

Pricing d'options asiatiques

T.AHN, T.AMADEI, T.METZ, T.SALOMON

Projet - Simulation and Monte Carlo Methods

25 avril 2023

Introduction

- *Objectif* : Estimer $C = \mathbb{E} \left[e^{-rT} \cdot \max \left(\frac{1}{k} \sum_{i=1}^k S(t_i) - K, 0 \right) \right]$
 → avec différentes méthodes
- *Outils de comparaison* : MSE, CPU time et variance (empirique) des estimateurs
- **Multilevel Monte Carlo path simulation**, Michael B. Giles

Sommaire

- 1 Prérequis
- 2 Simulations du sous-jacent
- 3 Approximations du prix d'une option asiatique
- 4 Comparaison des approximations du prix en termes de MSE et CPU time
- 5 Bonus : Utilisation d'un pont brownien
- 6 Appendix

Prérequis

Produits dérivés et options

Les produits dérivés, en particulier les options peuvent être utilisés pour:

- **Couverture de risque**
- **Spéculation**
- **Arbitrage**

Prérequis

Option asiatique

- *Objectif* : Estimer le prix d'une option asiatique

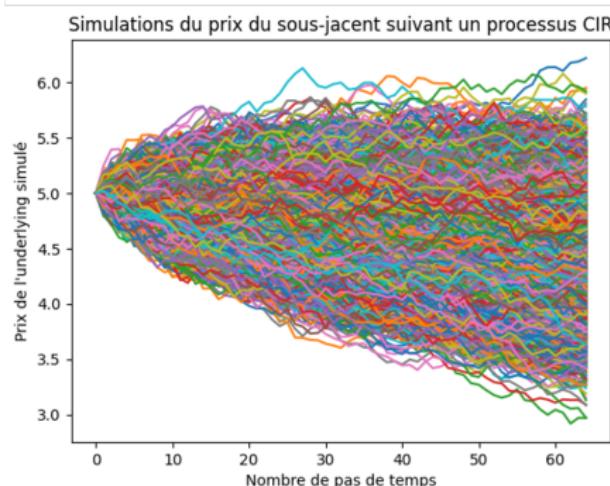
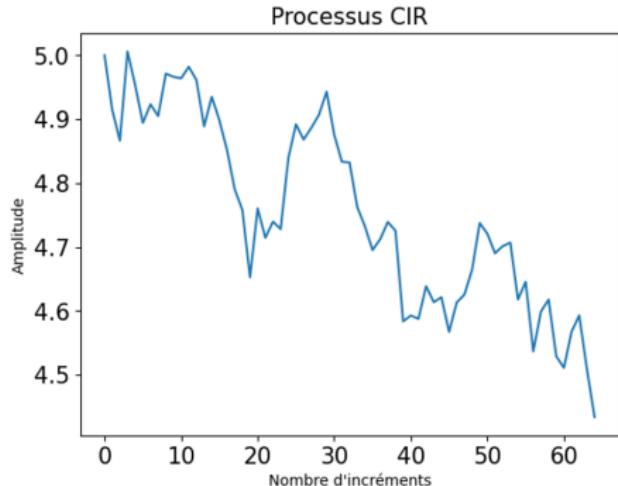
$$C = \mathbb{E} \left[e^{-rT} \cdot \max \left(\frac{1}{k} \sum_{i=1}^k S(t_i) - K, 0 \right) \right]$$

- Ce type d'option permet de se couvrir des manipulations possibles du sous-jacent à maturité car le payoff dépend du prix moyen sur une période donnée.
- Les options asiatiques sont aussi plus robustes à la volatilité du sous-jacent donc son prix est plus faible que celui d'un call européen (*vanille*):

$$C = \max(S_T - K, 0)$$

Simulation du sous-jacent

- Valeur à estimer: $C = \mathbb{E} \left[e^{-rT} \cdot \max \left(\frac{1}{k} \sum_{i=1}^k S(t_i) - K, 0 \right) \right]$
- Processus stochastique **CIR** (*Cox-Ingersoll-Ross*)
- Équation différentielle stochastique: $dS_t = \alpha(b - S_t)dt + \sigma\sqrt{S_t}dW_t$,
 W_t : mouvement brownien.



Simulation du sous-jacent

- Valeur à estimer: $C = \mathbb{E} \left[e^{-rT} \cdot \max \left(\frac{1}{k} \sum_{i=1}^k S(t_i) - K, 0 \right) \right]$
- Processus stochastique **CIR** (*Cox-Ingersoll-Ross*)
- Équation différentielle stochastique: $dS_t = \alpha(b - S_t)dt + \sigma\sqrt{S_t}dW_t$,
 W_t : mouvement brownien.
- Méthode de discréétisation **d'Euler-Maruyama**
- $S_{t_{i+1}} = S_{t_i} + \alpha(b - S_{t_i})(t_{i+1} - t_i) + \sigma\sqrt{S_{t_i}}(W_{t_{i+1}} - W_{t_i})$
avec $t_i = i \frac{T}{n}$

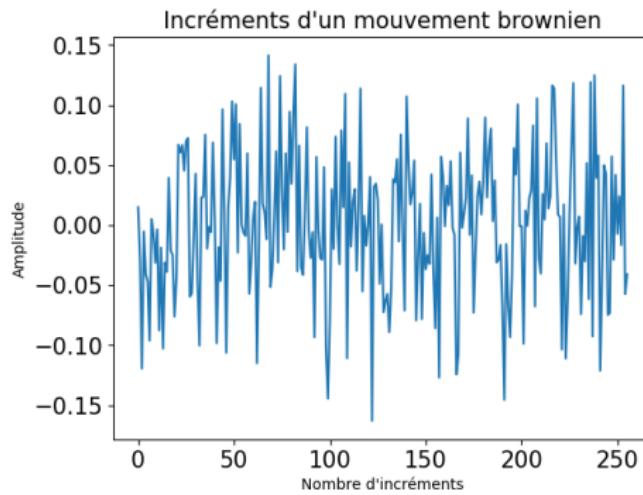
Simulation des incrément browniens

- $\forall t \in \{1, \dots, T\}, W_{t_i+1} - W_{t_i} \sim \mathcal{N}(0, \frac{T}{n})$
 \hookrightarrow simuler des $\mathcal{N}(0, 1)$ et les multiplier par $\sqrt{\frac{T}{N}}$
- Monte-Carlo standard \Rightarrow Box-Muller amélioré (NumPy)
- Quasi Monte-Carlo \Rightarrow séquence de Sobol combinée avec
 - ① Acceptation-rejet
 - ② Box-Muller
 - ③ Inversion de la CDF

Simulation des incrément browniens

Box-Muller amélioré (NumPy)

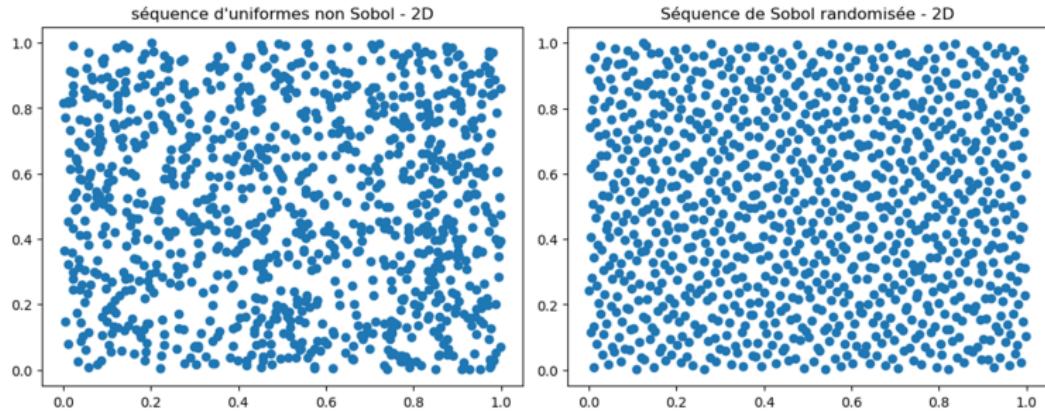
- ① $U, V \sim \mathcal{U}[-\infty, \infty]$
- ② si $S = U^2 + V^2 > 1$: recommencer à l'étape 1
- ③
$$\begin{cases} X = U \cdot \sqrt{-2 \frac{\log(S)}{S}} \\ Y = V \cdot \sqrt{-2 \frac{\log(S)}{S}} \end{cases}$$



Simulation des incrément browniens

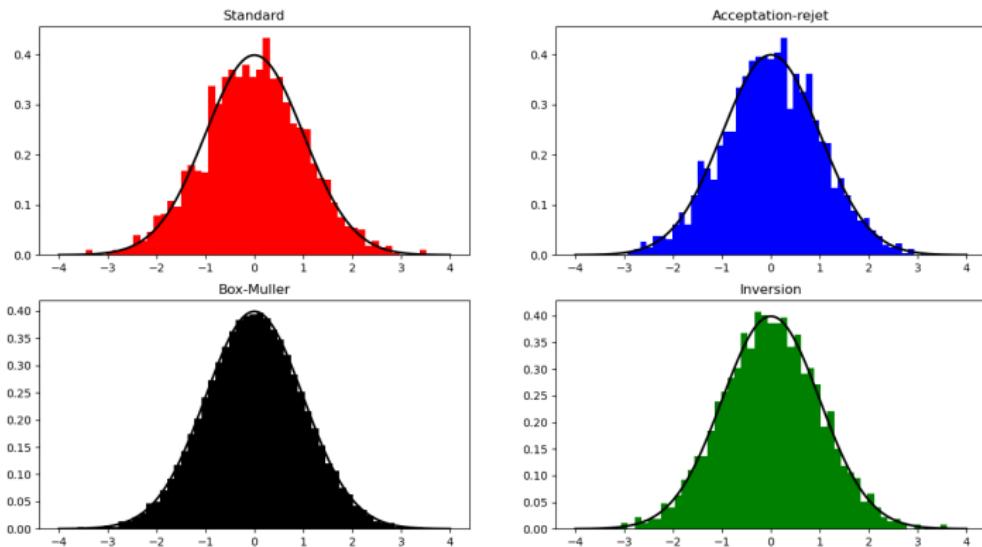
Séquence de Sobol randomisée

- Random Sobol: $\forall n \in \mathbb{N}, v_n = (u_n + w_n) \bmod 1$
 - u_n : uniforme Sobol
 - $w_n \sim \mathcal{U}[0, 1]$



Simulation des incrément browniens

Comparaison des méthodes de simulation de $\mathcal{N}(0, 1)$



Simulation des incrément browniens

Méthode Box-Muller + Random Sobol

- Simuler une séquence de Sobol randomisée 2D de dimension $\frac{N}{2}$

$$\forall i \in \{1, \dots, N\}, \begin{cases} X = \sqrt{-2 \log(U_{i_1})} \cdot \cos(2\pi U_{i_2}) \\ Y = \sqrt{-2 \log(U_{i_2})} \cdot \cos(2\pi U_{i_1}) \end{cases}$$

- $\forall i \in \{1, \dots, N/2\}, U_i = \begin{pmatrix} U_{i_1} \\ U_{i_2} \end{pmatrix} \Rightarrow U_{i_1} \text{ et } U_{i_2} \text{ sont indépendants}$

Approximations du prix d'une option asiatique

- Méthode Monte-Carlo Standard (MC)
- Méthode Quasi Monte-Carlo (QMC)
- Méthode Multi Level Monte-Carlo (MLMC)
- Méthode Multi Level Quasi Monte-Carlo (QMC MLMC)

Méthode Monte-Carlo Standard et Quasi Monte-Carlo

- Étapes :
 - ➊ Simulations de N_{sim} CIR (i.e N_{sim} trajectoires du prix du sous-jacent)
 - ↪ MC : incrément du mouvement brownien obtenus par méthode de Monte Carlo standard
 - ↪ QMC : incrément du mouvement brownien obtenus par méthode de Quasi Monte Carlo
 - ➋ N_{sim} approximations du prix de l'option.
 - ➌ Le prix final de l'option est la moyenne des N_{sim} approximations du prix de l'option.

Méthode Multi-Level Monte Carlo

Présentation de la méthode

- *Idée* : obtenir une estimation plus précise d'une quantité d'intérêt en utilisant des simulations à plusieurs niveaux de granularité
 - ↪ quantité d'intérêt : prix de l'option asiatique
- Méthode MLMC utilise 2 niveaux de granularité :
 - niveau de granularité "coarse" : discrétisation plus grossière (pas de temps plus grand)
 - niveau de granularité "fine" : discrétisation plus fine (pas de temps plus petit)

Méthode Multi-Level Monte Carlo

Présentation de la méthode

- Pour chaque niveau de granularité : un grand nombre de simulations du prix du sous jacent (processus CIR) est effectué et le prix de l'option est calculé à partir de ces simulations
- Prix final de l'option = différence entre le prix estimé à partir des simulations de niveau "fine" et le prix estimé à partir des simulations de niveau "coarse".

Méthode Multi-Level Monte Carlo

Présentation de la méthode

- Soit P_l approximation du prix de l'option au niveau l ie son approximation avec une discrétisation de pas $h_l = \frac{1}{M^l}$.

$$\mathbb{E}[P_L] = \mathbb{E}[P_0] + \sum_{l=1}^L \mathbb{E}[P_l - P_{l-1}]$$

Idée de MLMC : Estimer indépendamment chacune des espérances d'une manière qui minimise la variance globale pour un coût de calcul donné.

- \hat{Y} l'estimateur MLMC du prix de l'option
 \hat{Y}_l pour $l > 0$ un estimateur de $\mathbb{E}[P_l - P_{l-1}]$ utilisant N_l échantillons.

$$\hat{Y} = \sum_{l=0}^L \hat{Y}_l = \sum_{l=0}^L \frac{1}{N_l} \sum_{i=1}^{N_l} (P_l^i - P_{l-1}^i)$$

Méthode Multi-Level Monte Carlo

Choix des paramètres

- Paramètre M

$M=4$ fixé

↪ choix effectué par l'auteur

- Paramètre L

↪ le plus petit possible (pour diminuer la CPU time) telle que l'estimateur \hat{Y}_L soit convergent.

Le critère utilisé (biais suffisamment petit):

$$|\hat{Y}_L - \frac{\hat{Y}_{L-1}}{M}| < \frac{1}{\sqrt{2}}(M^2 - 1)\epsilon$$

avec ϵ un terme d'erreur choisi

Méthode Multi-Level Monte Carlo

Choix des paramètres

- Paramètres N_l pour $l \in [0, L]$

Sélection de ce paramètre est cruciale car un choix optimal de N_l permet de réduire la variance

- Méthode 1 : $N_l \propto \sqrt{V_l h_l} = \sqrt{\frac{V_l}{M^l}}$
↪ vu en cours
- Méthode 2 : $N_l = 2\epsilon^{-2} \sqrt{V_l \cdot h_l} \left(\sum_{l=0}^L \sqrt{\frac{V_l}{h_l}} \right)$
↪ expression proposée par Michael B. Giles

⇒ les paramètres L et N_l ont été choisis optimalement en implémentant un algorithme proproposé par Michael B. Giles

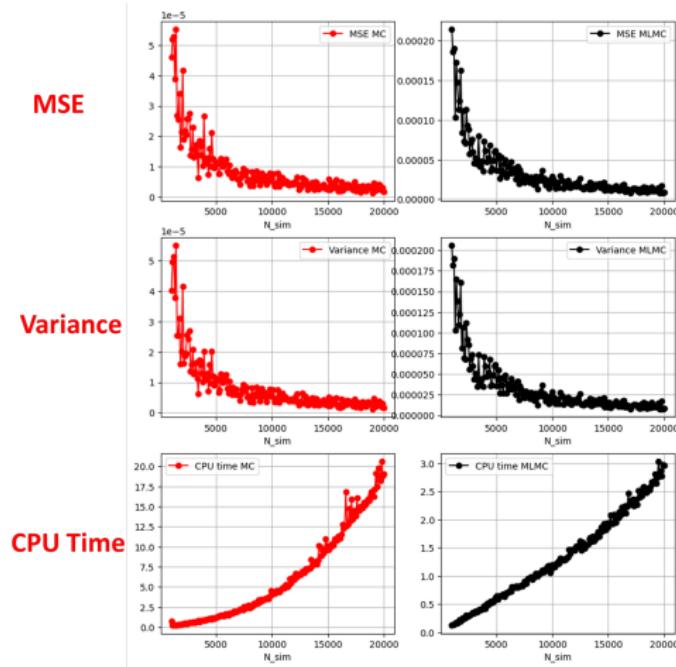
Méthode Multi-Level Quasi Monte-Carlo

QMC_MLMC : même principe que la méthode MLMC à une différence près

↪ incrément du mouvement brownien dans la simulation du processus CIR (prix du sous-jacent) obtenus par méthode de Quasi Monte-Carlo

Comparaison des approximations du prix en termes de MSE et CPU time

Comparaison entre MC Standard et MLMC avec paramètres non optimisés



Comparaison des approximations du prix en termes de MSE et CPU time

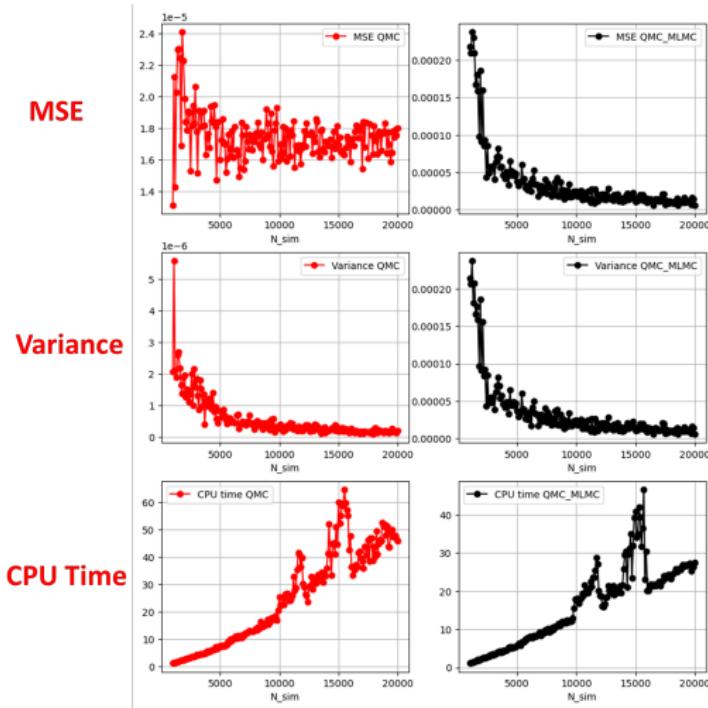
Comparaison entre MC Standard et MLMC avec paramètres optimisés

- Pour les faibles valeurs de tolérance ϵ , MLMC réduit significativement la variance
- Nécessité des valeurs optimales des N_i
- Dernier niveau d' ϵ : temps d'exécution très grand mais variance et MSE grandement réduites
- L' ϵ le plus judicieux semble être 10^{-3}

Monte-Carlo standard // MLMC Adaptatif			
ϵ	MSE	Variance	CPU Time (s)
3.10^{-4}	$1,2.10^{-4} // 7,3.10^{-7}$	$1,1.10^{-4} // \textcolor{red}{5,1.10^{-8}}$	$8,1 // 19110,2$
4.10^{-3}	$1,1.10^{-4} // 1,2.10^{-5}$	$1,2.10^{-4} // 1.2.10^{-6}$	$4,9 // 16,7$
7.10^{-3}	$2,0.10^{-4} // 1,7.10^{-5}$	$1,7.10^{-4} // 1,7.10^{-5}$	$3,9 // 6,9$

Comparaison des approximations du prix en termes de MSE et CPU time

Comparaison entre QMC et QMC_MLMC avec paramètres non optimisés



Comparaison des approximations du prix en termes de MSE et CPU time

Comparaison entre QMC et QMC_MLMC avec paramètres optimisés

- La méthode Quasi Monte-Carlo réduit considérablement la variance dans les deux cas
- Meilleure méthode pour faire baisser la variance : QMC MLMC avec paramètres optimisés

Quasi Monte-Carlo // QMC MLMC Adaptatif			
ϵ	MSE	Variance	CPU Time (s)
3.10^{-4}	$4.3.10^{-6} // 7, 3.10^{-7}$	$6.1.10^{-7} // 1, 4.10^{-8}$	71,4 // 38678,6
4.10^{-3}	$1,7.10^{-5} // 6, 5.10^{-6}$	$5,1.10^{-6} // 2, 6.10^{-7}$	40,5 // 59,7
7.10^{-3}	$1,8.10^{-5} // 1, 5.10^{-5}$	$2,1.10^{-5} // 1, 0.10^{-6}$	7,7 // 43,1

Utilisation du pont brownien

Séquence de Van der Corput

- L'objectif est de simuler un mouvement brownien $(W_{t_1}, W_{t_2}, \dots, W_{t_k})$ selon un pont brownien en suivant une séquence de Van der Corput de base 2.
- Séquence de Van der Corput en base 2

Résultat : Les t_i sont ordonnés tels que $t_k, t_{\frac{k}{2}}, t_{\frac{k}{4}}, \dots$

Utilisation du pont brownien

Le pont brownien

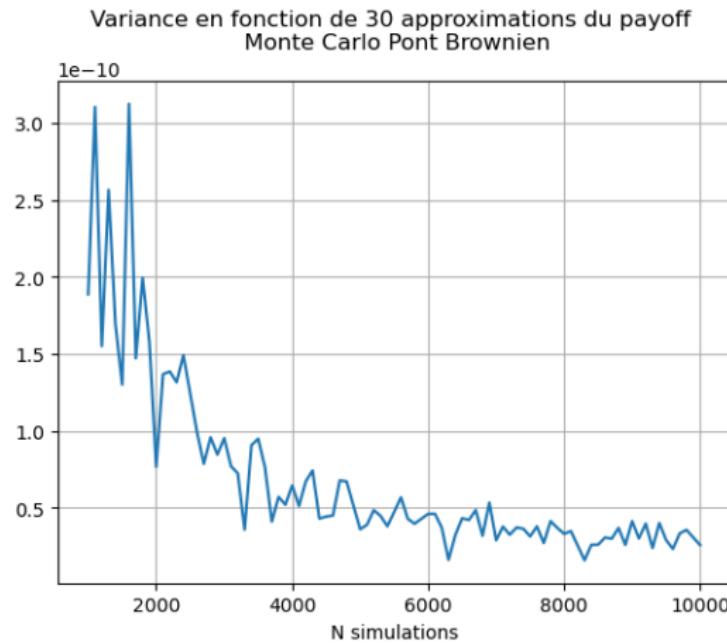
- Il faut désormais générer les :

$$W_{t_i} | W_{t_{i-1}}, W_{t_{i+1}} \sim \mathcal{N}\left(\frac{(t_{i+1}-t_i)W_{t_{i-1}} + (t_i-t_{i-1})W_{t_{i+1}}}{t_{i+1}-t_{i-1}}, \frac{(t_{i+1}-t_i)(t_i-t_{i-1})}{t_{i+1}-t_{i-1}}\right)$$

suivant l'ordre d'indice donné par la séquence de Van der Corput, soit
 W_{t_k} puis $W_{t_{\frac{k}{2}}}$, $W_{t_{\frac{k}{4}}}$, etc

Utilisation du pont brownien

Résultat



Conclusion

- MLMC = utiliser une somme télescopique d'estimateurs de plus en plus complexes
- Approximation sans biais + Réduction de variance de l'estimateur final
- Possibilité de combiner MLMC et des méthodes de réduction de variance
- Arbitrage réduction de variance et temps de calcul

Simulation des incrément browniens

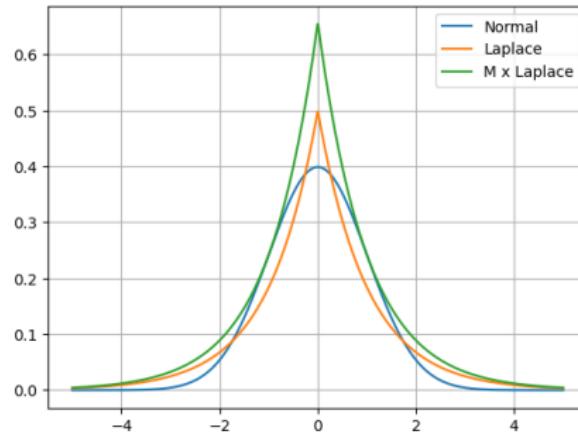
Méthode d'acceptation-rejet

① $X \sim \text{Laplace}$

② $U \sim \mathcal{U}[0, 1]$

③ si $U \leq \frac{f(X)}{M \cdot g(X)}$ on accepte $X \sim \mathcal{N}(0, 1)$, sinon répéter depuis 1

$$\hookrightarrow M = \sqrt{\frac{2e}{\pi}}$$



Simulation des incrément browniens

Méthode Inversion + Random Sobol

- Simuler $(U_i)_{1 \leq i \leq N}$: séquence de Sobol randomisée
- $\forall i \in \{1, \dots, N\}, X_i = F^{-1}(U_i) \sim \mathcal{N}(0, 1)$
- F^{-1} = approximation de la fonction quantile de $\mathcal{N}(0, 1)$

Démonstration : $N_l \propto \sqrt{V_l h_l}$

$$\mathbb{V}[\hat{Y}] = \sum_{l=1}^L \mathbb{V}[\hat{Y}_l] = \sum_{l=1}^L \frac{V_l}{N_l}$$

Nous souhaitons minimiser la variance. Le problème d'optimisation sous contrainte est le suivant :

$\min(\mathbb{V}[\hat{Y}])$ sous contrainte $\sum_{l=1}^L C_l N_l = A$ avec C_l le cout CPU par échantillon et A une constante représentant le cout total

$$\mathcal{L} = \sum_{l=1}^L \frac{V_l}{N_l} + \lambda \left(\sum_{l=1}^L C_l N_l - A \right)$$

$$\frac{\partial \mathcal{L}}{\partial N_l} = -\frac{V_l}{N_l^2} + \lambda C_l$$

$$\text{Donc : } N_l = \sqrt{\frac{V_l}{\lambda C_l}}$$

$$C_l \text{ peut être approximé par } M^l \text{ donc } N_l = \sqrt{\frac{V_l}{\lambda M^l}} \propto \sqrt{V_l h_l}$$