil) équipée du produit matrice vecteur et d'un algorithme de factorisation LU, d'une classe de maillage équipée d'un générateur de maillage d'un domaine rectangulaire.

3.1 L'objet `Vecteur`

Dans ce projet, une classe de vecteurs réelle est suffisante. On choisira de reprendre une des classes développées en TP (générique ou non).

3.2 L'objet `Matrice`

On développera une classe `Matrice` utilisant le stockage profil (cf [CiarLun14]). Idéalement, il serait souhaitable de développer une classe `Matrice` supportant à la fois les matrices symétriques (stockage de la partie inférieure seulement) et les matrices non symétriques à profil symétrique. On propose d'utiliser un schéma par héritage : la classe `Matrice` ayant pour descendant les classes `MatriceSym` et `MatriceNonSym`. On implémentera l'algorithme de factorisation/résolution LU pour les matrices non symétriques à profil symétrique. Il faudra également prévoir les opérations arithmétiques de matrices symétriques et non symétriques !

3.3 L'objet `Maillage`

Un maillage consiste en une liste de points (nœuds) et une liste de triangle définis par les numéros des 3 sommets et la liste des nœuds milieux dans le cas où on utilise une approximation P2. On développera donc les classes élémentaires `Point` et `Triangle`, cette dernière possédant un attribut définissant l'ordre de l'élément (1 ou 2). On écrira un utilitaire permettant de mailler le carré unité basé sur un découpage uniforme suivant x_1 et x_2 (pas nécessairement le même). A l'aide d'une translation et d'une homothétie, il est facile d'en déduire un maillage d'un rectangle quelconque $]a,b[\times]c,d[$. La classe maillage proposera également une fonctionnalité d'écriture de maillage sur un fichier : nombre de points, nombre de triangles, ordre des éléments finis, liste des points, liste des numéros des triangles pour une exploitation avec MATLAB.

4 Validation/visualisation des résultats

Afin d'exploiter les résultats, on écrira dans un fichier les vecteurs P^k. Ce fichier sera relu, conjointement avec le fichier de maillage, avec MATLAB pour faire une exploitation graphique des résultats : représentation à différents instants du prix (en échelle log, c'est mieux). On pourra tester le code avec le jeu de données suivantes (volatilité à 20% et taux d'intérêt à 5%) :

$$\Xi = \begin{bmatrix} 0.04 & -0.024 \\ -0.024 & 0.04 \end{bmatrix} \text{ et } r = 0.05$$

sur une durée de 2 ans avec des pas de temps de l'ordre de 1 à 5 jours.

5 Organisation du travail (3p)

La répartition du travail dans un premier temps peut se faire comme suit :
- Un étudiant se charge de la classe `Maillage`
- Un étudiant se charge de la classe `Matrice`
- Un étudiant se charge du développement des calculs éléments finis et de l'implémentation du schéma itératif

Vous devrez fixer très rapidement les fonctionnalités des classes `Maillage` et `Matrice` qu'aura à utiliser l'étudiant qui développe le calcul éléments finis. Chaque étudiant devra valider ses développements.

Dans un deuxième temps, un étudiant se consacrera à l'exploitation des résultats tandis que les 2 autres continueront le développement du code C++.

Références

[Seid04] R.Seidel, *Tools for Computational Finance*, Springer-Verlag, 2004

[LambLap96] D.Lamberton, B. Lapeyre, *Stochastic Calculus Applied to Finance*, Chapman & Hall , 1996

[HachPiro08] Y. Hachdou, O. Pironneau, *Computational Methods for Option Pricing*, Frontiers in Applied MAthematics, 2008

[CiarLun14] P. Ciarlet, E. Lunéville, *Cours MA201 : Méthode des éléments finis*, Presse de l'ENSTA-Paristech, 2014