Conception et Implémentation d'un pricer Monte-Carlo générique en C++

Jérôme Lelong

13 septembre 2017

Le but de ce projet est de réfléchir à la structure d'un outil permettant de calculer des prix d'options par une méthode de Monte Carlo ainsi que leurs dérivées par rapport au spot, puis de l'implémenter en C++.

La section 1 présente le problème auquel on s'intéresse et les motivations pratiques. La dynamique choisie pour modéliser l'évolution des cours des actifs (les sous–jacents) est détaillée dans la section 2. La section 3 vous invite à réfléchir à l'architecture du pricer.

L'exécutable développé devra lire les paramètres des produits dans un fichier texte comme expliqué à la section 5. Finalement, vous trouverez en section 6 la liste des produits que votre pricer devra prendre en charge.

1 Motivations pratiques

Lorsque l'on s'intéresse à un produit dérivé, il est important de pouvoir calculer son prix et de savoir construire son portefeuille de couverture.

Notons $S = (S_t,\ t \geq 0)$ la dynamique du sous-jacent et $\mathcal{F}$ sa filtration naturelle, en pratique nous nous restreindrons au cas du modèle de Black Scholes multidimensionnel (c.f. section 2.2). Pour simplifier les choses, nous allons supposer que le payoff des options que nous allons considérer s'écrit sous la forme

$$\varphi(S_{t_0}, S_{t_1}, \dots, S_{t_N})$$

où $0 = t_0 < t_1 < \cdots < t_N = T$ est une grille de dates de constatation équi–répartie.

Exemple. *Voici quelques exemples de fonctions φ dans le cas d'un sous-jacent unique.*

- $N = 1$ *et* $\varphi(S_0, S_1) = (S_1 - K)_+$ *correspond à une option d'achat*
- $T = 1$, $N = 12$ *et* $\varphi(S_0, S_1, \dots, S_{12}) = (S_{12} - K)_+ \mathbf{1}_{\{\forall 0 \leq i \leq 12,\ S_i \geq L\}}$ *correspond à une option d'achat barrière à monitoring discret (une date de constatation par mois). L'option est annulée si à une des dates de constatation, la valeur du sous-jacent est inférieure à L.*
- $T = 1$, $N = 365$ *et* $\varphi(S_0, \dots, S_{365}) = (\frac{1}{365}\sum_{i=0}^{364} S_i - K)_+$ *correspond à une option d'achat asiatique où la moyenne continue du sous-jacent $\frac{1}{T}\int_0^T S_u du$ a été remplacée par une moyenne discrète.*

Notons r le taux d'intérêt instantané que nous supposerons constant dans ce projet. Le prix à l'instant $0 \leq t \leq T$, avec $t_i \leq t < t_{i+1}$ d'une telle option est donné par

$$v(t, S_{t_0}, \dots, S_{t_i}, S_t) = \mathrm{e}^{-r(T-t)}\, \mathbb{E}(\varphi(S_{t_0}, S_{t_1}, \dots, S_{t_N}) | \mathcal{F}_t)$$

Cette quantité sera calculée par une méthode de Monte Carlo.

Prix à l'instant 0. Attardons nous un instant sur le cas particulier du calcul du prix à l'instant 0

$$v(0, S_0) = \mathrm{e}^{-rT}\,\mathbb{E}(\varphi(S_{t_0}, S_{t_1}, \dots, S_{t_N}))$$

On approche $v(0, S_0)$ par

$$\mathrm{e}^{-rT}\,\frac{1}{M}\sum_{j=1}^{M}\varphi(S_{t_0}^{(j)}, S_{t_1}^{(j)}, \dots, S_{t_N}^{(j)})$$

où les $N-$uplets $(S_{t_0}^{(j)}, S_{t_1}^{(j)}, \dots, S_{t_N}^{(j)})$ pour $j = 1, \cdots, M$ sont i.i.d selon la loi de $(S_{t_0}, S_{t_1}, \dots, S_{t_N})$.

Prix à un instant $t > 0$. Le calcul du prix à l'instant t est légèrement plus compliqué car il faut calculer une espérance conditionnelle. Lorsque cela est possible, une manière de traiter le calcul de cette espérance conditionnelle est de réécrire le sous-jacent S_{t+u} à l'instant $t + u$ pour $u > 0$ en fonction de la valeur S_t du sous–jacent à l'instant t et d'une quantité indépendante de $\mathcal{F}_t$. Nous verrons dans la suite que cela est en particulier possible dans le modèle de Black–Scholes et que l'on peut écrire

$$S_{t+u} = S_t \tilde{S}_u$$

où $\tilde{S}$ est un processus indépendant de $\mathcal{F}_t$, ie. indépendant du passé jusqu'à l'instant t inclus. Ainsi le prix à l'instant t se réécrit

$$v(t, S_{t_0}, \dots, S_{t_i}, S_t) = \mathrm{e}^{-r(T-t)}\,\mathbb{E}(\varphi(s_{t_0}, \dots, s_{t_i}, s_t\tilde{S}_{t_{i+1}-t}, \dots, s_t\tilde{S}_{t_N-t}))\Big|_{\substack{s_{t_k} = S_{t_k},\ k=0,\cdots,i \\ s_t = S_t}}$$

Les lettres s minuscules désignent des variables déterministes. On peut alors approcher le prix à l'instant t par la moyenne Monte Carlo suivante

$$\mathrm{e}^{-r(T-t)}\,\frac{1}{M}\sum_{j=1}^{M}\varphi(s_{t_0}, s_{t_1}, \dots, s_{t_i}, s_t\tilde{S}_{t_{i+1}-t}^{(j)}, \dots, s_t\tilde{S}_{t_N-t}^{(j)}) \tag{1}$$

où les $N-$uplets $(\tilde{S}_{t_{i+1}-t}^{(j)}, \dots, \tilde{S}_{t_N-t}^{(j)})$ sont i.i.d selon la loi de $(\tilde{S}_{t_{i+1}-t}, \dots, \tilde{S}_{t_N-t})$. Les quantités $(s_{t_0}, s_{t_1}, \dots, s_{t_i}, s_t)$ représentent les valeurs réellement observées jusqu'à l'instant t sur la trajectoire du sous-jacent sur laquelle on souhaite calculer le prix de l'option. D'un point du vue pratique, les valeurs $(s_{t_0}, s_{t_1}, \dots, s_{t_i}, s_t)$ sont les cotations du sous-jacent observées sur le marché jusqu'à la date t.

2 Modèle de Black Scholes

2.1 Cas de la dimension 1

Considérons un modèle de Black–Scholes pour modéliser l'évolution d'un sous–jacent

$$S_t = S_0\,\mathrm{e}^{(r-\sigma^2/2)t+\sigma B_t}$$

où B est un mouvement Brownien standard réel, $r > 0$ le taux d'intérêt instantané supposé constant et $\sigma > 0$ la volatilité du sous–jacent. Remarquons que pour tout $t, u \geq 0$, on peut écrire

$$S_{t+u} = S_t \, \mathrm{e}^{(r-\sigma^2/2)u+\sigma(B_{t+u}-B_t)} = S_t \tilde{S}_u \tag{2}$$

où $\tilde{S}$ est **indépendant** de $\mathcal{F}_t$ et la dynamique de $\tilde{S}$ est donnée par

$$\tilde{S}_u = \mathrm{e}^{(r-\sigma^2/2)u+\sigma\tilde{B}_u}$$

où $\tilde{B}$ est un mouvement Brownien réel standard indépendant de $\mathcal{F}_t$. Remarquons que $\tilde{S}$ suit la même dynamique que S mais a pour valeur initiale 1. La discrétisation de S sur la grille $t_0, t_1, \ldots, t_N$ s'écrit alors

$$S_{t_{i+1}} \overset{Loi}{=} S_{t_i} \, \mathrm{e}^{(r-\sigma^2/2)(t_{i+1}-t_i)+\sigma\sqrt{t_{i+1}-t_i}G_{i+1}}$$

où la suite $(G_i)_{i\geq 1}$ est une suite i.i.d. selon la loi normale centrée réduite.

2.2 Cas multidimensionnel

Considérons D actifs évoluant chacun selon un modèle de Black Scholes de dimension 1 et corrélés entre eux. La dynamique S de ces D actifs s'écrit

$$S_{t,d} = S_{0,d} \, \mathrm{e}^{(r-(\sigma_d)^2/2)t+\sigma_d B_{t,d}}, \quad d = 1 \ldots D \tag{3}$$

où r est le taux d'intérêt, $(\sigma_1, \ldots, \sigma_D)$ sont les volatilités de chacun des actifs et $B = (B_1, \ldots, B_D)^T$ est un vecteur de D mouvements Browniens standards et réels de matrice de corrélation $\mathrm{Cov}(B_t, B_t) = \Gamma t$ définie par $\mathrm{Cov}(B_{t,i}, B_{t,j}) = \Gamma_{ij} t$ avec $\Gamma_{ij} = \rho$ pour tout $i \neq j$ et $\Gamma_{ii} = 1$. Le paramètre ρ doit être choisi dans $]-\frac{1}{D-1}, 1[$ de telle sorte que Γ soit définie positive pour assurer que le marché soit complet. Attention B n'est pas un mouvement Brownien standard à valeurs dans $\mathbb{R}^D$ car ses composantes ne sont pas indépendantes mais on peut le réécrire en utilisant un mouvement Brownien standard à valeurs dans $\mathbb{R}^D$, noté dans la suite $W = (W^1, \ldots, W^d)^T$ et qui sera vu comme un vecteur colonne. Ainsi on a l'égalité en loi $(B_t, t \geq 0) = (LW_t, t \geq 0)$ pour tout $t \geq 0$ où L est la factorisée de Cholesky de la matrice Γ, .i.e $\Gamma = L'L$ avec L triangulaire inférieure. L'équation (3) se réécrit alors

$$S_{t,d} = S_{0,d} \, \mathrm{e}^{(r-(\sigma_d)^2/2)t+\sigma_d L_d W_t}, \quad d = 1 \ldots D$$

où L_d est la ligne d de la matrice L. Ainsi la quantité $L_d W_t$ est bien un réel. De cette équation, on remarque facilement que l'on peut déduire

$$S_{t_{i+1},d} \overset{Loi}{=} S_{t_i,d} \, \mathrm{e}^{(r-(\sigma_d)^2/2)(t_{i+1}-t_i)+\sigma_d\sqrt{(t_{i+1}-t_i)}L_d G_{i+1}}, \quad d = 1 \ldots D$$

où la suite $(G_i)_{i\geq 1}$ une suite i.i.d de vecteurs gaussiens centrés de matrice de covariance identité à valeurs dans $\mathbb{R}^D$. Ainsi pour simuler le processus S sur la grille $(t_i)_{i=0,\ldots,N}$ il suffit de savoir simuler des vecteurs gaussiens centrés de matrice de covariance identité.

Vous avez vu en cours d'*Introduction aux produits dérivés* qu'il était possible de construire un portefeuille de couverture pour les options d'achat et de vente dans le modèle de Black Scholes en dimension 1. Ce portefeuille de couverture est entièrement déterminé par la quantité d'actifs risqués à posséder à chaque instant t qui est donnée par la dérivée du prix par rapport à S_t, i.e $\frac{\partial v(t,S_{t_0},\cdots,S_{t_i},S_t)}{\partial S_t}$. Cette théorie est en fait valable bien au delà des options d'achat et de vente, elle est connue sous le nom de *couverture en delta*. Dans ce projet, nous l'appliquerons à d'autres produits bien plus complexes. Ces quantités sont connues sous le nom de delta de l'option et sont en pratique approchées par une méthode de différences finies

$$\frac{\mathrm{e}^{-r(T-t)}}{M2s_th}\sum_{j=1}^{M}\Big(\varphi(s_{t_0},s_{t_1},\ldots,s_{t_i},s_t(1+h)\tilde{S}^{(j)}_{t_{i+1}-t},\ldots,s_t(1+h)\tilde{S}^{(j)}_{t_N-t})$$
$$-\varphi(s_{t_0},s_{t_1},\ldots,s_{t_i},s_t(1-h)\tilde{S}^{(j)}_{t_{i+1}-t},\ldots,s_t(1-h)\tilde{S}^{(j)}_{t_N-t})\Big) \tag{4}$$

où les $N-$uplets $(\tilde{S}^{(j)}_{t_{i+1}-t},\ldots,\tilde{S}^{(j)}_{t_N-t})$ sont i.i.d selon la loi de $(S_{t_{i+1}-t},\ldots,S_{t_N-t})$.

Pour un produit faisant intervenir D sous-jacents, son portefeuille de couverture contient les D actifs et la quantité d'actifs d à détenir à l'instant t est donnée par

$$\frac{\partial v(t,S_{t_0},\cdots,S_{t_i},S_t)}{\partial S_{t,d}}$$

3 Architecture du Pricer Monte Carlo

3.1 Calcul des prix

L'ambition de ce projet est d'écrire un outil permettant de pricer n'importe quel produit dans un modèle de Black Scholes de dimension quelconque en ne spécifiant que 3 choses : le payoff de l'option, les paramètres du modèle et sa fonction de simulation. Pour ce faire nous allons implémenter trois classes : `MonteCarlo`, `Option`, `BlackScholesModel`.

Question 1. *Proposer une spécification d'une classe* `MonteCarlo` *permettant de réaliser le travail demandé.*

Question 2. *Proposer une architecture pour la classe* `Option`*, quelles fonctionnalités devra–t–elle implémenter ?*

Question 3. *Proposer une spécification de la classe* `BlackScholesModel` *permettant de simuler le sous–jacent. On s'assurera qu'il n'y ait pas de dépendance inutile entre les 3 classes.*

Prix à l'instant 0 Dans un premier temps, on se limitera au calcul du prix à l'instant 0. Implémenter la méthode `price` à la classe `MonteCarlo` permettant de calculer par une méthode de Monte Carlo le prix à l'instant 0 d'une option. Cette fonction calculera également la largeur de l'intervalle de confiance de l'estimateur à 95%. On rappelle que la variance de l'estimateur peut être approchée par

$$\xi_M^2 = \mathrm{e}^{-2rT}\left[\frac{1}{M}\sum_{j=1}^{M}(\varphi(S^{(j)}_{t_0},S^{(j)}_{t_1},\ldots,S^{(j)}_{t_N}))^2 - \left(\frac{1}{M}\sum_{j=1}^{M}\varphi(S^{(j)}_{t_0},S^{(j)}_{t_1},\ldots,S^{(j)}_{t_N})\right)^2\right]$$

et que l'intervalle de confiance à 95% est alors donné par

$$\left[\mathrm{e}^{-rT}\frac{1}{M}\sum_{j=1}^{M}\varphi(S^{(j)}_{t_0},S^{(j)}_{t_1},\ldots,S^{(j)}_{t_N}) \pm \frac{1.96\xi_M}{\sqrt{M}}\right]$$

Ecrire une méthode `MonteCarlo::price` qui fera appel aux méthodes `asset` et `payoff`.

Question 4. *Proposer des tests pour les différentes fonctionnalités et composants de votre code.*

Question 5. *Implémenter les différentes classes et le calcul du prix à l'instant* 0.

Prix à un instant $t > 0$ Une fois le calcul du prix à l'instant 0 testé et validé, vous pourrez passer au calcul du prix à tout instant $t > 0$.

Pour ce faire, il faut être capable de calculer l'estimateur (1). Cette espérance peut-être calculée par une méthode de Monte Carlo standard pourvu que l'on soit capable de générer des trajectoires i.i.d selon la loi conditionnelle de S sachant $\mathcal{F}_t$, i.e. grâce à la propriété (2) des trajectoires de la forme

$$(s_{t_0}, s_{t_1}, \ldots, s_{t_i}, s_t\tilde{S}^{(j)}_{t_{i+1}-t}, \ldots, s_t\tilde{S}^{(j)}_{t_N-t})$$

Question 6. *Modifier les spécifications des classes précédentes, pour pouvoir rajouter la simulation d'une trajectoire connaissant son passé jusqu'à une certaine date.*

Question 7. *Implémenter le calcul du prix à une date quelconque.*

Avant de continuer l'ajout de nouvelles fonctionnalités, il est essentiel réfléchir aux possibles optimisations de code (ordre des boucles, utilisation d'espaces de travail) tout en maintenant sa lisibilité. Pour ce faire, il est recommandé d'utiliser un outil dit de *profiling* : on pourra par exemple utiliser *gprof*.

1. Commpiler votre code avec l'option `-pg` (avec cmake `-DCMAKE_CXX_FLAGS=-pg`).
2. Lancer le code (ne pas s'inquiéter d'une exécution ralentie). Cela crée un fichier `gmon.out`.
3. Lancer la commande `gprof ./nom_executable gmon.out > profile.txt`.
4. Etudier le fichier `profile.txt`.

3.2 Calcul du delta

Dans cette partie, on cherche à écrire une méthode `MonteCarlo::delta` qui calcule le delta de l'option à un instant t à partir de la formule (4). Remarquons dans cette formule qu'il faut être capable de simuler 2 trajectoires utilisant les mêmes aléas Browniens mais shiftées l'une par rapport à l'autre. Considérons une trajectoire $(s_{t_0}, s_{t_1}, \ldots, s_{t_i}, s_t\tilde{S}^{(j)}_{t_{i+1}-t}, \ldots, s_t\tilde{S}^{(j)}_{t_N-t})$, nous souhaitons à partir de cette trajectoire construire pour $d = 1 \ldots D$ les trajectoires

$$(s_{t_0}, s_{t_1}, \ldots, s_{t_i}, g_{d,+}(s_t)\tilde{S}^{(j)}_{t_{i+1}-t}, \ldots, g_{d,+}(s_t)\tilde{S}^{(j)}_{t_N-t})$$
$$(s_{t_0}, s_{t_1}, \ldots, s_{t_i}, g_{d,-}(s_t)\tilde{S}^{(j)}_{t_{i+1}-t}, \ldots, g_{d,-}(s_t)\tilde{S}^{(j)}_{t_N-t})$$

où

$$g_{d,+}(x) = \begin{pmatrix} x_1 \\ \vdots \\ x_{d-1} \\ x_d(1+h) \\ x_{d+1} \\ \vdots \\ x_D \end{pmatrix} \qquad g_{d,-}(x) = \begin{pmatrix} x_1 \\ \vdots \\ x_{d-1} \\ x_d(1-h) \\ x_{d+1} \\ \vdots \\ x_D \end{pmatrix}$$

Question 8. *Implémenter une méthode* `BlackScholesModel::shift_asset` *et le calcul des delta.*

4 Simulation de la couverture

Pour les praticiens de la finance, parmi les critères qui interviennent dans le choix d'un modèle plutôt qu'un autre, la répartition du *P&L* (Profit and Loss, i.e. erreur de couverture) joue un rôle tout à fait central. Un modèle est jugé sur sa bonne capacité à couvrir les produits exotiques.

Calcul de P&L. Dans cette partie, nous allons supposer dans un premier temps avoir à disposition des trajectoires de marché sur une grille de temps $0 = \tau_0 < \cdots < \tau_H = T$ de pas T/H. On suppose d'une part que les paramètres de modèles associés à ces trajectoires de marché sont connus et que la grille régulière $\{\tau_0, \ldots, \tau_H\}$ des observations du marché satisfait $\{t_0, \ldots, t_N\} \subset \{\tau_0, \ldots, \tau_H\}$.

Le long de cette trajectoire, calculer le prix et le delta à chaque date τ_i et construire le portefeuille de couverture. Notons p_i et δ_i les prix et deltas calculés le long de cette trajectoire à la date τ_i alors l'évolution de la part investie au taux sans risque s'écrit.

$$\begin{cases} V_0 & = p_0 - \delta_0 \cdot S_0 \\ V_i & = V_{i-1} * \mathrm{e}^{\frac{r*T}{H}} - (\delta_i - \delta_{i-1}) \cdot S_{\tau_i}, \quad i = 1, \ldots, H \end{cases}$$

L'erreur de couverture est donnée par $P\&L = V_H + \delta_H \cdot S_{\tau_H} - \texttt{payoff}$.

Question 9. *Implémenter une méthode de calcul de P&L le long d'une trajectoire de marché.*

Simulation d'un marché. Dans le paragraphe, nous supposions avoir à disposition des trajectoires de marché, ce qui correspond en pratique à l'utilisation de données réelles qui n'ont aucune raison de suivre un modèle de Black Scholes.

Lors de la mise au point d'un outil de valorisation et de couverture, il est préférable de le tester d'abord sur des données de marché simulées pour lesquelles on connait les paramètres et le modèle utilisé. Le but de cette partie est donc de simuler des données de marché dans le modèle de Black Scholes multi–dimensionnel. Le marché évolue sous la probabilité historique ou objective, la dynamique du modèle de marché s'écrit

$$S_t^d = S_0^d \, \mathrm{e}^{(\mu^d - (\sigma^d)^2/2)t + \sigma^d L^d W_t}, \quad d = 1 \ldots D.$$

Le vecteur μ est connu sous le nom de tendance du modèle.

Question 10. *Ajouter le paramètre de tendance* `trend` *à la classe* `BlackScholesModel` *et implémenter une méthode* `BlackScholesModel::simul_market` *renvoyant une simulation du marché (c'est à dire une simulation du modèle sous la probabilité historique) avec un nombre de dates* `H`*, typiquement on prendra une date par jour ou par semaine.*

Problématique de gestion des dates. Pour la mise en œuvre du portefeuille de couverture, deux grilles de discrétisation entrent en jeu : la grille de pas T/N qui représente les dates de constatation du produit et la grille **plus fine** de pas T/H des dates de rebalancement du portefeuille, voir Figure 1.

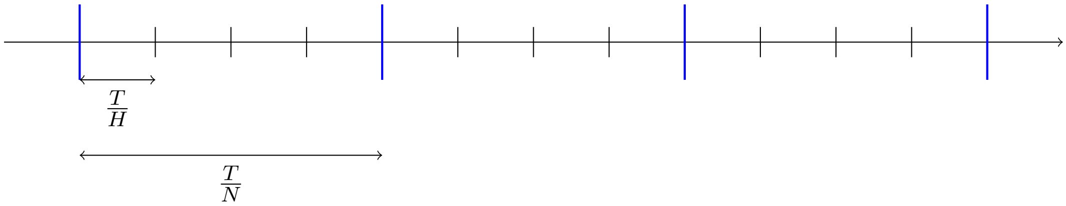

FIGURE 1 – Exemple avec $H = 12$ et $N = 4$.

5 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 différentes 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 de pricer fourni sur le kiosk.

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

Clé	Type
option size	int
spot	vector
maturity	float
volatility	vector
interest rate	float
correlation	float
trend	vector

Clé	Type
strike	float
option type	string
payoff coefficients	vector
timestep number	int
hedging dates number	int
fd step	float
sample number	long

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.

6 Produits à gérer

Voici la liste des options que votre pricer devra être capable d'évaluer (et de couvrir)

— Option panier (clé `basket`)

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

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

— Option asiatique discrète (clé `asian`)

$$\left(\sum_{d=1}^{D} \lambda_d \frac{1}{N+1} \sum_{i=0}^{N} S_{t_i,d} - K\right)_+$$

— Option performance sur panier (clé `performance`)

$$1 + \sum_{i=1}^{N} \left(\frac{\sum_{d=1}^{D} \lambda_d S_{t_i,d}}{\sum_{d=1}^{D} \lambda_d S_{t_{i-1},d}} - 1 \right)_+$$