Méthodes Déterministes en Finance

TP: Option avec coût de transaction

1ier Février 2010

On considère un modèle d'option avec coût de transaction [1]. On cherche à calculer le prix de l'option $v = v(t, s)$ à l'instant $t = T$. Posons $\Omega := (0, T) \times (0, S_{max})$. L'option satisfait l'équation suivante:

$$\partial_t v - \frac{1}{2} s^2 \Big(\sigma^2 \partial_{s,s} v - \epsilon \big| \partial_{s,s} v \big| \Big) - rs\, \partial_s v + r\, v = 0, \quad (t, s) \in \Omega, \tag{1a}$$

$$v(t, S_{max}) = 0, \quad t \in (0, T), \tag{1b}$$

$$v(0, s) = \varphi(s), \quad s \in (0, S_{max}) \tag{1c}$$

avec $\varphi(s) = (K - s)_+$ (dans un premier temps), où σ, r, K sont des constantes strictement positives, et $\epsilon \leq \sigma^2$. *Pour les applications numériques on prendra les paramètres suivant:* $K = 100$, $S_{max} = 200$, $T = 0.2$, $\sigma = 0.4$, $r = 0.1$, et $\epsilon = \frac{\sigma^2}{5}$.

1. Pour $\alpha \in \mathbb{R}$ on introduit $\mathcal{A}_\alpha$ l'opérateur linéaire t.q.

$$\mathcal{A}_\alpha v := -\frac{1}{2}\, \alpha\, s^2\, \partial_{s,s} v - rs\, \partial_s v + r\, v.$$

Mettre l'équation (1a) sous la forme d'une équation d'Hamilton-Jacobi-Bellman:

$$v_t + \max_\alpha \mathcal{A}_\alpha v = 0. \tag{2}$$

où α peut prendre deux valeurs possibles $\{a_1, a_2\}$ à préciser.

2. On considère un maillage uniforme (s_j), $s_j = jh$, $h = \frac{S_{max}}{I+1}$. On cherche $P_j^n \simeq v(t_n, s_j)$, une approximation de la solution exacte. On note $P^n = [P_0^n, \ldots, P_I^n]^T$. On note A^{a_k} ($k = 1, 2$) les matrices de discrétisation spatiale correspondantes à $\mathcal{A}_{a_k}$ [2]. Ecrire un schema d'Euler Explicite.

3. Implémentation: télécharger le fichier `tp.sce`.
a) Dans `tp.sce`, programmer les 2 matrices (cf la zone `// COMPLETER A1,A2`). Programmer le schéma d'Euler Explicite. (paramêtre: `SCHEMA='EE-CT'`, correspondant au schema Euler Explicite avec Cout de Transaction).
b) Tester avec I petit (Ex: $I = 20$, $N = 100$), et dans le cas du payoff $\varphi(s) = (K - s)_+$ (correspond au choix `test=1` en debut de Fichier). Constater que pour $I = 100$ le schema est instable.
c) Deviner la solution exacte dans ce cas; l'afficher pour comparaison (on completera la fonction `Pexact` située en début de fichier).

4. Ecrire et tester un schéma d'Euler Implicite de type splitting (décentré en espace), noté `SCHEMA='EI-SPLIT-CT'`. Le schema consiste à prendre la même formule que le schema explicite (maximum de deux estimations), mais en remplacant les estimations explicites par des estimations implicites.

[1] Modèle de Leland, 1985.

[2] c'est à dire t.q. $(\mathcal{A}_{a_k} v(t_n, \cdot))(s_j) \simeq (A^{a_k} P^n)_j$.

Exemple : prendre $(I, N) = (100, 100)$ et vérifier que le schema est stable.

5. **a)** Ecrire un schéma d'Euler complètement Implicite, noté EI.
b) Montrer que chaque étape de calcul de $x = P^{n+1}$ en fonction de P^n peut se mettre sous la forme d'un problème de recherche de zéro, $F(x) = 0$, où F est définie par

$$F(x) := \max(B^{a_1} x - b, B^{a_2} x - b), \quad b = P^n.$$

(maximum composante par composante), et où les B^{α_k} s'expriment simplement en fonction des A^{α_k}.
c) Proposer une méthode de type Newton pour résoudre $F(x) = 0$. Vérifier qu'elle est bien définie. Que dire sur la vitesse de convergence de cette méthode (nombre d'itérations) ?

6. Programmer la méthode EI. Pour cela, on commencera par programmer l'algorithme de Newton, en suivant l'approche utilisée pour les options americaines (fichier `Newton_q.sci`, réalisé les options américaines).

7. Tester l'algorithme dans le cas d'un payoff d'allure donnée en Fig.1. [3] (Mettre `test=2` dans `tp.sce`). Suivant le test réalisé, un fichier de données sera sauvegardé.

Télécharger et executer `comparer.sce` (permet de comparer graphiquement 2 solutions du problème). On pourra ainsi comparer graphiquement les valeurs et les delta des solutions obtenues dans le cas $\epsilon = 0$ (pas de coût de transaction) et le cas $\epsilon \neq 0$.

Eléments de solution:
1. $\{a_1, a_2\} := \{\sigma^2 - \epsilon,\ \sigma^2 + \epsilon\}$.
2. A^α*: obtenue comme pour les Diff. Finies;* $P^{n+1} = \min((I - \delta t A^{a_1})P^n, (I - \delta t A^{a_2})P^n)$.
3. $B^{\alpha_k} = I + \delta t A^{\alpha_k}$*; Solution exacte obtenue pour* $\alpha = a_1$ *!*
4. $P^{n+1} = \min((I + \delta t A^{a_1})^{-1} P^n, (I + \delta t A^{a_2})^{-1} P^n)$.
5-6. Cf. tp Américaines. Les matrices B^{a_1}, B^{a_2} *sont des* M*-matrices (dans le cas d'un décentrage à droite), donc toute matrice formée des lignes des* B^{a_k} *sera aussi une* M*-matrice, et donc une matrice monotone. Par équivalence avec l'algorithme de Howard, on obtient la convergence en un nombre fini d'itérations. Solutions:* `tp_solution.sce`, `Newton.sci`.

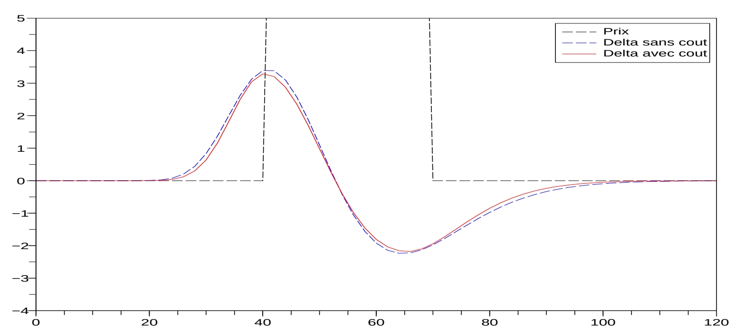

Figure 1: Deltas à $t = 0.2$.

[3] $\varphi(s) = 8\left(1_{(40,50]}(s)(s-40) + 10.1_{(50,60]} + 1_{(60,70]}(70-s)\right)$