# Projet du cours “Options américaines: Théorie et Algorithmes”

Jérôme Lelong

18 janvier 2021

Ce projet a pour but d'implémenter un outil de valorisation d'options américaines dans le modèle de Black Scholes multidimensionnel en utilisant l'algorithme de Longstaff Schwartz.

## 1 Description et objectifs du projet

Vous implémenterez en C++ un outil de calcul des prix en 0 dans le modèle de Black Scholes multidimensionnel d'options américaines dont le payoff à l'instant $t$ ne dépend que du cours $S_t$ des sous-jacents à cet instant (pas d'options path-dependent). Le modèle de Black Scholes multidimensionnel est décrit dans le paragraphe 2. Les options que votre outil devra prendre en charge sont listées dans le paragraphe 4. Le calcul du prix utilisera l'algorithme de Longstaff Schwartz avec régression polynomiale. A cet effet, je vous incite à lire le chapitre 7 du manuel de PNL [1] qui décrit les fonctionnalités offertes par la librairie pour mettre en œuvre de la regression polynomiale multivariée.

Vous baserez vos développements sur le squelette disponible sur Chamilo. Votre outil sera testé de manière automatique et devra obligatoirement s'exécuter en utilisant la commande

```
./mcam infile
```

où `infile` est un fichier dont le format est décrit dans le paragraphe 3 et qui sera lu par le `parser` utilisé dans le projet de début d'année. Des exemples de tels fichiers sont disponibles dans le squelette avec un prix de référence en en-tête. L'exécution de votre code devra produire une sortie au format `JSON`

```
{"price": valeur_du_prix}
```

et **ne rien afficher d'autre**. Il est recommandé d'utiliser l'instruction

```
std::cout << PricingResults(myprice) << std::endl;
```

Vous rendrez votre code dans une archive au format `.tar.gz` qui créera à l'extraction l'arborescence suivante

```
Equipe_i/
    |- AUTHORS
    |- CMakeLists.txt
    |- src/
    |- dat/
```

Vous pourrez utiliser le script python `CheckArchive.py` (disponible sur Chamilo) pour vérifier d'une part l'architecture de votre archive et d'autre part que votre code compile sans erreur.

1. `https://pnlnum.github.io/pnl/manual-html/pnl-manualse7.html#x9-1040007`

## 2 Modèle

On se place dans le modèle de Black Scholes de dimension $d \geq 1$. Pour $j \in \{1, \dots, d\}$

$$dS_{t,t} = S_{t,j}((r - \delta^j)dt + \sigma^j L_j dW_t)$$

où $r > 0$ est le taux d'intérêt, $\delta \in \mathbb{R}^d$ est le taux de dividende, $B$ est un mouvement brownien à valeurs dans $\mathbb{R}^d$, $\sigma = (\sigma^1, \dots, \sigma^d)$ est le vecteur des volatilités supposées constantes et strictement positives et $L_j$ est la $j$-ème ligne de la matrice $L$ définie comme une racine carrée de la matrice de corrélation $\Gamma = LL^T$

$$\Gamma = \begin{pmatrix} 1 & \rho & \dots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \dots & \rho & 1 \end{pmatrix}$$

où $\rho \in ]-1/(d-1), 1]$ pour garantir que la matrice $\Gamma$ est définie positive.

## 3 Paramètres du problème

Les paramètres du problème sont spécifiés dans un fichier suivant le format

```
clé <type> valeur
```

où *<type>* peut prendre les valeurs `int, long, float, vector, string` avec une seule clé par ligne. Les clés peuvent apparaître dans un ordre quelconque dans le fichier.

Un exemple de parseur capable de lire ce type de fichier et de créer la structure de map correspondante est disponible dans le squelette du projet disponible sur Chamilo.

Voici la liste des clés qui seront utilisées pour décrire le problème.

| Clé | Type | Nom |
|---|---|---|
| model size | int | $d$ |
| spot | vector | $S_0$ |
| maturity | float | $T$ |
| volatility | vector | $\sigma$ |
| interest rate | float | $r$ |
| dividend rate | vector | $\delta$ |
| correlation | float | $\rho$ |
| strike | float | $K$ |
| option type | string | |
| payoff coefficients | vector | $\lambda$ |
| dates | int | $N$ |
| MC iterations | long | $M$ |
| degree for polynomial regression | int | |

Si une clé attend un paramètre vectoriel et que le fichier ne contient qu'une seule valeur pour cette clé, alors cette valeur sera affectée à chacune des composantes du vecteur.

## 4 Produits à gérer

Voici la liste des options que votre pricer devra être capable de valoriser

— Option panier (clé `basket`). Payoff à l'instant $t$

$$\left(\sum_{\ell=1}^{d} \lambda_d S_{t,\ell} - K\right)_+$$

où $\lambda_i \in \mathbb{R}$ et $K \in \mathbb{R}$. Le cas $K \leq 0$ permet de considérer une option de vente.

— Option géométrique (clé `geometric_put`). Payoff à l'instant $t$

$$\left(K - \left(\prod_{\ell=1}^{d} S_{t,\ell}\right)^{1/d}\right)_+$$

avec $K > 0$ le strike. C'est une option de vente sur un panier géométrique.

— Option performance sur panier (clé `bestof`). Payoff à l'instant $t$

$$\left(\max_{\ell=1,\dots,d} \lambda_d \, S_{t,\ell} - K\right)_+$$

où $\lambda_i \in \mathbb{R}$ et $K > 0$.

L'intérêt de l'option géométrique est que $\left(\prod_{\ell=1}^{d} S_{t,\ell}\right)^{1/d}$ suit une loi log-normale qui peut s'écrire sous la forme d'un modèle de Black Scholes unidimensionnel

$$\hat{S}_0 \, \mathrm{e}^{\left((r-\delta-\hat{\sigma}^2/2)t+\hat{\sigma}B_t\right)}$$

où $B$ est un mouvement brownien à valeurs réelles et

$$\hat{S}_0 = \left(\prod_{j=1}^{d} S_0^j\right)^{1/d}; \quad \hat{\sigma} = \frac{1}{d}\sqrt{\sigma^t \Gamma \sigma}; \quad \hat{\delta} = \frac{1}{d}\sum_{j=1}^{d}\left(\delta^j + \frac{1}{2}(\sigma^j)^2\right) - \frac{1}{2}(\hat{\sigma})^2.$$

Une option vanille sur un panier géométrique de dimension quelconque peut donc se valoriser en utilisant une méthode uniquement disponible en petite dimension comme les méthodes d'arbre par exemple. L'option géométrique permet de valider une méthode numérique sur des exemples multidimensionnels.