

# Évaluation des options américaines par méthodes de Monte-Carlo

Jacky Mochel

3 décembre 2002

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Définitions et notations . . . . .	3
1.2	Quelques notions utiles . . . . .	3
1.3	Remarques . . . . .	4
<b>2</b>	<b>Étude théorique des options américaines</b>	<b>5</b>
2.1	L'enveloppe de Snell du pay-off . . . . .	5
2.2	Temps d'arrêt optimal . . . . .	6
<b>3</b>	<b>Méthode de Cox Ross Rubinstein</b>	<b>7</b>
3.1	Le modèle binomial . . . . .	7
3.2	Calcul de l'enveloppe de Snell . . . . .	7
3.3	Algorithme . . . . .	7
3.4	Choix des paramètres . . . . .	8
3.5	Résultats . . . . .	8
3.6	Applications . . . . .	8
<b>4</b>	<b>Méthode de Longstaff-Schwartz</b>	<b>10</b>
4.1	Détermination du temps d'arrêt optimal . . . . .	10
4.2	Calcul de l'espérance conditionnelle . . . . .	10
4.3	Algorithme . . . . .	10
4.4	Choix des paramètres . . . . .	11
4.5	Résultats . . . . .	11
<b>5</b>	<b>Méthode de quantification</b>	<b>13</b>
5.1	La quantification . . . . .	13
5.2	Calcul de la matrice de transition . . . . .	13
5.3	Algorithme . . . . .	13
5.4	Choix des paramètres . . . . .	14
5.5	Résultats . . . . .	14
<b>6</b>	<b>Option sur multi sous-jacents</b>	<b>16</b>
6.1	Méthode de Cox-Ross-Rubinstein . . . . .	16
6.2	Méthode de Longstaff-Schwartz . . . . .	17
6.3	Méthode de quantification . . . . .	18
	<b>Bibliographie</b>	<b>19</b>

## 1 Introduction

Les risques attachés à toute opération de bourse ont conduit à imaginer des systèmes permettant de les limiter. Ainsi s'est développé le marché des options, où l'acheteur acquiert non plus la valeur elle-même, mais le droit de l'acheter ou le droit de la vendre. Par exemple une option d'achat (call) ou de vente (put) européenne est un titre financier conditionnel qui donne le droit, mais non l'obligation, d'acheter ou de vendre un actif déterminé à un prix convenu à l'avance, et à une date déterminée. Une option américaine diffère de l'option européenne en ce sens qu'elle peut être exercée à n'importe quel instant entre la date d'achat et l'échéance.

Pour obtenir ce droit d'acheter ou de vendre l'actif sous-jacent à un prix fixé à l'avance, l'acheteur paie immédiatement au vendeur la valeur de l'option, appelée la prime. La question de la détermination de la prime est le problème du pricing : à quel prix vendre l'option, c'est à dire quelle doit être la valeur initiale d'un portefeuille autofinancé permettant de répliquer la richesse rapportée à l'exercice de l'option ?

Le but de mon EA est de comprendre quelques-unes des méthodes utilisées pour pricer ce type d'options, et de les implémenter en C++.

### 1.1 Définitions et notations

On appelle prix d'exercice et on note  $K$  le prix fixé à l'avance auquel l'option donne droit d'acheter ou vendre l'actif sous-jacent.

On appelle maturité ou échéance et on note  $T$  l'instant auquel l'option pourra être exercée (dans le cas des options européennes), l'instant final jusqu'auquel l'option peut être exercée (cas des options américaines).

On discrétisera l'intervalle de temps en considérant les instants  $0 = t_0 < t_1 < \dots < t_{N-1} < t_N = T$ .

On note  $S_t$  le prix du sous-jacent à l'instant  $t$  ( $0 \leq t \leq T$ ),  $S_n$  désignant alors  $S_{t_n}$ . On note  $(F_n)$  la filtration des informations disponibles à l'instant  $n$  ( $0 \leq n \leq N$ ).

On appelle pay-off et on note  $Z_n$  le gain rapporté par l'exercice de l'action à l'instant  $n$ . Dans le cas d'un put, ce gain vaut  $(K - S_n)_+$ .

### 1.2 Quelques notions utiles

#### L'actualisation

Il existe un actif sans risque  $S^0$ , typiquement un placement à la banque, dont la valeur à l'instant  $t_n$  est

$$S_n^0 = S_0^0 \prod_{k=0}^{n-1} (1 + r(k))$$

, où  $r(k)$  est le taux d'intérêt à la période  $t_k$ . On notera un zéro-coupon

$$B(j, l) = \frac{1}{\prod_{k=j}^{l-1} (1 + r(k))} = \frac{S_j^0}{S_l^0}$$

C'est le prix à l'instant  $j$  d'1 Euro disponible à l'instant  $l$ .

La valeur actualisée à l'instant  $n$  d'une somme  $A$  disponible à l'instant  $m > n$  est  $A \frac{S_n^0}{S_m^0}$ . Dans le cas simple du taux d'intérêt constant, elle est égale à  $\frac{A}{(1+r)^{m-n}}$ .

On notera  $\mathcal{A}_m$  (noter le changement de police) la valeur actualisée à l'instant 0 de la somme  $A$  disponible à l'instant  $m$ . Concrètement cela signifie qu'en disposant maintenant d'une somme  $\mathcal{A}_m$  et en la plaçant à la banque, on récupère la somme  $A$  à la date  $m$ .

### La probabilité de risque neutre

Il existe une probabilité  $\mathbb{P}^*$  dite probabilité de risque neutre sous laquelle la valeur actualisée  $\mathcal{S}_n$  de l'actif est une martingale, c'est à dire vérifie  $\mathbb{E}^*(\mathcal{S}_{n+1}|F_n) = \mathcal{S}_n$ .

Sous cette probabilité, la valeur actualisée d'un portefeuille de couverture autofinancé  $\mathcal{V}_n = \mathcal{V}_0 + \sum_{j=1}^n \varphi(j)(\mathcal{S}_{j+1} - \mathcal{S}_j)$  est une martingale également, quelque soit la stratégie admissible  $\varphi$ .

En particulier,  $\mathcal{V}_n = \mathbb{E}^*(\mathcal{V}_m|F_n)$  pour tout  $m \geq n$ .

Dans la suite, nous nous placerons systématiquement sous cette probabilité.

### 1.3 Remarques

Le prix du call américain est le même prix que celui du call européen, dont on connaît une formule exacte. Dans toute la suite, nous ne nous intéresserons alors qu'à l'option de vente.

## 2 Étude théorique des options américaines

Une option américaine peut être exercée à tout instant  $t$  entre 0 et  $T$ . Nous nous contenterons ici d'étudier les options bermudiennes, qui peuvent être exercées à des instants discrets  $0 = t_0 < t_1 < \dots < t_N = T$ . Ces options approchent bien les options américaines lorsque  $N$  devient grand.

Je ne considérerai dans mes algorithmes que des temps équidistribués  $t_i = i \frac{T}{N}$ .

Une telle option est définie par une suite  $(Z_n)$  positive et adaptée à la filtration  $F_n$ , représentant le profit que donnerait l'exercice de l'option à l'instant  $n$ . Nous notons  $\mathcal{Z}_n$  la valeur actualisée du profit.

### 2.1 L'enveloppe de Snell du pay-off

Pour définir la valeur de l'option à l'instant  $t_n$ , il faut raisonner par récurrence à partir de l'échéance  $t_N$ . La valeur  $U_N$  d'une option de vente à l'échéance est, comme pour l'option européenne,  $U_N = Z_N = (K - S_N)_+$ .

À tout instant  $n - 1$ , le détenteur de l'option a le choix entre :

- Exercer son option et en tirer immédiatement un profit  $Z_{n-1}$
- Conserver son option, et détenir à l'instant suivant la valeur  $U_n$ , dont la valeur actualisée à l'instant  $n - 1$  est  $U_n B(n - 1, n)$ . L'espérance de cette valeur, connaissant l'histoire jusqu'à l'instant  $n - 1$ , est  $B(n - 1, n) \mathbb{E}^*(U_n | F_{n-1})$ .

La valeur  $U_{n-1}$  de l'option à l'instant  $t_{n-1}$  s'en déduit :

$$U_{n-1} = \max [Z_{n-1}, B(n - 1, n) \mathbb{E}^*(U_n | F_{n-1})]$$

Le système vérifié par  $U_n$  est donc :

$$\begin{cases} U_N &= Z_N \\ U_{n-1} &= \max [Z_{n-1}, B(n - 1, n) \mathbb{E}^*(U_n | F_{n-1})] \end{cases} \quad (1)$$

En considérant les quantités actualisées :

$$\begin{cases} \mathcal{U}_N &= \mathcal{Z}_N \\ \mathcal{U}_{n-1} &= \max [\mathcal{Z}_{n-1}, \mathbb{E}^*(\mathcal{U}_n | F_{n-1})] \end{cases} \quad (2)$$

La suite  $\mathcal{U}_n$  des valeurs actualisées de l'option est l'*enveloppe de Snell* sous  $\mathbb{P}^*$  de la suite  $\mathcal{Z}_n$ , c'est à dire la plus petite surmartingale majorant  $\mathcal{Z}_n$ .

Une conséquence importante en est que la suite  $(\mathcal{U})$  est une surmartingale, puisque  $\mathcal{U}_{n-1} \geq \mathbb{E}^*(\mathcal{U}_n | F_{n-1})$ . De plus elle vérifie la propriété de martingale "sur les premiers termes" :

$$\mathcal{U}_{n-1} = \mathbb{E}^*(\mathcal{U}_n | F_{n-1}) \text{ si } \mathcal{U}_{n-1} > \mathcal{Z}_{n-1} \quad (3)$$

Dans le cas où  $(S_n)$  est une chaîne de Markov, de matrice de transition  $M$ , à valeurs dans  $\mathcal{E}$ , l'espérance conditionnelle se calcule :

$$\begin{aligned} \mathbb{E}^*(\mathcal{U}_n | F_{n-1}) &= \mathbb{E}^*(\mathcal{U}_n | S_{n-1}) = \sum_{k \in \mathcal{E}} (\mathcal{U}_n(k) \mathbb{P}^*(S_n = k | S_{n-1})) \\ &= \sum_{k \in \mathcal{E}} (\mathcal{U}_n(k) M_{S_{n-1}, k}) = (M \times \mathcal{U}_n)(S_{n-1}) \end{aligned} \quad (4)$$

## 2.2 Temps d'arrêt optimal

L'option va être exercée à un instant  $\tau = t_n$  entre l'origine et l'échéance, bornes incluses, et elle va rapporter un pay-off  $Z_n$ .  $\tau$  est un temps d'arrêt, puisque la décision d'exercer l'option à un instant  $n$  se fera en connaissance du seul cours de l'actif sur la période  $[0 \dots t_n]$ .

En considérant la suite  $\mathcal{U}_n$  définie dans la formule (2) comme l'enveloppe de Snell de la suite  $\mathcal{Z}_n$ , on a  $\mathcal{U}_n \geq \mathcal{Z}_n$  pour tout  $0 \leq n \leq N$ , et  $\mathcal{U}_N = \mathcal{Z}_N$ .

Définissons  $\tau_0$  comme le premier instant où  $\mathcal{U}_{\tau_0} = \mathcal{Z}_{\tau_0}$ . C'est évidemment un temps d'arrêt, à valeur dans  $\{0 \dots N\}$ . Pour tout  $t_n < \tau_0$ , on a  $\mathcal{U}_n > \mathcal{Z}_n$ , donc la propriété de martingale (3). On en déduit le prix de l'option :

$$\mathcal{U}_0 = \mathbb{E}^*(\mathcal{U}_{\tau_0} | F_0) = \mathbb{E}^*(\mathcal{Z}_{\tau_0} | F_0)$$

Notons  $\mathbb{T}_{[i,j]}$  l'ensemble des temps d'arrêt à valeur dans  $\{t_i, \dots, t_j\}$ .

Par définition, un temps d'arrêt  $\tau^*$  est optimal s'il réalise le sup sur tous les temps d'arrêt du pay-off actualisé :  $\mathcal{Z}_{\tau^*} = \sup_{\tau \in \mathbb{T}_{[0,N]}} \mathcal{Z}_{\tau}$

Le temps d'arrêt  $\tau_0$  est optimal, puisque :

$$\begin{cases} \mathcal{U}_0 = \mathbb{E}^*(\mathcal{Z}_{\tau_0} | F_0) \\ \mathcal{U}_0 \geq \mathbb{E}^*(\mathcal{U}_{\tau} | F_0) \geq \mathbb{E}^*(\mathcal{Z}_{\tau} | F_0) \text{ pour tout temps d'arrêt } \tau \end{cases} \quad (5)$$

La valeur  $U_0$  de l'option à l'instant initial est donc le sup sur  $\mathbb{T}_{[0,N]}$  de l'espérance de pay-off actualisé :

$$U_0 = \mathcal{U}_0 = \sup_{\tau \in \mathbb{T}_{[0,N]}} \mathbb{E}^*(\mathcal{Z}_{\tau})$$

De façon générale, posons

$$\tau_n = \inf (t \geq n ; \mathcal{U}_t = \mathcal{Z}_t) \quad (6)$$

Le prix de l'option à l'instant  $n$  est :

$$U_n = \sup_{\tau \in \mathbb{T}_{[n,N]}} \mathbb{E}^*(B(n, \tau) \mathcal{Z}_{\tau} | F_n) = \mathbb{E}^*(B(n, \tau_n) \mathcal{Z}_{\tau_n} | F_n) \quad (7)$$

### 3 Méthode de Cox Ross Rubinstein

#### 3.1 Le modèle binomial

Le taux d'intérêt est supposé constant de valeur  $r$ . Il existe deux réels  $-1 < a < b$  tels que le rapport  $\frac{S_{n+1}}{S_n}$  ne peut prendre que les valeurs  $1 + a$  et  $1 + b$ , avec probabilité  $p$  et  $(1 - p)$  respectivement.

#### 3.2 Calcul de l'enveloppe de Snell

Dans ce cadre, la suite de variables aléatoires  $(S_n)$  est une chaîne de Markov homogène par rapport à la filtration  $F_n$ . Sa matrice de transition  $Q$  vérifie :

$$Q_{x,y} = \begin{cases} p & \text{si } y = x(1+a) \\ 1-p & \text{si } y = x(1+b) \\ 0 & \text{sinon} \end{cases} \quad (8)$$

En écrivant alors  $Z_n$  comme une fonction  $\psi$  du temps  $n$  et du cours  $S_n$ , par exemple  $\psi(n, S_n) = \frac{1}{(1+r)^n} (K - S_n)_+$  dans le cas du put, on sait que son enveloppe de Snell  $\mathcal{U}_n$  s'écrit comme une fonction  $u(n, S_n)$  qui vérifie :

$$\begin{cases} u(N, \cdot) = \psi(N, \cdot) \\ u(n, \cdot) = \max[\psi(n, \cdot), [Q \times u(n+1)](\cdot)] \\ \text{avec } [Q \times u(n+1)](x) = p u(n+1, x(1+a)) + (1-p) u(n+1, x(1+b)) \end{cases} \quad (9)$$

#### 3.3 Algorithme

Pour calculer alors le prix  $u(0, S_0)$  d'un put, on peut utiliser un algorithme récursif, qui sera loin d'être optimal vu qu'il va appeler plusieurs fois la même évaluation, et risque de ne pas être implantable vu que la taille de la pile est de l'ordre de  $N$ , qui peut être très grand.

Représentons plutôt  $u(n, \cdot)$  par le vecteur de ses valeurs aux points qui servent effectivement au calcul, et qui sont en nombre fini. Pour calculer  $u(0, S_0)$  il faut connaître de  $u(1, \cdot)$  uniquement les valeurs aux points  $(1+a)S_0$  et  $(1+b)S_0$ . En poursuivant le raisonnement, on constate qu'il suffit de connaître les valeurs de la fonction  $u(n, \cdot)$  aux points  $S_0(1+a)^k(1+b)^{n-k}$ .

D'où l'idée de représenter  $u$  par un vecteur  $U_n$  de taille  $n+1$  tel que

$$U_n[i] = u(n, S_0(1+a)^i(1+b)^{n-i}), \quad 0 \leq i \leq n$$

Le système (9) s'implémente de la manière suivante :

$$\begin{cases} U_N[i] = (S_0(1+a)^i(1+b)^{N-i} - K)_+ \\ U_n[i] = \max \left[ (S_0(1+a)^i(1+b)^{n-i} - K)_+, \frac{1}{1+r} (p U_{n+1}[i+1] + (1-p) U_{n+1}[i]) \right] \end{cases} \quad (10)$$

L'intérêt de l'algorithme est d'avoir une matrice de transition  $Q$  très simple.

### 3.4 Choix des paramètres

Dans le cas des options européennes, un choix judicieux des constantes  $a, b, p, r$  permet à ce modèle binomial d'approcher le prix donné par le modèle de Black et Scholes avec un taux d'intérêt  $R$  et une volatilité  $\sigma$ . Ces constantes doivent être choisies comme suit :

$$\begin{aligned} r &= \frac{RT}{N} \text{ est le taux d'intérêt sur une période} \\ 1 + a &= (1 + r) \exp(-\sigma \sqrt{\frac{T}{N}}) \\ 1 + b &= (1 + r) \exp(\sigma \sqrt{\frac{T}{N}}) \\ p &= \frac{b-r}{b-a} \text{ définit la probabilité de risque neutre} \end{aligned}$$

Il est démontré qu'avec ce choix de constantes, le prix obtenu par le modèle de Cox-Ross pour les options américaines converge avec  $N$  vers le prix obtenu dans le modèle de Black et Scholes.

### 3.5 Résultats

On obtient un algorithme entièrement déterministe et rapide pour évaluer le prix d'une option américaine, dont le seul paramètre à ajuster est  $N$ , qui doit être très grand pour assurer la convergence du modèle.

Considérons le jeu de paramètres suivant :

$$\begin{aligned} T &= 0.5 \text{ ( 6 mois)} \\ S_0 &= K = 100 \\ R &= 4\% \\ \sigma &= 30\% \end{aligned} \tag{11}$$

On obtient les prix suivants, selon  $N$  :

$N$	1	10	100	1000	10000
Prix	9.48	7.459	7.572	7.583	7.584

La précision devient très vite bonne, puisque pour  $N = 1000$  elle est déjà inférieure à 2%. L'algorithme est rapide, puisqu'il suffit de moins d'1 seconde pour résoudre le problème avec  $N = 1000$ . Par contre le coût est en  $N^2$ , et il faut environ 1min 30 pour  $N = 10000$ .

À titre indicatif, le prix de l'option européenne correspondante est de 7,411 environ. Il est normal que l'option américaine soit plus chère, puisqu'elle donne davantage d'opportunités à son acheteur.

### 3.6 Applications

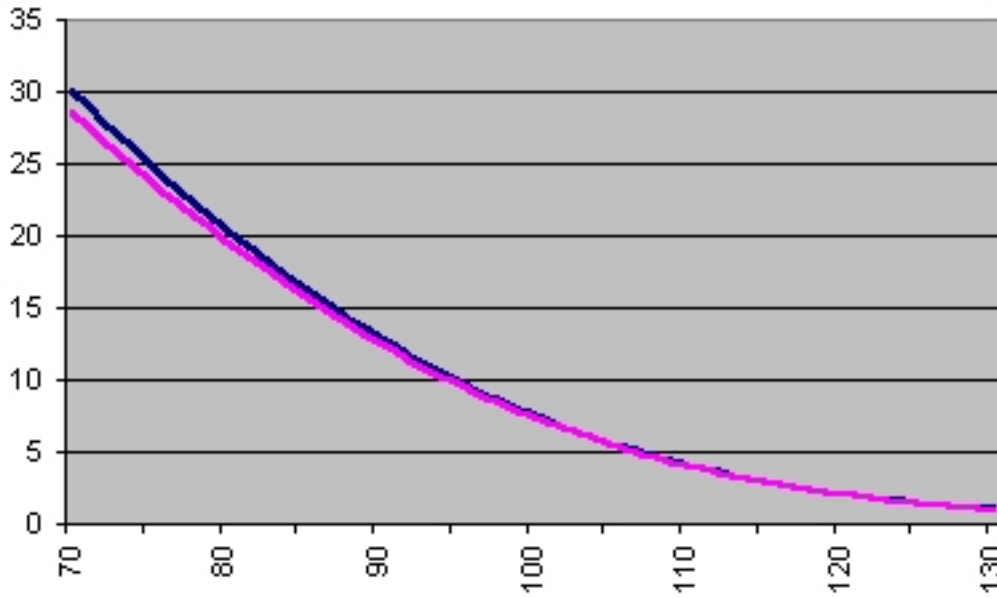
#### Fonction valeur

En conservant le jeu de paramètres (11), avec  $N = 200$ , on peut déterminer la fonction prix (en bleu) sur la figure 1.

On obtient une courbe proche de 0 quand le cours de l'action est très haut par rapport au prix d'exercice. Ceci est dû au fait que l'action n'a que peu de chance de passer sous le prix d'exercice avant l'échéance. Au fur et à mesure que la valeur de l'action baisse, le prix du put augmente progressivement d'abord, puis linéairement avec le cours. La progression linéaire est une conséquence logique de la linéarité du pay-off quand le cours est inférieur au prix d'exercice.

On a représenté en rose le prix du put européen.



FIG. 1 – Prix du put en fonction de  $S_0$ 

### Cours de l'option

Les figures 2 et 3 représentent en bleu un cours possible du sous-jacent sur la période d'exercice  $[0 \dots T]$ , et en jaune le cours du put (attention au changement d'ordonnée).

Le "gain effectif", c'est à dire le strike oté du cours du put (en bleu clair), est naturellement toujours inférieur au cours de l'action, et ce d'autant plus que :

- le cours de l'action est fort, puisque le gain effectif sera toujours au-dessous de  $K$
- on est loin de l'échéance, puisque le cours doit tendre vers le pay-off lorsque l'on approche du terme

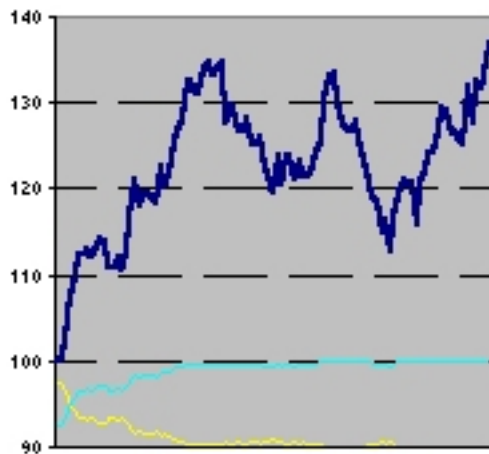


FIG. 2 – Cours de l'option - action à la hausse

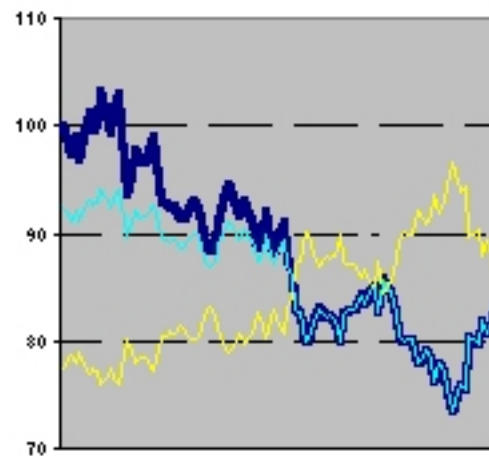


FIG. 3 – Cours de l'option - action à la baisse

## 4 Méthode de Longstaff-Schwartz

### 4.1 Détermination du temps d'arrêt optimal

Étant donnée la suite  $(\mathcal{U})$  définie par (7), définissons  $(\zeta_j)$  sur  $\{0 \dots N\}$  par la récurrence arrière :

$$\begin{cases} \zeta_N &= N \\ \zeta_n &= n \mathbb{1}_{[\mathcal{Z}_n = \mathcal{U}_n]} + \zeta_{n+1} \mathbb{1}_{[\mathcal{Z}_n < \mathcal{U}_n]} \end{cases} \quad (12)$$

On a  $\zeta_0 = \inf \{j ; \mathcal{Z}_j = \mathcal{U}_j\} = \tau_0$ , défini en (6), qui est un temps optimal. Et pour tout  $0 \leq n \leq N$ ,  $\zeta_n = \tau_n$ , temps optimal également étant donné  $F_n$ .

En utilisant la propriété (3) de martingale avec la propriété (7), on obtient que :

$$\{\mathcal{Z}_n < \mathcal{U}_n\} = \{\mathcal{Z}_n < \mathbb{E}^* [B(n, n+1) U_{n+1} | F_n]\} = \{\mathcal{Z}_n < \mathbb{E}^* [B(n, \tau_{n+1}) Z_{\tau_{n+1}} | F_n]\} \quad (13)$$

Il faut calculer l'espérance  $Z_{\tau_{n+1}} B(j, \tau_{n+1})$  conditionnellement à  $F_n$ . Vu que la connaissance de  $F_n$  n'intervient dans le futur que par la connaissance de  $S_n$ , cela revient à calculer l'espérance par rapport à  $S_n$ .

### 4.2 Calcul de l'espérance conditionnelle

On se propose ici de calculer cette espérance conditionnelle par une méthode des moindres carrés. À l'instant  $j$ , elle est par définition solution du problème de minimisation suivant : Parmi les fonctions  $f$  telles que  $\mathbb{E}^* [f(S_j)^2] < \infty$ , quelle est celle qui minimise l'écart  $\mathbb{E}^* [(X_{j+1} - f(S_j))^2]$  ? Nous noterons  $\psi_j$  cette fonction solution.

La résolution numérique n'est possible qu'en se limitant à un sous-espace de  $L^2$  de dimension finie  $d$ . En notant  $(g)$  une base du sous-espace considéré, il s'agit de trouver un vecteur  $\alpha^j \in \mathbb{R}^d$  tel que la fonction  $\widetilde{\psi}_j = \alpha^j \cdot g = \sum_{k=1}^d \alpha_k^j g_k$  minimise au mieux le carré.

L'espérance est évaluée par Monte-Carlo à partir d'un échantillon de  $M$  cours notés  $S^m, m \in \{1 \dots M\}$ . Toutes les grandeurs notées  $A^m$  seront relatives à la simulation  $m$ .

### 4.3 Algorithme

On cherche à calculer le temps d'arrêt optimal  $\tau_0$  par la méthode (12). On définit pour chaque instant  $j \in \{0 \dots N\}$  un vecteur  $\tau$  de longueur  $M$ , tel que  $\tau[m] = \tau_j^m$ .

À l'instant final  $N$ , le vecteur  $\tau$  doit valoir  $N$  pour tout  $m$ . Pour  $0 \leq j < N$ ,  $\tau_{j+1}$  étant donné, on détermine le vecteur  $\alpha^j$  qui minimise la quantité

$$\chi^2 = \frac{1}{M} \sum_{m=1}^M \left[ (\alpha \cdot g)(S_j^m) - B(j, \tau_{j+1}[m]) Z_{\tau_{j+1}[m]}^m \right]^2 \quad (14)$$

Ceci nous permet alors de calculer le vecteur  $\tau_j$ , en comparant pour tout  $m$  les valeurs de  $Z_j^m$  et  $\widetilde{\psi}_j(S_j^m)$ , et en utilisant (12) avec la règle (13).

Par application de (7), le prix de l'option est finalement donné à l'instant initial par l'espérance du pay-off actualisé maximal, obtenu par la moyenne des maxima sur chaque trajectoire :

$$C_0 = \frac{1}{M} \sum_{m=1}^M (B(0, \tau_0^m) Z_{\tau_0^m}^m)$$

### Calcul des moindres carrés

L'algorithme de recherche du vecteur minimisant (14) est inspiré de [[3]], qui calcule la matrice du système dérivé, et résout par pivot de Gauss.

Vu les problèmes de valeurs singulières (système sous-déterminé) et les problèmes d'arrondis, l'algorithme de décomposition des valeurs singulières semblait peut être plus indiqué. Mais les résultats sont en fait identiques, seul change le temps de calcul.

### 4.4 Choix des paramètres

Une des difficultés de cette méthode est de choisir un système de  $L^2$  qui soit à la fois de cardinal réduit et permette une approximation assez fine d'une fonction à priori inconnue. Une possibilité est de prendre les fonctions polynômes  $(1, X, \dots, X^{d-1})$ . On conçoit bien que  $d$  trop grand alourdit considérablement les calculs sans accroître la précision, puisque les erreurs arrondis deviennent trop importantes.

L'autre paramètre important est le nombre de chemins  $M$  qu'on envisage. Il donne en fait le nombre de points d'interpolation de la fonction  $\psi$ . Il est évident que plus ce nombre est grand, meilleure sera l'approximation. Mais le calcul en est d'autant plus long.

On modélise le cours par la dynamique du brownien géométrique

$$S_t = S_0 e^{(R - \frac{\sigma^2}{2})t + \sigma W_t}$$

Sous ce modèle, la valeur actualisée  $\mathcal{S}_t = e^{-Rt} S_t$  est une martingale.

### 4.5 Résultats

Avec le même jeu de paramètres (11), on obtient les résultats approximatifs suivants pour différentes valeurs de  $M, d, N$  (sont indiqués à chaque fois la moyenne et la variance sur un échantillon de 10 tirages) :

$N = 100$  :

	$M$	10	100	1000	10000
$D = 1$	$\mu$	4.89	5.6	5.27	5.17
	$\sigma$	1.46	0.5	0.03	0.00
$D = 2$	$\mu$	8.26	6.85	7.13	7.12
	$\sigma$	9	0.5	0.05	0.01
$D = 3$	$\mu$	8.11	7.35	6.78	6.85
	$\sigma$	7	0.6	0.03	0.01
$D = 4$	$\mu$	6.18	6.75	6.29	6.46
	$\sigma$	6.5	0.5	0.02	0.01

	$M$	10	100	1000	10000
$D = 5$	$\mu$	6.33	5.97	6.37	
	$\sigma$	4.5	0.13	0.09	
$D = 6$	$\mu$	5.95	5.72	7.35	
	$\sigma$	8	0.43	0.05	
$D = 7$	$\mu$	9.0	4.95	5.03	
	$\sigma$	4	0.22	0.05	
$D = 8$	$\mu$	5.27	4.36	4.54	
	$\sigma$	2	0.25	0.02	

On constate qu'effectivement l'exactitude des résultats faiblit avec les dimensions grandes, ceci étant du à des nombres à manipuler trop gros (on travaille avec des nombres de l'ordre de  $100^d$ ) qui provoquent des troncatures. Les meilleurs résultats sont obtenus en dimension  $3 \leq d \leq 5$ .

Curieusement, le prix a tendance à trop baisser lorsque  $M$  devient grand. Ceci met en évidence je pense la faiblesse de la base de polynômes, qui ne donne qu'une piètre moyenne lorsque le nombre de points d'interpolation est élevé.

La figure 4 présente l'allure des fonctions  $\psi_j$  dans le cas de la dimension 4, pour différentes valeurs de  $j$  ( $N = 100$ , en bleu  $j = 90$ , en rose  $j = 50$  et en jaune  $j = 10$ ).

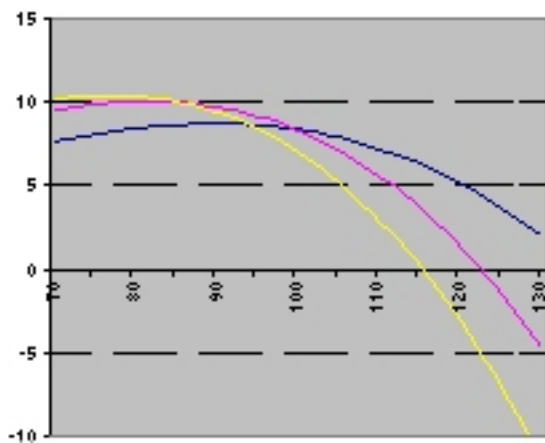


FIG. 4 – Allure des courbes  $\psi_j$

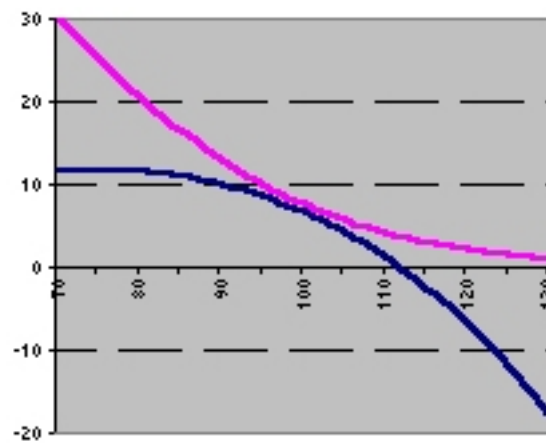


FIG. 5 – Comparaison de  $\psi_0$  à la "vraie" valeur

La figure 5 compare la fonction  $\psi_0$  (en bleu) avec la fonction valeur obtenue par la méthode de Cox-Ross (en rose).

Sur ces 2 exemples, la faiblesse des fonctions polynôme est flagrante. La sous-évaluation constatée dans le tableau de valeur est flagrante sur ces courbes. En particulier, les  $\psi_j$  prennent des valeurs négatives !

Une correction nécessaire dans l'algorithme est de remplacer  $\psi_j$  par  $\psi_j^+ = \max(\psi_j, 0)$ .

Une idée intéressante est d'ajouter dans les fonctions de base la fonction pay-off  $(K - x)_+$ . On obtient alors des résultats légèrement meilleurs, mais toujours insuffisants.

## 5 Méthode de quantification

### 5.1 La quantification

Le principe de cet algorithme est de discrétiser, pour chaque instant  $j$ , les valeurs prises par le sous-jacent, afin d'en faire une chaîne de Markov et de dégager la matrice de transition  $Q^j$ . Celle-ci permet de calculer  $U_j$  à partir de  $U_{j+1}$  par (4) dans (2).

On détermine  $q$  niveaux de quantification  $y_1^j, \dots, y_q^j$ , et on fait l'approximation de  $S_j$  par  $\tilde{S}_j$ , où  $\tilde{S}_j \in \{y_1^j, \dots, y_q^j\}$  est tel que  $\tilde{S}_j = y_k^j$  si  $S_j$  appartient à  $C_k^j$ , la cellule de Voronoï du point  $y_k^j$

$$C_k^j = \left\{ z \in \mathbb{R} ; \|z - y_k^j\| = \min_{1 \leq l \leq q} \|z - y_l^j\| \right\}$$

On pourrait tout à fait envisager de faire varier  $q$  avec  $j$ , en se contentant de peu de niveaux au début (le cours est très proche de sa valeur initiale  $S_0$ ) et en augmentant  $q$  avec le temps.

### 5.2 Calcul de la matrice de transition

À partir de la quantification précédente, on peut déterminer la matrice de transition  $\tilde{Q}^j$  :

$$\tilde{Q}_{k,l}^j = \frac{\mathbb{P}^*(\tilde{S}_j = y_k^j, \tilde{S}_{j+1} = y_l^{j+1})}{\mathbb{P}^*(\tilde{S}_j = y_k^j)} = \frac{\mathbb{P}^*(S_j \in C_k^j, S_{j+1} \in C_l^{j+1})}{\mathbb{P}^*(S_j \in C_k^j)} \quad (15)$$

En effectuant  $M$  tirages  $S^m$ , on peut calculer cette probabilité en dénombrant le nombre de tirages vérifiant  $\{S_j^m \in C_k^j, S_{j+1}^m \in C_l^{j+1}\}$  et en le divisant par le cardinal de  $\{S_j^m \in C_k^j\}$ , pour tout  $1 \leq k, l \leq q$ .

### 5.3 Algorithme

On se donne  $M$  tirages indépendants  $S^m$ .

Le principe de cet algorithme est sensiblement le même que celui présenté en (9), mais la matrice de transition est plus compliquée.

On procède par récurrence inverse. À chaque instant  $n$ , on représente  $u(n, \cdot)$  par un vecteur de taille  $q$ , tel que  $u[k] = u(n, y_k^n)$ . Puis il faut :

- Choisir  $q$  valeurs  $y_1^j, \dots, y_q^j$  approchant “au mieux” les valeurs  $S_j$
- Calculer les cellules de Voronoï correspondant à ce choix : il suffit de calculer les valeurs des frontières, qui sont  $\frac{y_k^j + y_{k+1}^j}{2}$
- Calculer la matrice de transition par (15)
- En déduire le vecteur  $u^j$  par :

$$\begin{cases} u^N[k] &= z(N, y_k^N), 1 \leq k \leq q \\ u^j[k] &= \max \left( z(j, y_k^j), \sum_{l=1}^q (B(j, j+1) u^{j+1}[l] \tilde{Q}_{k,l}^j) \right), N-1 \geq j \geq 0, 1 \leq k \leq q \end{cases} \quad (16)$$

Le coût de l'option est alors la valeur de  $u(0, S_0)$ .

## 5.4 Choix des paramètres

Les paramètres sont  $M$ , le nombre de chemins considérés pour calculer la matrice de transition,  $N$  le nombre de discrétisations de l'intervalle de temps, et surtout les niveaux de quantification. Jouent un rôle important d'une part le nombre de niveaux introduits, et d'autre part les valeurs de ces niveaux.

Une première méthode pour choisir les niveaux  $y_k^j$  est de choisir  $q$  valeurs distinctes aléatoires dans le domaine des valeurs prises par le cours à l'instant  $j$ . J'ai pris pour  $y_k^j$  des valeurs distinctes parmi les valeurs prises au temps  $j$  par l'échantillon  $S$ .

Ce choix n'est pas optimal. Une méthode plus efficace est de choisir les niveaux  $y$  de tels sorte qu'ils minimisent la distortion  $D(y) = \mathbb{E}(\min_{1 \leq k \leq n} \|X - y_k\|^2)$ , où  $X$  désigne la variable à quantifier.

## 5.5 Résultats

Avec le même jeu de paramètres (11), on obtient les résultats approximatifs suivants pour différentes valeurs de  $M, q, N$  (sont indiqués à chaque fois la moyenne et la variance sur un échantillon de 10 tirages) :

$N = 100 :$

$q \backslash M$	10	100	1000	10000
10	14.2	10.7	11.3	10.8
	12	3	1	2
100		13.8	14.4	8.1
		0.5	0.8	0.02
500			9.26	7.81
			0.09	0.00

$N = 1000 :$

$q \backslash M$	10	100	1000	10000
10	15.0	16.5	16.9	16.5
	11	3	11	6
100		14.1	8.86	9.16
		2	0.06	0.06
500			8.65	7.91
			0.07	0.01

On remarque que l'option est toujours sur-évaluée, mais qu'augmenter les paramètres  $M$  et  $q$  a tendance à rendre le résultat plus proche du prix exact. Nous sommes cependant limités par les capacités de l'ordinateur ; tirer  $M$  tirages sur  $N$  dates en double précision nécessite un espace mémoire de l'ordre de  $8MN$  octets, soit 80 Mo pour  $N = 1000, M = 10000$ . Il est difficile de pousser le calcul plus en avant. (Le temps n'est jamais un facteur limitant, le programme s'exécute toujours très rapidement, quelques minutes dans les pires cas).

Il faut que  $M$  soit très supérieur à  $q$ , sinon la quantification perd son sens, et revient à étudier indépendamment  $M$  tirages isolément.

## Représentation de la fonction valeur

Les figures 6 et 7 donnent l'allure des fonctions de valeur  $u^j$  à différentes dates  $j$ , en prenant  $N = 100$  et  $M = 1000$ .

La valeur ressemble pour les dates proches de l'échéance à une "discrétisation" du pay-off  $(K - S)_+$ , et a tendance à se rapprocher de la vraie valeur pour les dates proches de l'origine. La précision est bien entendu d'autant meilleure que le nombre de niveaux de quantification est grand.

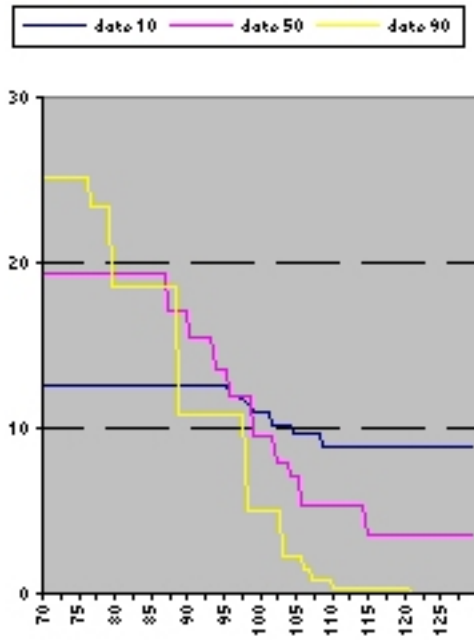


FIG. 6 – Fonction de valeur - 10 niveaux de quantification

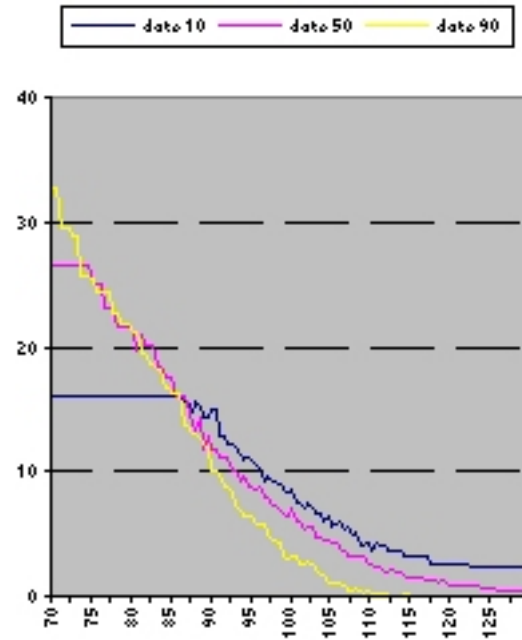


FIG. 7 – Fonction de valeur - 100 niveaux de quantification

On remarque aussi que  $u^j$  pour  $j$  proche de l'origine ne représente la fonction valeur qu'au voisinage immédiat de  $S_0$ , qui vaut 100 ici, et tend en 0 vers une fonction "à une seule marche".

## 6 Option sur multi sous-jacents

Nous pouvons maintenant généraliser au cas des options sur plusieurs sous-jacents. En considérant  $r$  actifs risqués  $S^1, \dots, S^r$ , on peut s'intéresser à une option permettant d'acheter ou de vendre ces actifs simultanément, dans une proportion donnée  $\lambda$ .

Le pay-off  $Z_n$  à l'exercice de l'option à l'instant  $n$  sera une fonction  $\psi$  du vecteur aléatoire  $S_n$  et du strike  $K$  par la relation :

$$Z_n = \lambda \cdot S_n = \lambda_1 S_n^1 + \dots + \lambda_r S_n^r$$

Pour simplifier, nous considérerons dans la suite les actifs indépendants, et suivant la même loi (avec éventuellement des volatilité différentes). Sous la probabilité  $\mathbb{P}^*$  définie pour la dimension 1, le vecteur  $\mathcal{S}$  est aussi une martingale, et toute l'étude théorique du chapitre 2 se généralise très facilement.

### 6.1 Méthode de Cox-Ross-Rubinstein

#### 6.1.1 Le nouvel algorithme

La difficulté vient de la nouvelle écriture de la matrice de transition. Dans le cas unidimensionnel, elle était très simple à exprimer par (8). Le rapport  $\frac{S_{n+1}}{S_n}$  était à valeur dans l'ensemble  $\{1+a, 1+b\}$ , à deux états seulement.

En dimension 2, il y a 4 états auxquels on peut arriver à partir de  $(S_j^1, S_j^2)$ , à savoir :

$\left\{ \left( S_j^1 (1+a_1), S_j^2 (1+a_2) \right), \left( S_j^1 (1+b_1), S_j^2 (1+a_2) \right), \left( S_j^1 (1+a_1), S_j^2 (1+b_2) \right), \left( S_j^1 (1+b_1), S_j^2 (1+b_2) \right) \right\}$ ,  
avec probabilités  $(p_1 p_2, p_1 (1-p_2), p_1 (1-p_2), (1-p_1) (1-p_2))$  respectivement.

Il apparaît qu'en dimension  $r$ , il y a  $2^r$  états auxquels on peut arriver.

Cette matrice intervient dans le calcul de la valeur de l'option par (9). En dimension 2, la fonction vectorielle  $u$  va donc se calculer par :

$$\begin{cases} u(N, \cdot) = \psi(N, \cdot) \\ u(n, \cdot) = \max[\psi(n, \cdot), \frac{1}{1+r} [Q \times u(n+1)](\cdot)] \\ \text{avec } [Q \times u(n+1)](\mathbf{x}) = \begin{array}{ll} p_1 p_2 & u(n+1, (x_1 (1+a_1), x_2 (1+a_2))) \\ + & p_1 (1-p_2) & u(n+1, (x_1 (1+a_1), x_2 (1+b_2))) \\ + & p_2 (1-p_1) & u(n+1, (x_1 (1+b_1), x_2 (1+a_2))) \\ + & (1-p_1)(1-p_2) & u(n+1, (x_1 (1+b_1), x_2 (1+b_2))) \end{array} \end{cases} \quad (17)$$

On pouvait discrétiser  $u$  car il suffisait de connaître à l'instant  $n$  les seules valeurs  $\{S_0 (1+a)^k (1+b)^{n-k}, 0 \leq k \leq n\}$ .

En dimension  $r$ , il faut connaître à chaque instant  $n$  la valeur en tous les  $r$ -uplets  $\{(S^p (1+a_p)^{k_p} (1+b_p)^{n-k_p}, 1 \leq p \leq r), 0 \leq k_1, \dots, k_r \leq n\}$ , soit  $(n+1)^r$  valeurs à calculer et stocker.

L'algorithme de Cox-Ross a un coût exponentiel, ce qui empêche son utilisation pratique sur des dimensions grandes.



### 6.1.2 Résultats

Dans un marché à 2 actifs risqués,  $S^1$  et  $S^2$ , de volatilité respective  $\sigma_1 = 0.3$  et  $\sigma_2 = 0.2$ , de même valeur initiale  $S_0 = 100$ , le prix d'une option d'achat sur  $\frac{1}{2}(S^1 + S^2)$  au prix  $K = 100$  est :

$N$	1	10	100	500	1000
Prix	4.245	4.299	4.294	4.2933	4.2931
Temps de calcul			1s	153s	1205s

Effectivement, le temps de calcul est nettement plus important ; ce problème s'aggrave dramatiquement avec des dimensions encore plus grandes.

### 6.1.3 Application

Le graphique 8 montre le prix de l'option définie ci-avant rapportant  $\lambda S^1 + (1 - \lambda) S^2$  pour  $\lambda$  variant de 0 à 1.

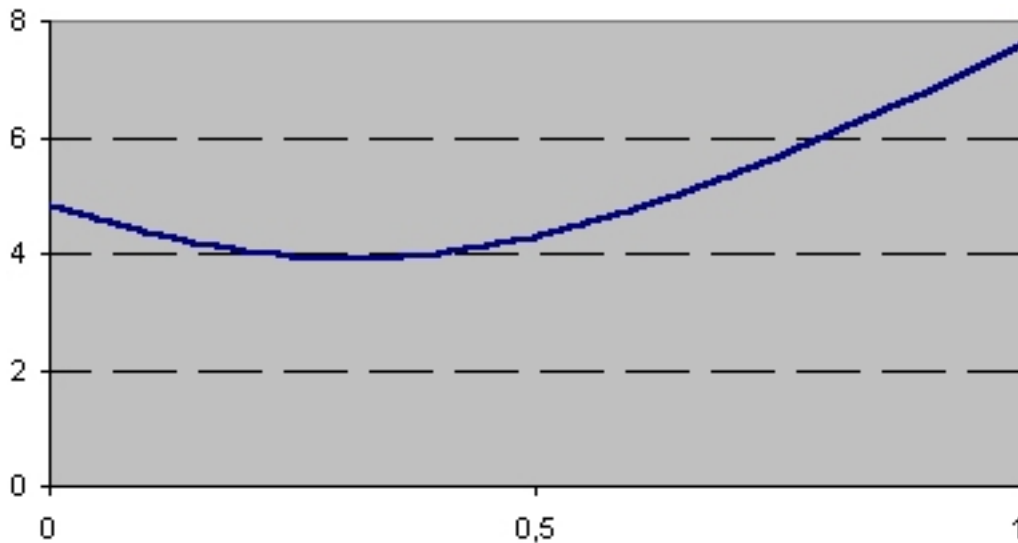


FIG. 8 – Prix du put en fonction de  $\lambda$

En 0 et 1, on retrouve bien sûr le prix dans le cas unidimensionnel (on rappelle que les volatilités sont différentes). Il est intéressant de constater que le prix présente un minimum pour une certaine combinaison non triviale des actifs.

## 6.2 Méthode de Longstaff-Schwartz

Le passage à la dimension supérieure se fait en tirant, pour tout  $1 \leq m \leq M$ ,  $r$  cours  $S^{1,m}, \dots, S^{r,m}$ . Le coût est simplement linéaire avec la dimension.

La minimisation de (14) se fait alors sur les fonctions réelles à  $r$  variables par le même algorithme. Il faut simplement choisir une base de fonctions adéquate, par exemple les polynômes à  $r$  indéterminées. Cependant, pour obtenir la même précision, la cardinalité de la base doit être plus importante : il y a par exemple 10 polynômes de degré inférieur à 2 en

dimension 3 :  $\{1, X, Y, Z, X^2, Y^2, Z^2, XY, XZ, YZ\}$ .

Il y a plus exactement  $\binom{d+r-1}{r-1} = \frac{(d+r-1)!}{(r-1)!d!}$  polynômes à  $r$  variables de degré  $d$ .

À  $d$  donné, le nombre de fonctions à considérer est de l'ordre de  $r^d$ , le nombre de tirages est  $rM$ . Le coût de cet algorithme est finalement en  $r^d$ , ce qui est bien meilleur que le précédent.

### 6.3 Méthode de quantification

Le passage à la dimension supérieure nécessite là aussi le tirage, pour tout  $1 \leq m \leq M$ , de  $r$  cours  $S^{1,m}, \dots, S^{r,m}$ , donc  $rM$  tirages au lieu de  $M$ .

Les niveaux de quantification seront des  $r$ -uplets  $y_1, \dots, y_q$ , en faisant éventuellement augmenter  $q$  avec la dimension  $r$ . La matrice de transition ne pose pas de difficultés, c'est exactement la même que (15) (avec  $S$  vectoriel).

Le problème est en fait déplacé sur la discrétisation du vecteur  $S_j$  en un vecteur  $\tilde{S}_j$  par l'appartenance à une cellule de Voronoï. Si le calcul de ces cellules est aisé en dimension 1 (il s'agit de segments), il l'est nettement moins en dimension plus grandes. Une méthode simple bien que lente pour déterminer à quelle cellule appartient un  $r$ -uplet  $S_j^m$  est de calculer sa distance par rapport à chacun des  $y_l^j$ , et de garder le minimum. Ce calcul par contre ne dépend que linéairement de la dimension.

**Références**

- [1] *Introduction au calcul stochastique appliqué à la finance*  
D. Lamberton, B. Lapeyre
- [2] *Understanding Numerical Analysis for Financial Models*  
B. Lapeyre, A. Sulem, D. Talay, 2002
- [3] Numerical Recipes in C  
<http://www.library.cornell.edu/nr/bookcpdf.html>