

Pricing d'options par FFT

LE FAY Yvann, GOUMENT Daniel, DE LAHARPE Gabriel

Janvier 2022



Résumé

Nous présentons la méthode de pricing d'options européennes de Carr-Madan. Dans le cas où l'on connaît la fonction caractéristique φ du log-prix, $\log S_t$, cette méthode permet d'approcher le prix de l'option à l'aide de transformées de Fourier. Elle permet de pricer l'option pour N strikes répartis autour du prix du sous-jacent, en un temps $O(N \log N)$. Nous nous référons à la thèse de master de ManWo Ng, PhD, *Option Pricing via the FFT*.

1 Modèles de pricing

1.1 Modèle de Black-Scholes

On suppose que le sous-jacent S_t satisfait l'équation différentielle stochastique

$$dS_t = rS_t dt + \sigma S_t dW_t.$$

La forme close est

$$S_t = S_0 \exp \left\{ (r - \sigma^2/2)t + \sigma W_t \right\}.$$

La fonction caractéristique associée est

$$\varphi(u) = \exp \left\{ iu(s_0 + (\mu - \sigma^2/2)t) - 1/2 u^2 \sigma^2 t \right\}.$$

1.2 Modèle de Merton

Le prix du sous-jacent suit un mouvement brownien géométrique auquel on rajoute un processus de Poisson Z_t avec une distribution log-normale des tailles de jumps $Y_i \sim \mathcal{N}(\mu, \delta^2)$ indépendants de

W_t et de N_t le processus de Poisson d'intensité λ indépendant de W_t que suivent les jumps. Merton propose

$$\frac{dS_t}{S_t} = rdt + \sigma dW_t + dZ_t.$$

Ce qui se résout en

$$S_t = S_0 \exp \left\{ \mu^M t + \sigma W_t + \sum_{i=1}^{N_t} Y_i \right\}$$

où $\mu^M = r - \sigma^2 - \lambda \{ \exp(\mu + \delta^2/2) - 1 \}$.

La fonction caractéristique est

$$\varphi(u) = \exp \{ t(-\sigma^2 u^2/2 + i\mu^M u + \lambda(e^{-\delta^2 u^2/2 + i\mu u - 1})) + i s_0 \}.$$

1.3 Modèle de Heston

Nous précisons pas ici le modèle, voir Hull et White (1987) pour plus de détails. La fonction caractéristique est

$$\varphi(u)e^{-ius_0} = \frac{\exp\{\kappa\theta t(\kappa - i\rho v u)/v^2 + iutr\}}{(\cosh \frac{\gamma t}{2} + \frac{\kappa - i\rho v u}{\gamma} \sinh \frac{\gamma t}{2})^{\frac{2\kappa\theta}{v^2}}} \exp\left\{-\frac{(u^2 + iu)v_0}{\gamma \coth \frac{\gamma t}{2} + \kappa - i\rho v u}\right\}$$

où $\gamma(u) = \sqrt{v^2(u^2 + iu) + (\kappa - i\rho v u)^2}$

2 Pricing avec FFT

2.1 Formule d'inversion de Carr-Madan

Notons S_T le prix à la maturité du sous-jacent pour un call européen de strike K . Définissons $s_T = \log S_T$ et la densité risque-neutre q_T . La fonction caractéristique de q_T est

$$\varphi_T(u) = \int_{\mathbb{R}} e^{ius} q_T(s) ds$$

Posons $k = \log K$, la principe de valorisation risque-neutre donne le prix du call

$$\begin{aligned} C_T(K) &= e^{-rT} \mathbb{E}(S_T - K)^+ \\ &= e^{-rT} \int_k^\infty (e^s - e^k) q_T(s) ds \end{aligned}$$

avec $\lim_{K \rightarrow 0} C_T(K) = S_0$. Ainsi $k \mapsto C_T(e^k)$ n'est pas intégrable et l'intégrale de Fourier n'existe pas. Considérons une version modifiée du prix, soit $\alpha > 0$ que l'on viendra préciser plus tard,

$$c_T(k) = e^{\alpha k} C_T(e^k)$$

avec α tel que c_T admet une transformée de Fourier. Notons ψ_T la transformée de Fourier de c_T ,

$$\psi_T(u) = \int_{\mathbb{R}} e^{iuk} c_T(k) dk$$

La transformée de Fourier inverse donne

$$C_T(K) = \frac{e^{-\alpha k}}{2\pi} \int_{\mathbb{R}} e^{-iuk} \psi_T(u) du = \frac{e^{-\alpha k}}{\pi} \operatorname{Re} \left\{ \int_0^\infty e^{-iuk} \psi_T(u) du \right\}$$

A l'aide de Fubini, on obtient

$$\psi_T(u) = \frac{e^{-rT} \varphi_T(u - (\alpha + 1)i)}{\alpha^2 + \alpha - u^2 + i(2\alpha + 1)u}$$

Une condition suffisante pour que $c_T \in L^1$, est $\mathbb{E}S_T^{\alpha+1} < \infty$.

2.2 Approximation numérique et FFT

Il s'agit d'approcher C_T par

$$C_T(K) = \frac{\exp(-\alpha k)}{\pi} \operatorname{Re} \left\{ \sum_{j=0}^{N-1} e^{-i\eta j k} \psi(\eta j) \eta \right\}$$

La somme dans la partie réelle se calcule à l'aide d'un algorithme FFT. L'algorithme de Turkey-Cooley permet de calculer les sommes de la forme $\sum_{j=0}^{N-1} e^{-i\frac{2\pi}{N}ju} x_j$ pour $u = 0, \dots, N-1$ en un temps $O(N \log N)$.

On considère des log-strikes répartis uniformément près du log-prix s_0 ,

$$k_u = s_0 + \zeta u - \frac{1}{2}N\zeta$$

pour $u = 0, \dots, N-1$.

On obtient

$$C_T(k_u) \approx \frac{\exp(-\alpha k_u)}{\pi} \operatorname{Re} \left\{ \sum_{j=0}^{N-1} e^{-i\zeta \eta j u} e^{i\{(\frac{1}{2}N\zeta - s_0)\eta j\}} \psi(\eta j) \eta \right\}$$

Il s'agit donc d'appliquer l'algorithme FFT sur les $x_j = e^{i\{(\frac{1}{2}N\zeta - s_0)\eta j\}} \psi(\eta j)$ pour $j = 0, \dots, N-1$, en fixant la condition

$$\zeta \eta = \frac{2\pi}{N}$$

Le paramètre N caractérise le temps de calcul, la variable ζ caractérise le pas pour les log-strikes. On souhaite ζ aussi petit que possible de manière à avoir le plus de points proches de s_0 , où l'approximation est la meilleure. Mais la contrainte précédente impose alors un paramètre η plus grand, ce qui vient diminuer la qualité d'approximation de l'intégrale de Fourier par la Fourier discrète.

2.3 Choix de α

Il est montré dans la thèse de ManWo Ng que la valeur $\alpha = 0.75$ est telle que $\mathbb{E}S_T^{1+\alpha} < \infty$ pour les modèles ici considérés, ce qui permet l'application de la transformation de Carr-Madar. Toutefois, le choix de α impacte la tractabilité du calcul de l'intégrale de Fourier car celle-ci vient modifier les oscillations de l'intégrande.

Il est convenable de penser qu'une valeur α , fonction des paramètres du modèle, qui minimise l'intégrande, rende le calcul plus tractable,

$$\alpha = \operatorname{argmin}_{\alpha > 0} \sup_{u \in \mathbb{R}} \left\| \frac{e^{-rT} \phi_T(u - (\alpha + 1)i)}{\alpha^2 + \alpha - u^2 + i(2\alpha + 1)u} \right\|$$

Voir la référence pour des expressions analytiques dans les cas de Black-Scholes et Heston de α .

2.4 Structure, utilisation du programme et difficultés rencontrées

Dans ce projet, nous avons fait le choix d'organiser notre code autour de l'utilisation de classes. Ceci nous a permis de profiter des avantages offerts par la programmation orientée objet.

- La classe `ClosedForm` permet le calcul des prix à partir de la formule close dans le modèle de Black Scholes pour un call européen. Elle contient les attributs suivants : risk free rate, volatilité, maturité et prix S du sous-jacent en 0.
- La classe `fft` permet de calculer la FFT d'un tableau en $O(N \log N)$.
- La classe `Cdf` (respectivement `CdfMerton` et `CdfHeston`) permet de calculer la fonction de répartition du log du prix du sous-jacent ainsi que la transformée de Fourier de c_T pour le modèle de Black Scholes (respectivement Black Scholes Merton et Heston).
- La classe `Calculator` (respectivement `CalculatorMerton`, `CalculatorHeston`) permet de calculer la liste des prix de call pour une liste de strikes autour du prix du sous-jacent en 0 en suivant le modèle de Black Scholes (respectivement Black Scholes, Merton, Heston).

L'utilisation du fichier `main.cpp` se décompose en cinq parties :

- Remplissage des paramètres propres au modèle que l'on souhaite utiliser (dans le code donné en exemple, tous les modèles sont utilisés)
- Construction d'un objet de la classe de calcul correspondant au modèle utilisée : `ClosedForm` pour un calcul de call européen à l'aide de la formule exacte et `Calculator` (respectivement `CalculatorMerton`, `CalculatorHeston` pour les modèles de Black Scholes, BSMerton ou Heston) à l'aide de la méthode FFT.
- Construction d'une liste qui va contenir les strikes (ou les log-strikes dans le cas des calculator FFT) et d'une liste qui va contenir les valeurs des call correspondants.
- Appel de la fonction de calcul de la classe correspondante (utilisation d'une boucle `for` pour la closed form).
- Extraction des résultats à l'aide de la fonction correspondante (`ClosedForm` ou FFT) dans un fichier texte. Ici, les résultats sont dans un format adapté à la lecture par Mathematica. Les fonctions d'extraction peuvent être modifiées afin que les résultats sortent dans un format lisible par Pandas par exemple.

Chaque membre du groupe a déjà une certaine expérience en programmation et a déjà fait de la POO. Nous n'avons donc rencontré aucune véritable difficulté.

L'étape majeure du projet a été de penser la structure du programme : établir les classes, puis les fonctions et les variables que chacune avaient en commun de manière à bien exploiter l'héritage. O Ce projet a été pour nous l'occasion de se familiariser avec la syntaxe C++, quelques techniques en calcul numérique (algorithme FFT), ainsi que sur différents modèles de pricing.

2.5 Comparaisons numériques

On pose, $N = 8192$, $r = 0.05$, $v = 0.7$, $\sigma = 0.2$, $T = 1$, $S_0 = 100$, $\eta = 0.0125$, $\alpha = 0.75$, $\rho = -0.5$, $v_0 = 0.2$, $\lambda = 0.13$, $\mu = 0$, $\delta = 0.0004$, $\kappa = 10$.

On compare les résultats obtenus par FFT avec, lorsque l'on peut, les résultats obtenus avec la forme close. Dans le cas où l'on ne connaît pas la forme close, on compare les résultats FFT avec ceux obtenus par la méthode `NIntegrate` de Mathematica pour l'évaluation de l'intégrale de Fourier définissant C_T . Voir Fig. 1, 2 et 3.

2.6 Comparaison avec Monte-Carlo

Il est difficile d'effectuer une comparaison toutes choses égales par ailleurs de l'algorithme présenté avec celui de Monte-Carlo. L'algorithme de MC a une complexité qui dépend de la précision attendue entre le prix Monte-Carlo et le prix exact. En effet, si on fait M tirages, le théorème centrale limite assure que la différence est de l'ordre de $1/\sqrt{M}$. Ainsi, à une précision ε fixée, la complexité de l'algorithme est $O(M) = O(1/\varepsilon^2)$. Cet algorithme donne le prix pour une seule valeur de strike.

Dans le cas de l'algorithme FFT, le paramètre qui caractérise la complexité est le nombre de strikes, N . L'écart entre l'intégrale exacte de Fourier et la transformée de Fourier discrète n'est pas explicite (à notre connaissance) mais dépend positivement de η i.e négativement de N .

Ainsi, on contrôle la précision dans un premier cas mais pas dans le second, dans le premier cas on calcule C_T pour un strike, dans le second pour N . La comparaison est délicate.

Toutefois, il est possible de trouver une configuration des deux méthodes de manière à obtenir des valeurs numériques similaires. Dans ce cas, il est possible de comparer le temps de calcul des deux algorithmes. Voir le tableau suivant.

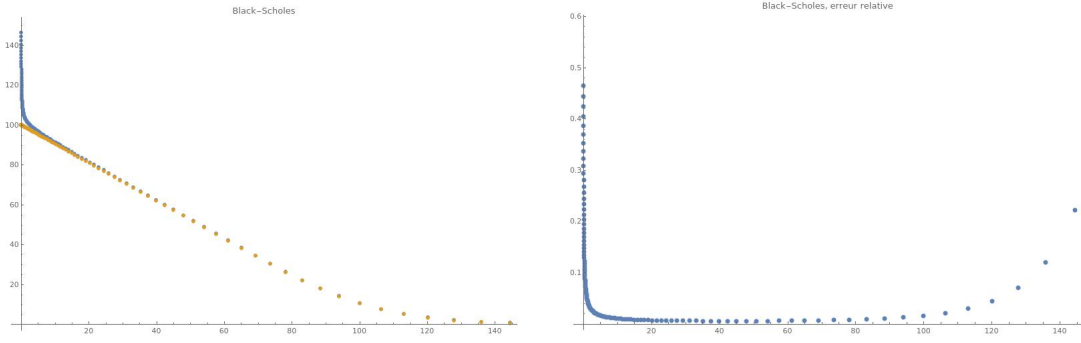


Figure 1: A gauche, en bleu FFT, en orange forme close

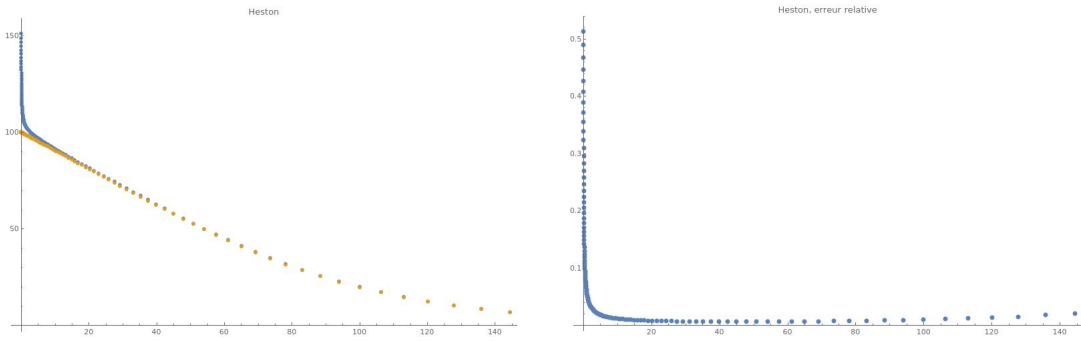


Figure 2: A gauche en orange, NIntegrate de Mathematica

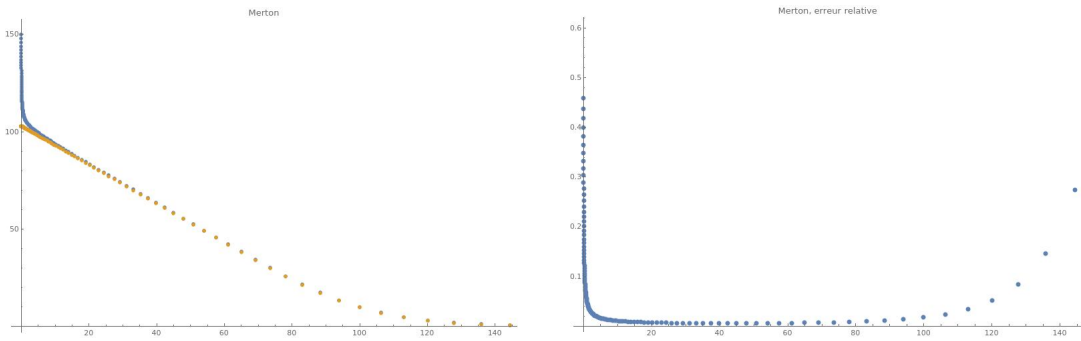


Figure 3: A gauche en orange, NIntegrate de Mathematica

| Modèle | FFT | MC |
|--------|------|------|
| Merton | 0.01 | 25.1 |
| Heston | 0.01 | 27.6 |

Figure 4: Temps de calcul en secondes pour les deux méthodes, 20 strikes différents et résultats moyennés sur 5000 essais.