Méthodes de Monte-Carlo et réduction de variance

TP 2

Exercice 1. On souhaite comparer l'efficacité de différentes méthodes de réduction de variance sur une option sur maximum discret de payoff

$$\left(K\left(\max_{t\in\{t_0,\dots,t_J\}} S_t\right) - S_T\right)_+$$

où $0 = t_0 < t_1 < \cdots < T_J = T$ et T sa maturité.
Pour la valorisation, on se place dans un modèle de Black-Scholes de dimension 1 sous la probabilité risque neutre, de sorte que la dynamique du sous-jacent est donnée par

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

où W est un mouvement Brownien réel standard, $r > 0$ est le taux d'intérêt, $\sigma > 0$ est la volatilité et S_0 est la valeur initiale du sous-jacent, encore appelé prix spot.
Pour les tests numériques, on prendra :

S_0	K	r	σ	T	J
100	0.95	0.02	0.25	2	24

Avec ces valeurs, on trouve comme prix 18.83 pour un intervalle de confiance de demi-largeur 0.12.
Remarque : Dans toutes les questions utilisant une méthode de Monte-Carlo, on prendra soin de fournir un intervalle de confiance.

1. Ecrire une fonction de simulation du sous-jacent.
2. Ecrire une fonction qui, étant donnée une trajectoire du modèle, calcule le payoff de l'option.
3. Ecrire une fonction calculant le prix de l'option par une méthode de Monte-Carlo standard à M tirages.
4. Même question que précédemment avec une technique de variables antithétiques.
5. Même question que précédemment en utilisant une technique de variable de contrôle linéaire adaptative avec comme contrôle $S_T - S_0\,\mathrm{e}^{rT}$.
6. Comparer la précision des résultats obtenus par les 3 méthodes pour différents strikes. Pour ce faire, tracer le prix et l'intervalle de confiance en fonction de K pour chacune des méthodes.

Exercice 2. Le but de cet exercice est de se familiariser avec la méthode l'*importance sampling* sur un exemple jouet. On considère une option d'achat dans le modèle de Black-Scholes

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

où W est un mouvement Brownien standard réel. On prendra comme paramètres

$$S_0 = 100,\ r = 0.03,\ \sigma = 0.2,\ T = 2,\ K = 120.$$

Le prix avec ces paramètres est 6.57.

D'après les résultats vus en cours, le prix de l'option d'achat vérifie l'égalité

$$\mathbb{E}[\mathrm{e}^{-rT}(S_T - K)_+] = \mathbb{E}\left[\mathrm{e}^{-rT}\left(S_0\,\mathrm{e}^{(r-\sigma^2/2)T+\sigma\sqrt{T}(G+\theta)} - K\right)_+ \mathrm{e}^{-\theta G-\theta^2/2}\right], \forall \theta \in \mathbb{R}$$

où G suit une loi normale centrée réduite. On introduit la fonction $\psi : \mathbb{R} \to \mathbb{R}$ définie par

$$\psi(g) = \mathrm{e}^{-rT}\left(S_0\,\mathrm{e}^{(r-\sigma^2/2)T+\sigma\sqrt{T}g} - K\right)_+.$$

Bien que le prix de l'option d'achat soit donné par une formule fermée, nous allons le calculer par une méthode de Monte Carlo. Dans ce cadre, nous cherchons à déterminer la valeur θ^* du paramètre θ qui minimise

$$v(\theta) = \mathbb{E}\left[\psi(G)^2\,\mathrm{e}^{-\theta G+\theta^2/2}\right].$$

La fonction v est fortement convexe et de classe C^∞, de plus

$$v'(\theta) = \mathbb{E}\left[(\theta - G)\psi(G)^2\,\mathrm{e}^{-\theta G+\theta^2/2}\right].$$

On note $U : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ la fonction

$$U(\theta, g) = (\theta - g)\psi(g)^2\,\mathrm{e}^{-\theta g+\theta^2/2}.$$

de sorte que $\mathbb{E}[U(\theta, G)] = v'(\theta)$.

1. Récupérer le squelette. Il se compile en utilisant *CMake*. Ne pas oublier de définir la variable `CMAKE_PREFIX_PATH` sur la ligne de commande en indiquant le chemin vers la librairie PNL.
2. Dans cette question, on cherche à approcher $\theta^\star = \arg\min_{\theta\in\mathbb{R}} v(\theta)$. Soit $\theta_0 = 0$ et $\alpha_0 = 0$, on définit les suites de variables aléatoires $(\theta_n)_n \in \mathbb{R}^\mathbb{N}$ et $(\alpha_n)_n \in \mathbb{N}^\mathbb{N}$ par

$$\begin{cases} & \theta_{n+\frac{1}{2}} = \theta_n - \gamma_{n+1}U(\theta_n, G_{n+1}), \\ \text{si } (\theta_{n+\frac{1}{2}})^2 \le \log(\alpha_n + 1) & \theta_{n+1} = \theta_{n+\frac{1}{2}} \quad \text{et} \quad \alpha_{n+1} = \alpha_n, \\ \text{si } (\theta_{n+\frac{1}{2}})^2 > \log(\alpha_n + 1) & \theta_{n+1} = \theta_0 \quad \text{et} \quad \alpha_{n+1} = \alpha_n + 1. \end{cases} \tag{1}$$

   $(G_n)_n$ est une suite i.i.d de loi normale centrée et réduite et $\gamma_n = \frac{\gamma}{(n+1)^\beta}$, avec $1/2 < \beta \le 1$. La suite $(\theta_n)_n$ converge p.s. vers θ^* pour toute valeur initiale de θ_0. En pratique, on pourra prendre $\beta = 0.75$.

Implémenter le calcul de la suite $(\theta_n)_n$.

```
void MonteCarlo::is(PnlVect *theta, double gamma, int n, PnlRng *rng);
```

En sortie, le paramètre `theta` contient l'ensemble des valeurs $\theta_1, \ldots, \theta_n$.

3. Tracer l'évolution de la suite $(\theta_n)_n$ en fonction de n. On tracera plusieurs graphiques pour différentes valeurs de $\gamma = 50, 5, 0.5, 0.05, 0.01$.
4. Implémenter une méthode de Monte Carlo utilisant l'approximation θ_n de θ^*.

```
void MonteCarlo::mcis(double &prix, double &stddev, double theta_n, PnlRng *rng);
```

5. Comparer la précision avec celle de la méthode de Monte Carlo standard déjà implémentée dans le squelette.
6. Reprendre la question précédente avec la suite

$$\left(\frac{1}{n}\sum_{\ell=100}^{n}\theta_\ell\right)_{n\geq 100}$$

7. Tracer sur un même graphique, la convergence des 2 méthodes de Monte-Carlo `mc` et `mcis`.
8. Implémenter une version adaptive de l'estimateur de la question 4.