FFT Option Pricing method for dummies

Matteo Gardini*

May 20, 2020

Abstract

The aim of this document is to explain, as simple as possibile, the Carr-Madan algorithm for option pricing applied to Lévy processes. We cover only the case of European Options and we provide a working Matlab code. It tooks me a lot of time to understand this algorithm so, just to be clear: the dummy is not you: it's me!

1 Introduction

The aim of this short document is to explain, in a simple and complete way, the algorithm proposed by Carr and Madan for European option pricing. The algorithm assumes that the characteristic function of risk-neutral density is known analitically. Given any characteristic function the option price could be obtained in a very efficient way using the FFT algorithm: this algorithm permits a real time pricing.
The document is organized as follow. In Section [2] we birefly introduce Lévy processes and exponential Lévy processes and we discuss briefly about risk-neutral pricing. In Section [3] we introduce Carr-Madan algorithm while in Section [4] we provide characteristic functions in Black-Scholes and in Variance Gamma framework. In Section [5] we analyze some numerical results and, in the end, in Section [6] we provide a *MATLAB* code.

2 Lévy processes and exponential Lévy processes

In this section we give the definition of Lévy process and of its characteristic exponent. Then we provide a way to model the price of a financial asset using Lévy processes.

2.1 Lévy processes

In this section we provide a concise overview of Lévy processes. A very good reference is [2]. Lévy processes can be viewd as generalization of the well know brownian motion.

*ERG Spa, Genoa and Mathematical Departement University of Genoa. For comments and suggestions write to gardini@dima.unige.it

Definition 2.1. A cadlag process $(X_t)_{t\geq 0}$ on $(\Omega, \mathcal{F}, \mathcal{P})$ with values on $\mathcal{R}^d$ such that $X_0 = 0$ is called a Lévy process if it stisfies the following properties:

- Indipendent Increments: for every sequence of time $t_0, \ldots, t_n$ the random variable $X_{t_1} - X_{t_0}, \ldots, X_{t_n} - X_{t_{n-1}}$ are indipendent.
- Stationary law: the law of$X_{t+h} - X_t$ is indipendent on h.
- Stochastic continuity: $\forall \epsilon > 0, \quad \lim_{h\to 0} \mathbb{P}\left(|X_{t+h} - X_t| \geq \epsilon\right)$

The third condition simply means that jumps don't occur at fixed (non random) times. One can see that the brownian motion satisfies all these properties.
Levy processes are usefull and have the excellent properties that, even if the probability distribution of X_t is not know, its characteristic function is know in closed form, which is essential to apply Carr-Madan algorithm.

Definition 2.2. The Characteristic Function of a $\mathbb{R}^d$ random variable X is the function $\Phi_X : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$\forall z \in \mathbb{R}^d, \quad \Phi_X(z) = \mathbb{E}\left[\exp iz \cdot X\right]$$

For a Levy process the following proposition [2] holds:

Proposition 2.1. *Let $(X_t)_{t\geq 0}$ a Lévy process on $\mathbb{R}^d$. There exist a continuos function $\psi : \mathbb{R}^d \to \mathbb{R}$, called the characteristic exponent of X, such that:*

$$\mathbb{E}\left[e^{iz\cdot X_t}\right] = e^{t\psi(z)} \quad z \in \mathbb{R}^d$$

2.2 Exponential Lévy processes and risk-neutrality

Given a Lévy process X_t one can model the price of an asset in risk-neutral world, as suggested in [2], as:

$$S_t = S_0 e^{rt+X_t} \tag{1}$$

In order to guarantee that the disconted price $e^{-rt}X_t$ is a martingale we must have that:

$$\mathbb{E}\left[e^{-rt}S_t\right] = S_0.$$

and this means that, using (1)

$$\mathbb{E}\left[e^{rt+X_t}\right] = S_0 e^{rt}.$$

which implies

$$\mathbb{E}\left[e^{X_t}\right] = 1.$$

Now, observe that $1 = e^{t\cdot 0} = \mathbb{E}\left[e^{X_t}\right] = \Phi_X(-i) = e^{t\psi(-i)}$. This means that, in order to obtain risk-neutrality, we must require that:

$$\psi(-i) = 0.$$

3 FFT Method for Option Pricing

This section is divided in three parts: in the first part we explain the option pricing algorithm based on characteristic function as proposed by Carr and Madan in [2]. Then we describe the FFT algorithm from a general point of view and we apply it to out particular problem. In the end we provide two pratical examples: the first one is in Black-School framework, that should more familiar to the reader, then we apply the algorithm to in case of Variance Gamma process, as proposed by Carr and Madan themselves [1].

3.1 Carr-Madan algorithm

Suppose we want to price an European Call option $C_T(k)$ where $k = \log K$ is the log-strike and T is the maturity. Let the risk-neutral density of the log-price s_T in T denoted by q_T (note that this density could be unknown in closed form). Let the characteristic function Φ of this density be:

$$\Phi_T(u) = \int_{-\infty}^{\infty} e^{ius} q_T(s)\, ds \tag{2}$$

Remember that $\Phi_T(u)$ is assumed to be note analitically.
The initial Call value $C_T(k)$ can be related to risk-neutral density $q_T(s)$ by:

$$\begin{aligned} C_T(k) &= \int_{-\infty}^{\infty} e^{-rT} (S_T - K)^+ \tilde{q}_T(S)\, dS \\ &= \int_{K}^{\infty} e^{-rT} (S_T - K)\, \tilde{q}_T(S)\, dS \\ &= \int_{k}^{\infty} e^{-rT} \left(e^s - e^k\right) q_T(s)\, ds \end{aligned} \tag{3}$$

We note that we can not compute the Fourier Transform of $C_T(k)$ because $\lim_{k\to-\infty} C_T(k)$ is constant and so $C_T(k)$ is not integrable. But we can chose $\alpha > 0$ and define $c_T(k) = e^{\alpha k} C_T(k)$: we note tha now $c_T(k) \to 0$ for $k \to -\infty$ and so we can compute the Fourier transform in k.

$$\Psi_T(v) = \int_{-\infty}^{\infty} e^{ivk} c_T(k)\, dk \tag{4}$$

Now observe that one can compute $C_T(k)$ by inverting (4), obtaining

$$\begin{aligned} C_T(k) &= e^{-\alpha k} c_T(k) \\ &= e^{-\alpha k} \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ivk} \Psi_T(v)\, dv \\ &= e^{-\alpha k} \frac{1}{\pi} \int_{0}^{\infty} e^{-ivk} \Psi_T(v)\, dv \end{aligned} \tag{5}$$

So, if we are able to compute the last integral in the previous equation we get the option value. We need an expression for $\Psi_T(v)$.

$$
\begin{aligned}
\Psi_T(v) &= \int_{-\infty}^{\infty} e^{ivk} c_T(k)\, dk \\
&\overset{(3)}{=} \int_{-\infty}^{\infty} e^{ivk} \int_{k}^{\infty} e^{\alpha k} e^{-rT} \left(e^s - e^k\right) q_T(s)\, ds dk \\
&\overset{s.o.}{=} \int_{-\infty}^{\infty} e^{-rT} q_T(s) \left(\int_{-\infty}^{s} \left(e^{s+\alpha k} - e^{(1+\alpha)k}\right) e^{ivk} dk \right) ds
\end{aligned}
$$

where $s.o$ stays for switching the order of integration. Now, observe that the inner integral split up in two parts: let's compute them separately.

$$
\begin{aligned}
\int_{-\infty}^{s} e^{s+\alpha k+ivk} dk &= \int_{-\infty}^{s} e^{s+(\alpha+iv)} dk \\
&= \left. \frac{e^{s+k(\alpha+iv)}}{\alpha+iv} \right]_{-\infty}^{s} = \frac{e^{(i+\alpha+iv)s}}{\alpha+iv}, \\
\int_{-\infty}^{s} e^{(s+\alpha k+ivk)k} dk &= \frac{e^{(1+\alpha+iv)s}}{1+\alpha+iv}
\end{aligned}
$$

Now we have

$$
\begin{aligned}
\Psi_T(v) &= \int_{-\infty}^{\infty} e^{-rT} q_T(s) \left[\frac{e^{(i+\alpha+iv)s}}{\alpha+iv} - \frac{e^{(1+\alpha+iv)s}}{1+\alpha+iv} \right] \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s)\, e^{s(1+\alpha+iv)} \frac{1+\alpha+iv-\alpha-iv}{(\alpha+iv)(1+\alpha+iv)} \\
&= \frac{e^{-rT}}{\alpha^2+\alpha-v^2+i(2\alpha+1)v} \int_{-\infty}^{\infty} q_T(s)\, e^{s(1+\alpha+iv)} ds
\end{aligned}
$$

Looking at the last integral of the previous equation we note that:

$$
\int_{-\infty}^{\infty} q_T(s)\, e^{s(1+\alpha+iv)} = \int_{-\infty}^{\infty} q_T(s)\, e^{is(-\alpha i+v-i)} \overset{(2)}{=} \Phi_T(v-(\alpha+1)i),
$$

which is the characteristic function of a Lévy processes and it is know in analytic form for a lot of processes. In the end we have that:

$$
\Psi_T(v) = \frac{e^{-rT}\Phi_T(v-(\alpha+1)i)}{\alpha^2+\alpha-v^2+i(2\alpha+1)v} \tag{6}
$$

If we substitute (6) in (5) we obtain

$$
\begin{aligned}
C_T(k) &= e^{-\alpha k} \frac{1}{\pi} \int_{0}^{\infty} e^{-ivk} \Psi_T(v)\, dv \\
&\overset{(5)}{=} e^{-\alpha k} \frac{1}{\pi} \int_{0}^{\infty} e^{-ivk} \frac{e^{-rT}\Phi_T(v-(\alpha+1)i)}{\alpha^2+\alpha-v^2+i(2\alpha+1)v} dv
\end{aligned}
$$

Computing this integral we get the required Call Option price.
By the way, note that if $\alpha = 0$ then the denominator vanishes if $v = 0$ this is the reason why we need the a $\alpha > 0$. The choice of α is discussed in [1]

3.2 FFT algorithm

FFT algorithm is a very efficient way to compute the sum:

$$w(k) = \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(k-1)} x(j) \quad k = 1, \dots, N, \tag{7}$$

where N is typical a power of 2.
Take Equation (5) and apply the trapezoid rule for quadrature:

$$C_T(k) = e^{-\alpha k} \frac{1}{\pi} \int_0^{\infty} e^{-ivk} \Psi_T(v)\, dv \simeq e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-iv_j k} \Psi_T(v_j)\, \eta \tag{8}$$

where $v_j = \eta(j-1)$. Observe that we truncated the upper limit of the integral, using N points with a step size of η. The upper limit of integration is then $a = N\eta$.
The FFT algorithm returns N values of k so we'll have N option values, starting from a minimum value to a maximum value, with a step size of λ, so that our values of k are:

$$k_u = -b + \lambda(u-1) \quad u = 1, \dots, N. \tag{9}$$

This gives log-strikes k running from $-b$ to b where:

$$b = \frac{1}{2} N\lambda$$

Observe that we have two dimension of discretizazione: the first one presents N points that discretize the integral in Equation (5). The second one has N points discretize the space of the log price. They can have a different step size, η for the integral and λ for the log-strike, but there are always N points.
Substituiting (9) in (8) yelds:

$$\begin{aligned} C_T(k_u) &\simeq e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-iv_j k_u} \Psi_T(v_j)\, \eta \\ &\overset{(9)}{=} e^{-\alpha k_u} \frac{1}{\pi} \sum_{j=1}^{N} e^{-iv_j[-b+\lambda(u-1)]} \Psi_T(v_j)\, \eta \\ &= e^{-\alpha k_u} \frac{1}{\pi} \sum_{j=1}^{N} e^{-iv_j \lambda(u-1)} e^{ibv_j} \Psi_T(v_j)\, \eta \quad u = 1, \dots, N. \end{aligned}$$

Remember that $v_j = \eta(j-1)$ and use substitute the v_j in the first exponent of the previous equation:

$$C_T(k_u) \simeq e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-i\eta\lambda(j-1)(u-1)} e^{ibv_j} \Psi_T(v_j)\, \eta \quad u = 1, \dots, N.$$

To apply the Fast Fourier Transform we need that $\lambda\eta = \frac{2\pi}{N}$.

$$C_T(k_u) \simeq e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)} e^{ibv_j} \Psi_T(v_j)\, \eta \quad u = 1, \ldots, N.$$

Using Simpson's rule weightings we get:

$$C_T(k_u) \simeq e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)} e^{ibv_j} \Psi_T(v_j) \frac{\eta}{3} \left[3 + (-1)^j - \delta_{j-1}\right] \quad u = 1, \ldots, N.$$

with δ_n is the Kronecker detal function. Compare this equation with Equation (7):

$$C_T(k_u) = e^{-\alpha k} \frac{1}{\pi} \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)} e^{ibv_j} \Psi_T(v_j) \frac{\eta}{3} \left[3 + (-1)^j - \delta_{j-1}\right] \quad u = 1, \ldots, N.$$

$$w(k) = \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(k-1)} x(j) \quad k = 1, \ldots, N.$$

and note that it is the exact application of the FFT with:

$$x(j) = e^{ibv_j} \Psi_T(v_j) \frac{\eta}{3} \left[3 + (-1)^j - \delta_{j-1}\right]$$

4 Characteristic functions: the Black-Scholes and Variance Gamma cases

In this section we derive the characteristic function of risk neutral density of the stock for Black-Scholes framework and for Variance Gamma framework.

4.1 Back-Scholes model

Suppose to model the stock S_t as

$$S_t = S_0 e^{rt + X_t + \omega t}$$

where X_t is a Brownian motions $X_t = \mu t + \sigma W_t$ and ω is a parameter that must be set to be in a risk-neutral framework. The aim is to derive the characteristic function of $\log S_t$ in a risk-neutral world.
Recalling what we say in Section 2.2 we have that we are in a risk neutral world if:

$$\begin{aligned} \mathbb{E}[S_t] &= S_0 e^{rt} \\ \mathbb{E}\left[S_0 e^{rt + X_t + \omega t}\right] &= S_0 e^{rt} \\ \mathbb{E}\left[e^{X_t}\right] &= e^{-\omega t} \end{aligned} \tag{10}$$

Remember that $X_t \sim \mathcal{N}\left(\mu t, \sigma^2 t\right)$ so its characteristic function is

$$\phi\left(u\right) = \mathbb{E}\left[e^{iuX_t}\right] = e^{i\mu t u - \frac{1}{2}u^2\sigma^2 t} \tag{11}$$

From this equation using (10) we get:

$$\begin{aligned}\phi\left(-i\right) &= e^{\mu t + \frac{1}{2}\sigma^2 t} = \mathbb{E}\left[e^{X_t}\right] \\ &\overset{(10)}{=} e^{-\omega t}\end{aligned}$$

and this implies that $\omega = -\mu - \frac{1}{2}\sigma^2$.
We are ready to compute the characteristic function of $\log S_t$ in risk-neutral world.

$$\begin{aligned}\Phi_t\left(u\right) &= \mathbb{E}\left[e^{iu\left(\log S_0 + rt + X_t + \omega t\right)}\right] \\ &= \mathbb{E}\left[e^{iu\left(\log S_0 + rt + X_t + \left(-\mu - \frac{1}{2}\sigma^2\right)t\right)}\right] = e^{iu\log S_0} e^{iu\left(r - \mu - \frac{\sigma^2}{2}\right)t}\mathbb{E}\left[e^{iuX_t}\right] \\ &\overset{(11)}{=} e^{iu\left(\log S_0 + r - \mu - \frac{\sigma^2}{2}\right)t} e^{iu\mu t - \frac{\sigma^2}{2}u^2 t} = e^{iu\log S_0 + iu\left(r - \frac{\sigma^2}{2}\right)t - \frac{\sigma^2}{2}u^2 t} \\ &= e^{iu\log S_0 + iu\left(r - \frac{\sigma^2}{2}\right)t - \frac{\sigma^2}{2}u^2 t}\end{aligned}$$

So we have

$$\Phi_T\left(u\right) = e^{iu\log S_0 + iu\left(r - \frac{\sigma^2}{2}\right)T - \frac{\sigma^2}{2}u^2 T} \tag{12}$$

that we can use in Equation (6) to obtain $\Psi_T\left(u\right)$ that can be used Equation (8) to compute call price using the Fast Fourier Transform.

4.2 Variance-Gamma model

Let's now considere a Variance-Gamma model $X_t\left(\theta, \sigma, \nu\right)$ and set, as in previous chapter:

$$S_t = e^{rt + X_t + \omega t}. \tag{13}$$

The aim of this section is to compute the characteristic function of a Variance-Gamma process and derive the characteristic function of $\log S_t$ in the risk-neutral world.

4.3 Characteristic Function

Consider $X_t\left(\theta, \sigma, \nu\right)$ of the form

$$X_t = \theta G\left(t\right) + \sigma\sqrt{G\left(t\right)}Z, \tag{14}$$

where

$$\begin{aligned}Z &\sim \mathcal{N}\left(0, 1\right) \\ G\left(t\right) &\sim \Gamma\left(\frac{t}{\nu}, \frac{1}{\nu}\right).\end{aligned}$$

We want compute the characteristic funcion of $X\left(t\right)$. Observe that (14) is conditionally normal (c.n.) to $G\left(t\right)$.

$$
\begin{aligned}
\mathbb{E}\left[e^{iuX_t}\right] &= \mathbb{E}\left[e^{iu\left(\theta G_t+\sigma\sqrt{G(t)}Z\right)}\right] = \mathbb{E}\left[\mathbb{E}\left[e^{iu\left(\theta g+\sigma\sqrt{g}Z\right)}|G_t=g\right]\right] \\
&\overset{c.n}{=} \mathbb{E}\left[e^{iu\theta g-\frac{\sigma^2 gu^2}{2}}\right] = \mathbb{E}\left[e^{ig\left(u\theta+\frac{\sigma^2}{2}iu^2\right)}\right] \overset{\left(z=\theta u+\frac{\sigma^2}{2}iu^2\right)}{=} \mathbb{E}\left[e^{izg}\right] \\
&= \left(1-\frac{iz}{\frac{1}{\nu}}\right)^{-\frac{t}{\nu}} = (1-iz\nu)^{-\frac{t}{\nu}} = \left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)^{-\frac{t}{\nu}}
\end{aligned}
$$

So, we have that:

$$
\phi(u) = \mathbb{E}\left[e^{iuX_t}\right] = \left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)^{-\frac{t}{\nu}} = e^{-\frac{t}{\nu}\log\left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)} \tag{15}
$$

4.4 Risk-Neutrality condition

Preceding as in Black-Scholes case in order to obtain risk neutrality we have to request that:

$$
\phi(-i) = e^{-\omega t}. \tag{16}
$$

Using Equation (15) in (16) we obtain:

$$
\omega = \frac{1}{\nu}\log\left(1-\frac{\sigma^2\nu}{2}-\theta\nu\right)
$$

We are ready to compute the characteristic function of $\log S_t$ [1] in risk-neutral world in a Variance Gamma framework.

$$
\begin{aligned}
\Phi_t(u) &= \mathbb{E}\left[e^{iu(\log S_0+rt+X_t+\omega t)}\right] \\
&= e^{iu\log S_0+iu(r+\omega)t}\mathbb{E}\left[e^{iuX_t}\right] \\
&\overset{(15)}{=} e^{iu\log S_0+iu(r+\omega)t}e^{-\frac{t}{\nu}\log\left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)} \\
&= e^{iu\log S_0+iu(r+\omega)t}\left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)^{-\frac{t}{\nu}}
\end{aligned} \tag{17}
$$

Again we have

$$
\Phi_T(u) = e^{iu\log S_0+iu(r+\omega)T}\left(1-i\left(u\theta+\frac{\sigma^2}{2}iu^2\right)\nu\right)^{-\frac{T}{\nu}} \tag{18}
$$

that we can use in Equation (6) to obtain $\Psi_T(u)$ that can be used Equation (8) to compute call price using the Fast Fourier Transform.

[1] Note that this is the characteristic function of **log-price** because $rt+X_t+\omega t$ are the log-prices!

Parameter	Value
S_0	100
K	99
T	1
r	0.01
σ	0.2

Table 1: Black-Scholes parameters set.

Method	Call Value
Explicit Formula	8.9185
FFT Method	8.9161

Table 2: Black-Scholes Call value.

5 Numerical Results

The aim of this article is not to provide the best way to price every type of option but to show how the method works. In this section we use the FFT method to price European Call Option in non pathological cases. In the first part of this section we price European Call Options in a Black-Scholes framework using closed-form option pricing formula and FFT method. In the second section we use FFT method to price European Call Options in a Variance Gamma framework and compare this results with the ones obtained using a Monte Carlo method.

5.1 Option Pricing in Black-Scholes framework

In order to check the method we start with the well known Black-Scholes methods. In Black-Scholes framework a close formula for European Options pricing is avaiable: this give us a useful tool to check the model. For this experiment we use parameters shown in Table 1.
With this set of parameters the Call value are the ones showed in Table 2. In Figure 1 are shown Call Prices obtained via FFT Method varying strike K and maturity T, whereas in Figure 2 we display the error obtained using FFT methods. As observed in [1] we notice that the biggest error is for small maturities and at the money derivatives: this is due to the hight oscillatory integrands of the Fouries method. We observe that the surface shows small errors.

Anyway, we observe that $RMSE = 0.0024$ and so we can state that the approximation error, for this experiment, is acceptable.

5.2 Option Pricing in Variance-Gamma framework

Now we move to the more complicated Variance Gamma Framework. Here, we do not use a closed form approach but we compare option pricing via FFT and via Monte Carlo method. For this experiment we choose the set of parameters

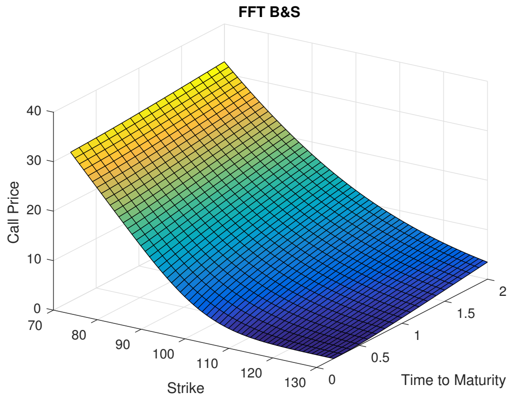

Figure 1: Black-Scholes surface prices varying K and T

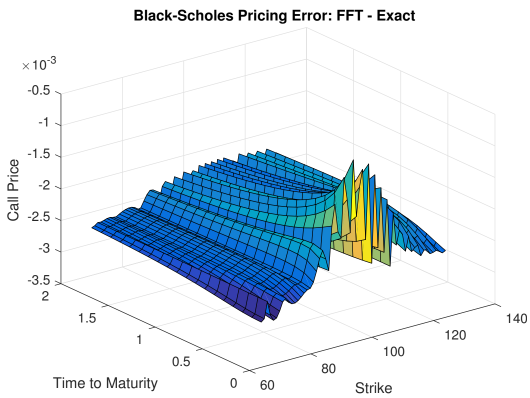

Figure 2: Error of the FFT method for Black-Scholes framework varying K and T

Parameter	Value
S_0	100
K	99
T	1
r	0.01
σ	0.12136
θ	-0.1435
ν	0.05

Table 3: Variance Gamma parameters set.

Method	Call Value
Monte Carlo Method	6.0408
FFT Method	5.9768

Table 4: Variance Gamma Call value.

displayed in Table 3, whereas in Table 4 Call values obtained with the two methods.

Again, we obeserve that the prices are very closed. In Figure 3 we show the Call Value surface obtaned with the FFT method and in Figure 4 we display the price differences obtained with MC method and with FFT method. We have that $RMSE = 0.0702$ so we can state that the FFT methods and Monte Carlo method produce the "same" results. We didn't investigate the reason of error trend in Figure 4. Moreover in Table 5 computational time is shown: observe that FFT methos is sharply faster that Monte Carlo methods and so should be preferred for plain vanilla derivatives.

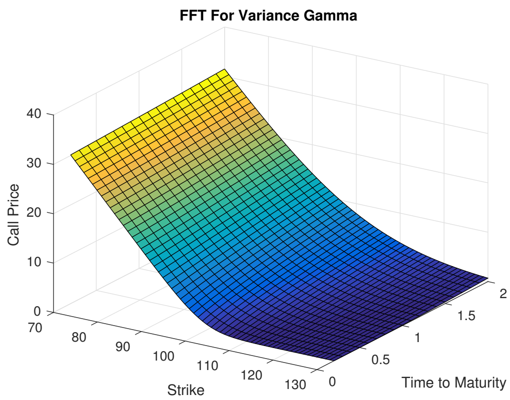

Figure 3: Variance Gamma surface prices varying K and T

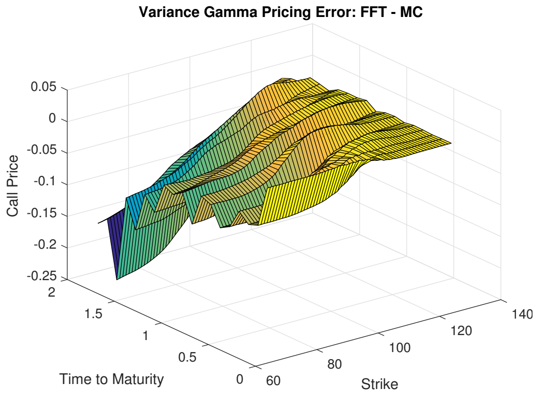

Figure 4: Difference between prices obtained with Monte Carlo method and with FFT method varying K and T

Method	**Elapsed Time (s)**
Monte Carlo Method	126.43388
FFT Method	0.01253

Table 5: Computational time (in seconds) of FFT method and Monte Carlo method.

6 *MATLAB* code

In this section we provide the code we used for the experiment. It is our opinion that, sometimes, providing the code is a good idea for at least three reasons:

- The reader can easly check the results presented in the paper.
- If the reader is not so prone to programming (maybe because it is at the beginning of her carrier) having the code can be usefull to learn.
- The author is not a perfect man: he commits a lot of mistakes! So a kind reader can help him to fix them!

6.1 main

This script show you how to use the other functions that have been developed to use FFT algorithm.

```
%% Set the parameters
r = 0.01; % risk-free
sigma = 0.12136; % volatility
theta = -0.1436; % drift (no-risk-neutral)
nu = 0.05; % subordinato paramets
S0 = 100; % Underlying starting value
K = 100; % Strike price
T = 1; % maturity

r = 0.01;
sigma = 0.2;
theta = -0.1436;
nu = 0.05;
S0 = 100;
K = 95;

Nsim = 5e5; % Number of simulations
Nsim = 1e2;
Ndates = 10000; % Discretization step
dt = T/Ndates; % time interval

% Call FFT method
[CallPrices,LogStrikes] = FFTPricing(T,r,...
    @(w)phi_bls(w,S0,r,T,sigma));

% FFT pricing
Call_GBM_FFT = interp1(LogStrikes,CallPrices,log(K));

% Explicit formula by Black-Scholes
Call_GBM_EXACT = blsprice(S0,K,r,T,sigma);

%% Variance Gamma Framework
params(1) = theta;
params(2) = nu;
params(3) = sigma;

% Set omega
omega = (1/nu).*...
    log(1-theta*nu - sigma*sigma*nu/2);

% Call FFT Method
```

```
[CallPrices,LogStrikes] = FFTPricing(T,r,...
    @(w)phi_vg(w,S0,r,omega,T,theta,nu,sigma));


%% Perform Monte Carlo Simulation
X = VG_simulation(Nsim,Ndates,dt,params);
t = linspace(0,T,Ndates);

% Build underlying dynamics
S = S0.*exp(r.*repmat(t,Nsim,1) + ...
    X + omega.*repmat(t,Nsim,1));

% FFT prices
Call_VarianceGamma = interp1(LogStrikes,CallPrices,log(K));

% Monte Carlo Prices
Call_VarianceGamma_MC = exp(-r*T).*mean(max(S(:,end)-K,0));
```

6.2 Pricing using FFT

This function contains the implementation of the algorithm. It has been developed for European Call Options pricing and can be easly rewritten to price European Put options. This is a very basic implementation: feel free to develop an optimized and more versatile version of it!

```
function [CallPrices,LogStrikes] = FFTPricing(T,r,phi)
% [CallPrices,Strikes] = FFTPricing(S0,K,T,phi)
% Pricing Call values using Method prodivided by Carr-Madan 1999
% T: maturity
% r: risk free
% phi: Characterist function of Log-prices
% For black-Scholes Framework that is
% phi=@(x) exp(1i*(log(S0) + (r-0.5*sigma^2)*T).*x ...
    -0.5*sigma^2*T.*x.^2);
%
% MG: 02marzo2017

% Integer (needed to define quadrature nodes)
L = 14;

% N, power of 2
N = 2^L;

% integration step
dv = 0.25;
% dk integration step on log(K) grid
dk = 2*pi/(N*dv);

% Alpha
alpha = 0.75;

% Vector v (integrands)
jvec=1:N;
v = (jvec-1)*dv;

% vector k (log strike)
uvec = 1:N;

% lower bound for ku
```

```
b=(N * dk) / 2;
ku=-b + dk*(uvec-1);

% To transform....
psi= exp(-r*T) .*phi(v-(alpha+1)*1i)./(alpha^2+alpha - v.^2 ...
    +1i*(2*alpha+1).*v);

% Simpson Quadrature Rule
SimpsonW=zeros(1,N) ;
SimpsonW(1)=1/3;
for i =2:N

    if(mod( i ,2)==0)
        SimpsonW( i )=4/3;
    else
        SimpsonW( i )=2/3;
    end
end
tmp=dv.*psi .* exp(1i * v *b ).*SimpsonW;


% Call Pricing
CallPrices=exp(-alpha.*ku)./pi .* real(fft(tmp));
LogStrikes=ku;


end
```

6.3 Characteristic Functions

Here we provide the code that implements characteristic function in Black-Scholes and Variance Gamma framework. Note that, if you want to test another model, you can write your own characteristic function and use it in FFT code.

```
function v = phi_bls(u,S0,r,T,sigma)
% function v = phi_bls(u,S0,r,T,sigma)
% Black-Scholes Characteristic function
% S0: starting value
% r: risk-free
% sigma: volatility
% T: time to maturity

v = exp((log(S0) + (r -0.5.*sigma.^2).* T).*1i.*u ...
    -0.5.*sigma^2.*T.*u.^2);



end
```

```
function v = phi_vg(u,S0,r,omega,T,theta,nu,sigma)
% function v = phi_vg(u,S0,r,omega,T,theta,nu,sigma)
% Variance Gamma Characteristic Funcion
% S0: starting value
% r: risk free
% omega: correcting parameter
% T: time to maturity
% nu: subordinator parameter
```

```
% sigma: volatility

    argomento = 1 - 1i*theta * nu * u + 0.5 * sigma * sigma .* u ...
        .* u *nu;
    v = exp((log(S0) + (r + omega) * T).*1i.*u) .* ...
        argomento.^(-T/nu);



end
```

6.4 Monte Carlo for Variance Gamma

This is the code that implements Monte Carlo simulations in Variance Gamma frameworlk. Probably it is not the most performing code but we preferred semplicity to efficiency.

```
function X = VG_simulation(Nsim,T,dt,params)
% function X = VG_simulation(Nsim,T,dt,params)
% Monte Carlo Simulation of a Variance Gamma process

theta = params(1);
nu = params(2);
sigma = params(3);
g = gamrnd(dt/nu,nu,Nsim,T);

X = zeros(Nsim,T);

for j = 2:T
    X(:,j) = X(:,j-1) + theta.*g(:,j) + ...
        sigma.*sqrt(g(:,j)).*randn(Nsim,1);

end


end
```

7 Conclusions

In this paper we presented the FFT method and we tested in Variance Gamma and Black-Scholes framework. FFT method is very suitable to price vanilla options: it is harder to apply it to more complex derivatives. For this reason FFT method is used mainly in model calibration [2] where a fast pricing method is needed because it is usually used in a optimization method that is, almost surely, iterative.
We tested FFT method in both framework and we verified that pricing results are comparable to the ones obtained with other methods (such as closed formula or Monte Carlo). We observed that FFT method is sharply faster that Monte Carlo method. It can be shown that it is faster than PDE method: this sounds obvious because the strength of FFT method is that it permits the pricing of a large set of option with diffent strikes at the same times.

References

[1] P. Carr and D. Madan. Option valuation using the fast fourier transform. *Journal of Computational Finance*, (2):61–73, 1999.

[2] P. Tankov and R. Cont. *Financial Modeling with Jump Processes*. Chapman and Hall, 2004.