

Options Asiatiques

Sixtine Sphabmixay

Janvier 2024

Comment établir le prix d'une option asiatique ?

- Qu'est-ce qu'une option asiatique ?
- Article de Bernard Lapeyre : Comment estimer $\frac{1}{T} \int_0^T S_t dt$?
 - Méthode Monte-Carlo
- Article de Rogers et Shi : Résoudre une équation différentielle partielle
 - Méthode de différences finies

Définition

Qu'est-ce qu'une option asiatique ?

- Différence avec les options européennes et américaines
- Différents prix d'exercices
 - Call avec prix d'exercice fixé K : $(\frac{1}{T} \int_0^T S_t dt - K)_+$
 - Put avec prix d'exercice fixé K : $(K - \frac{1}{T} \int_0^T S_t dt)_+$
 - Call avec prix d'exercice 'flottant' : $(S_T - \frac{1}{T} \int_0^T S_t dt)_+$
 - Put avec prix d'exercice 'flottant' : $(\frac{1}{T} \int_0^T S_t dt - S_T)_+$

Pourquoi acheter une option asiatique ?

- Réduction de la volatilité quotidienne
- Réduction des risques de manipulation du marché
- Alignement avec les objectifs à long terme
- Economie de coûts

Modèle de Black-Scholes

On représente le marché par l'ensemble $(\Omega, \mathbb{F}, (\mathcal{F}_t)_t, \mathbb{P})$ où $(\mathcal{F}_t)_t$ est une filtration représentant l'information disponible et \mathbb{P} la mesure de probabilité du monde 'réel'. Ce marché est composé de :

- Un actif sans risque de prix S_t^0 au temps t et tel que $dS_t^0 = rS_t^0 dt$
- Un actif risqué de prix S_t au temps t et tel que $dS_t = S_t(rdt + \sigma dB_t)$.

Prix d'une option asiatique

Le prix d'une option asiatique est donné par la formule suivante dans le cas d'un call avec prix d'exercice fixé : (moyenne actualisée du payoff)

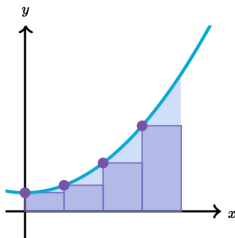
$$e^{-rT} \mathbb{E} \left(\left(\frac{1}{T} \int_0^T S_t dt - K \right)_+ \right)$$

Méthode Monte Carlo

- Utilisée pour estimer $M := \mathbb{E}(g(X))$
- $M \approx \frac{1}{n} \sum_{k=1}^n g(X_k)$ où (X_1, \dots, X_n) sont n réalisations iid de X

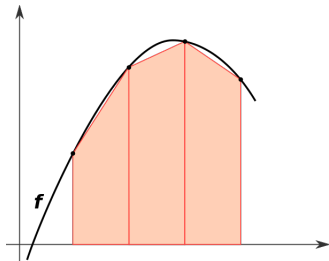
Méthode des rectangles

$$Y_T = \int_0^T S_t dt$$



On découpe l'intervalle $[0, T]$ en N morceaux de taille $h = t_{k+1} - t_k$ où $t_k = \frac{Tk}{N}$. On somme les aires des N rectangles obtenus pour obtenir une approximation de $\int_0^T f(x)dx$. Ainsi $\int_0^T f(x)dx = \sum_{k=1}^{n-1} f(t_k)(t_{k+1} - t_k)$.

Méthode des Trapèzes



Idée de la méthode : Approximer l'aire sous la courbe f en construisant des trapèzes. On découpe l'intervalle $[0, T]$ en N morceaux de taille $h = t_{k+1} - t_k$ où $t_k = \frac{Tk}{N}$. L'aire d'un trapèze est donnée par $h \frac{f(t_k) + f(t_{k+1})}{2}$ où $h = t_{k+1} - t_k$. On somme les aires des N trapèzes obtenus pour obtenir une approximation de $\int_0^T f(x) dx$

Article de Lapeyre : Premier et deuxième schéma

On applique ces deux méthodes pour estimer $Y_T = \int_0^T S_t dt$:

Premier schéma :

$$\begin{aligned} Y_T^{r,N} &= \sum_{k=0}^{N-1} S_{t_k} (t_{k+1} - t_k) \\ &= h \sum_{k=0}^{N-1} S_{t_k} \end{aligned}$$

Pour le deuxième schéma :

$$\frac{1}{T} \sum_{k=0}^{N-1} h \frac{S_{t_k} + S_{t_{k+1}}}{2} = \frac{1}{T} \sum_{k=0}^{N-1} \frac{h S_{t_k}}{2} (e^{\sigma(W_{t_{k+1}} - W_{t_k}) - \frac{\sigma^2}{2}h + rh} + 1)$$

soit

$$Y_T^{e,N} = \frac{h}{T} \sum_{k=0}^{N-1} S_{t_k} \left(1 + \frac{rh}{2} + \sigma \frac{W_{t_{k+1}} - W_{t_k}}{2} \right)$$

Article de Lapeyre : Troisième schéma

On utilise le fait qu'un mouvement brownien est un processus gaussien. En utilisant que $S_u = S_{t_k} e^{\sigma(W_u - W_{t_k}) + (r - \frac{\sigma^2}{2})(u - t_k)}$ pour tout $k \in \{0, \dots, n-1\}$, $t_k \leq u \leq t_{k+1}$ on obtient que

$$\frac{1}{T} \int_0^T S_t dt = \frac{1}{T} \sum_{k=0}^{N-1} S_{t_k} \int_{t_k}^{t_{k+1}} e^{\sigma(W_u - W_{t_k}) + (r - \frac{\sigma^2}{2})(u - t_k)} du$$

on aboutit au dernier schéma :

$$Y_T^{p,N} = \frac{1}{T} \sum_{k=0}^{N-1} S_{t_k} \left(h + \frac{rh^2}{2} + \sigma \int_{t_k}^{t_{k+1}} (W_u - W_{t_k}) du \right)$$

On peut alors utiliser la méthode de Monte Carlo pour déterminer le prix d'une option : $\frac{e^{-rT}}{M} \sum_{j=1}^M (Y_{T_j}^{p,N} - K)_+$

Résultats Python

Simulation du prix d'un call asiatique avec prix d'exercice fixé avec $r = 0.1$, $T = 1$, $N = 50$, $K = S_0 = 100$, $M = 10^5$. Lorsque $\sigma = 0.2$, le prix théorique est de 7.04.

σ	Pricing Riemann scheme	C.I	Error	Variance
0.05	4.63	[4.62, 4.65]	0.0083	6.96
0.2	6.92	[6.86, 6.97]	0.026	70.7
0.3	8.93	[8.86, 9.01]	0.039	153.23

σ	Pricing Trapez scheme	C.I	Error	Variance
0.05	4.73	[4.71, 4.74]	0.0084	7.02
0.2	6.98	[6.93, 7.03]	0.027	71.01
0.3	8.96	[8.85, 9.0]	0.039	151.67

σ	Pricing BS scheme	C.I	Error	Variance
0.05	4.74	[4.72, 4.75]	0.0085	7.25
0.2	7.07	[7.01, 7.12]	0.027	73.79
0.3	9.04	[8.96, 9.12]	0.04	157.16

Réduction de la variance

Une méthode efficace pour réduire la variance est celle de la variable de contrôle.

- utilise une autre variable aléatoire X dont on connaît l'espérance, et une constante $b \in \mathbb{R}$ telle que $\mathbb{V}(Y_T + b(X - \mathbb{E}(X))) \ll \mathbb{V}(Y_T)$.
- utilise la corrélation de X avec la variable Y_T
- L'article suggère d'étudier la variable $Z_T := \exp\left(\frac{1}{T} \int_0^T \log(S_u) du\right)$
- $Z_T = S_0 e^{\frac{T}{2}(r - \frac{\sigma^2}{2})} e^{\frac{\sigma}{T} \int_0^T (T-u) dW_u}$
- Comme $\frac{\sigma}{T} \int_0^T (T-u) dW_u \sim \mathcal{N}(0, \frac{\sigma^2 T}{3})$
- On peut donc prendre comme variable de contrôle

$$\hat{Z} = e^{-rT} (S_0 e^{(r - \frac{\sigma^2}{2})\frac{T}{2} + \frac{\sigma}{T} \int_0^T W_u du} - K)_+$$

Réduction de la variance

Même difficulté que précédemment pour simuler $\int_0^T W_u du$: on utilise les trois schémas mentionnés.

- $Z_T^{r,N} = e^{-rT} (S_0 e^{(r - \frac{\sigma^2}{2})\frac{T}{2} + \frac{\sigma}{T} \sum_{k=0}^{N-1} h W_{t_k}} - K)_+$
- $Z_T^{e,N} = e^{-rT} (S_0 e^{(r - \frac{\sigma^2}{2})\frac{T}{2} + \frac{\sigma}{T} \sum_{k=0}^{N-1} \frac{h}{2} (W_{t_k} + W_{t_{k+1}})} - K)_+$
- $Z_T^{p,N} = e^{-rT} (S_0 e^{(r - \frac{\sigma^2}{2})\frac{T}{2} + \frac{\sigma}{T} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} W_u du} - K)_+$

Résultats Python : Réduction de la variance

En utilisant les mêmes paramètres $r = 0.1$, $T = 1$, $N = 50$,
 $K = S_0 = 100$, $M = 10^5$:

σ	Pricing first scheme	C.I	Error	Variance	MC Variance
0.05	4.63	[4.63, 4.634]	0.00015	0.002	6.96
0.2	6.99	[6.988, 6.992]	0.0014	0.15	70.7
0.3	9.02	[9.015, 9.025]	0.0027	0.68	153.23

σ	Pricing Trapez scheme	C.I	Error	Variance	MC Variance
0.05	4.72	[4.72, 4.727]	0.00018	0.0028	7.05
0.2	6.98	[6.981, 6.987]	0.0011	0.12	71.38
0.3	8.96	[8.959, 8.968]	0.0023	0.55	153.57

σ	Pricing BS scheme	C.I	Error	Variance	MC Variance
0.05	4.72	[4.724, 4.725]	0.00015	0.0024	7.23
0.2	7.04	[7.039, 7.044]	0.0013	0.16	73.79
0.3	9.05	[9.049, 9.06]	0.0027	0.732	157.16

Article de Rogers et Shi

PDE

Rogers et Shi ont établi que

$$e^{-rT} \mathbb{E} \left(\int_0^T (S_u - K) \frac{du}{T} \right)_+ = S_0 f(0, \frac{K}{S_0})$$

où f est solution de

$$\frac{\partial f}{\partial t} + \mathcal{G}f = 0$$

avec

$$\mathcal{G} = \frac{\sigma^2 x^2}{2} \frac{\partial^2}{\partial x^2} - (rx + \rho_t) \frac{\partial}{\partial x}$$

et

$$f(T, x) = \max(0, -x)$$

Etape 1 : Maillage

- On considère le maillage suivant

$$\{(t_n, x_i) := (nh, \alpha + i\delta); 0 \leq n \leq m, 0 \leq i \leq l+1\}$$

où $h = \frac{T}{m}$ et $\delta = \frac{\beta - \alpha}{l+1}$.

- On approxime $F := (f(t, x))_{(t,x) \in [0,T] \times [\alpha,\beta]}$ par $\tilde{F} := (f_i^n)_{0 \leq i \leq l+1, 0 \leq n \leq m}$ tel que

$$f_i^n = f(t_n, x_i)$$

Etape 2 : Initialisation

- Pour $x \leq 0$

$$f(t, x) = e^{-r(T-t)} \left(\frac{e^{r(T-t)} - 1}{rT} - x \right)$$

- Condition au bord : $f_0^n = f(t_n, \alpha) = e^{-r(T-t_n)} \left(\frac{e^{r(T-t_n)} - 1}{rT} - \alpha \right)$
- Condition terminale : $f_i^m = f(T, x_i) = \max(0, -x_i)$

Méthode de différences finies

Etape 3 : Itération En appliquant la méthode de Crank-Nicholson pour estimer \mathcal{G} , et une méthode implicite pour estimer $\frac{\partial f}{\partial t}(t_n, x_i)$ on est amené à résoudre à chaque $n = m - 1, \dots, 0$ l'équation suivante :

$$Xf^n = y$$

où $X = (I - \frac{1}{2}hA)$ et $y = (I + \frac{1}{2}hA)f^{n+1} + h\frac{1}{2}(g^n + g^{n+1})$.

- A est une matrice tridiagonale $\mathbb{R}^{I \times I}$ de coefficients

$$a_{i,i-1} = \frac{\sigma^2(x_i)}{2\delta^2} - \frac{b(x_i)}{2\delta}$$

$$a_{i,i} = -\frac{\sigma^2(x_i)}{\delta^2}$$

$$a_{i,i+1} = \frac{\sigma^2(x_i)}{2\delta^2} + \frac{b(x_i)}{2\delta}$$

avec $\sigma(x) := \sigma x$ et $b(x) := -(rx + \rho_t)$

g^n est dans \mathbb{R}^I et est tel que

$$g_i^n = 0$$

pour tout $i \in \{2, \dots, I\}$ et

$$g_1^n = \left(\frac{\sigma^2(x_1)}{2\delta^2} - \frac{b(x_1)}{2\delta} \right) \left(\frac{e^{r(T-t_n)} - 1}{rT} - \alpha \right)$$

- **Etape 4 : Arrêt**

La fonction renvoie le coefficient de \tilde{F} correspondant à $f(0, \frac{K}{S_0})$

Résultats Python : PDE

En prenant $r = 0.1$, $T = 1$, $S_0 = K = 100$ and $m = 10^3$, $l = 10^3$, $\alpha = 0$, $\beta = 2 \frac{K}{S_0}$:

σ	Price PDE	Price Riemann scheme	Price Trapez	Price BS
0.05	4.71	4.63	4.72	4.72
0.2	7.04	6.99	6.98	7.04
0.3	9.05	9.01	8.96	9.06

σ	Price PDE, $l, m = 50$	Price PDE, $l, m = 100$	Price PDE, $l, m = 200$	Price PDE, $l, m = 300$
0.05	4.79	4.24	4.29	4.48
0.2	6.93	6.96	7.02	7.03
0.3	8.97	9.03	9.05	9.05

On a seulement besoin 100 time step pour avoir des résultats 'corrects' quant $\sigma = 0.3$, 200 quant $\sigma = 0.2$ et encore plus pour $\sigma = 0.05$.

Conclusion