Christian - Albrechts - University Kiel
Faculty of Business, Economics and Social Sciences
Institute for Quantitative Business and Economic Research
Professorship for Empirical Finance

Option Pricing Using The Fast Fourier Transformation

Seminar Paper - Empirical Finance
Summer Term 2021

Submitted by:
Gian Luca Estivariz Lopez
Matriculation Number:
Semester:
Field of Study:
Kiel, 4th June, 2021

Contents

List of Figures

List of Abbreviations

BSM	Black-Scholes Model
CF	Characteristic Function
CRR	Cox-Ross-Rubinstein
DFT	Discrete Fourier Transformation
EMM	Equivalent Martingale Measure
FFT	Fast Fourier Transformation
FrFFT	Fractional Fast Fourier Transformation
GBM	Geometric Brownian Motion
MC	Monte-Carlo
MSE	Mean Squared Error

List of Symbols

a	Upper Integration Limit
B	Bond
b	Strike Price Interval
C	Call Option Price
c	Dampened Call Option Price
E	Even Part of Fourier Transform
e	Euler's Number
$\mathbb{E}$	Expectation
$\mathbf{F}$	Filtration
$\mathcal{F}$	σ-Field
$\mathbb{F}$	Fourier Transformation
$\mathbb{F}^{-1}$	Inverse Fourier Transformation
i	Complex Number
$\Im$	Imaginary Part
K	Strike Price
k	Log Transformed Strike Price
L^p	Lebesgue Space
M	Number of Simulations
N, n	Number of Observations
O	Odd Part of Fourier Transform
$\mathcal{O}(x)$	Complexity
$\mathcal{P}$	Probability
Q	Equivalend Martingale Measure
q	Risk Neutral Density
r	Risk Free Rate
$\Re$	Real Part
S	Stock Price
s	Log Transformed Stock Price
T	Maturity Time
t	Time
V	Portfolio Value
W	Standard Brownian Motion
W_{DFT}	Disrete Fourier Transform Matrix
α	Damping Factor
δ	Kronecker Delta

η	Grid Spacing
λ	Strike Price Spacing
μ	Local Drift Rate
ν	Grid Points
π	Pi
σ	Local Diffusion Coefficient
ϕ	Characteristic Function
φ	Self-Financing Strategy
ψ	Fourier Transformed Option Price
ω_n	n-th Unit Root
Φ	Cummulative Distribution Function
Ω	Set of All Possible Outcomes

1 Introduction

Option pricing plays a key role in financial markets. Only a few versatile valuation techniques exist that provide both computationally efficient and accurate results. This paper will introduce one of them, Fourier-based option pricing. The method was introduced by Stein and Stein (1991, p.743-746) and Heston (1993, p.331). The former derived an explicit pricing formula by using the distribution function, whereby the latter was an approach based on the characteristic function (CF) of the underlying process. A large body of literature shows the usefulness of Fourier-methods in different settings (cf. Bakshi et al. (1997)). We will focus on the approach of Carr and Madan (1999), since it was the first to make use of the fast Fourier transformation (FFT) and is often taken as an introduction to the topic. There are, however, extensions of this method. A Fourier-cosine expansion can been used to price bermudan and american options (cf. Zhang and Oosterlee (2014)). Other than Carr and Madan, here, the density function in the risk-neutral pricing formula is directly replaced by its Fourier-series representation. Lord et al. (2008) proposed a convolution method that dynamically prices european options, which makes pricing of american-style options also possible.

The presented work involves, amongst other things, a simulation study that will highlight the differences in the considered option valutation techniques. Besides the Fourier-based approach, the analysis includes pricing via Monte-Carlo (MC) simulation as well as via the method of Cox et al. (1979) (CRR). As a benchmark for the accuracy, we choose the analytical solution given by Black and Scholes (1973).

This paper proceeds as follows. Section 2 revisits the basics of option pricing and explains the Discrete Fourier Transformation (DFT) and a faster computation by the FFT. Section 3 derives the Fourier-based approach for option pricing in the Black-Scholes model. Section 4 implements this method and compares its performance with other valuation methods. Section 5 discusses the simulation results by pointing out advantages and disadvantages of the method. Section 6 concludes.

2 Background

2.1 Option Pricing

In the following we recall some basic ideas of option pricing in the Black-Scholes Model (BSM) that are necessary for later discussions.

2.1.1 Valuation Problem

We work in a filtered probability space $(\Omega, \mathcal{F}, \mathbf{F}, \mathcal{P})$, where Ω denotes the set of all possible outcomes, $\mathbf{F}$ the filtration on Ω that allows us to only use the information up to a time point $t \geq 0$, i.e. $\mathbf{F}(\mathcal{F}_t)$, where the σ-field $\mathcal{F}_t$ contains all known events up to time t, and $\mathcal{P}$ the probability assigned to each event. Furthermore, the market has a fixed interest rate $r \geq 0$ and consists of two assets, a riskless bond $B(t) = e^{rt}$ and a risky, non-dividend paying stock $S(t)$ whose random behaviour can be modelled as a geometric Brownian motion (GBM), i.e. it satisfies the following stochastic differential equation.

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t) \tag{1}$$

, with local drift rate μ, local diffusion coefficient σ and standard Brownian motion W (cf. Shreve (2004, p.264)). We want to numerically evaluate the price of an option at time t, for which at maturity T the value C_T is paid. As an introduction to the topic, we choose a vanilla european call option with strike price K and underlying stock S.

$$C_T(S(T), K) = \max\{S(T) - K, 0\} \tag{2}$$

Now, assume that the market is complete, i.e. there exists a self-financing portfolio V_φ consisting of bond and stock for which its terminal value $V_\varphi(T) = C_T$. Under a unique Equivalent Martingale Measure (EMM) $Q \sim \mathcal{P}$ the market is free of arbitrage and the fair value of the option at time t is the conditional expectation of the discounted payoff with respect to Q.

$$\begin{aligned} V_\varphi(t) &= B(t)\mathbb{E}^Q(C_T B(T)^{-1}|\mathcal{F}_t) = e^{-r(T-t)}\mathbb{E}^Q(C_T|\mathcal{F}_t) \\ &= e^{-r(T-t)} \int_0^\infty \max\{S(T) - K, 0\} q_t(S(T)|\mathcal{F}_t)\, dS(T) \\ &= e^{-r(T-t)} \int_K^\infty (S(T) - K) q_t(S(T)|\mathcal{F}_t)\, dS(T) \end{aligned} \tag{3}$$

, which depends on the risk neutral density $q_t(S)$, which is *not* always available in closed-form (cf. Cox and Ross (1976, p.151-154)).

2.1.2 Black Scholes Model

Consider the mentioned call option (2). Under the assumptions of the BSM, the market is also complete and arbitrage-free. It further assumes that the stock price also follows a GBM under the EMM Q. Moreover, in this specific model, the risk neutral density $q_t(S)$ is *known* in closed-form, hence we get the following analytical solution to the aforementioned valuation problem (cf. Black and Scholes (1973, p.644)).

$$V(t) = S(t)\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2) \tag{4}$$

, where Φ is the cummulative distribution function of the standard Normal distribution with

$$d_1 = \frac{log(\frac{S(t)}{K}) + r(T-t) + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = \frac{log(\frac{S(t)}{K}) + r(T-t) - \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}} \tag{5}$$

2.2 Fast Fourier Transformation

This subsection explains the idea behind the Fourier transformation and indicates its usefulness for option pricing.

2.2.1 Fourier Transforms

We can use a Fourier series to approximate periodic functions $f(x)$ as an infinite sum of sinusoids and Euler's identity $e^{ix} = \cos(x) + i\sin(x)$ to rewrite these sinusoids as complex exponentials (cf. Champeney (1987, p.156)).

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(\underbrace{a_k\cos(kx)}_{\text{real part } \Re} + \underbrace{b_k\sin(kx)}_{\text{imaginary part } \Im}) = \sum_{k=1}^{\infty} c_k e^{ikx} \tag{6}$$

, with Fourier coefficients a_0, a_k, b_k. We can use the Fourier transform, which is the limit of the Fourier series for periods $\to \pm\infty$, to extend such an approximation to non-periodic functions (cf. Champeney (1987, p.45)).

$$\mathbb{F}(f(x)) = \hat{f}(u) = \int_{-\infty}^{\infty} e^{iux} f(x)\,dx = \int_{-\infty}^{\infty} \cos(kx)f(x)\,dx + i\int_{-\infty}^{\infty} \sin(kx)f(x)\,dx \tag{7}$$

, where $i = \sqrt{-1}$ is the imaginary unit.

Next we need the CF. The motivation being that it is often much easier to find the CF of a random variable in closed form than it is to find its density function. The CF for a random variable X, defined for $t \in \mathbb{R}$ with distribution function $f(x)$, is given as follows.

$$\phi_T(u) = \mathbb{E}(e^{iux}) = \int_{-\infty}^{\infty} e^{iux} f(x)\,dx \tag{8}$$

As one might have already noticed, $\phi_T(u)$ is, by definition, the Fourier transform of the distribution function $f(x)$. Therefore, it makes sense to use $\phi_T(u)$ to find $f(x)$ by Fourier inversion. In general, the Fourier transform $\hat{f}(u)$ of the function $f(x)$ can be reversed as follows (cf. Abate and Whitt (1992, p.14)).

$$\mathbb{F}^{-1}(\hat{f}(u)) = f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-iux}\hat{f}(u)\,du \tag{9}$$

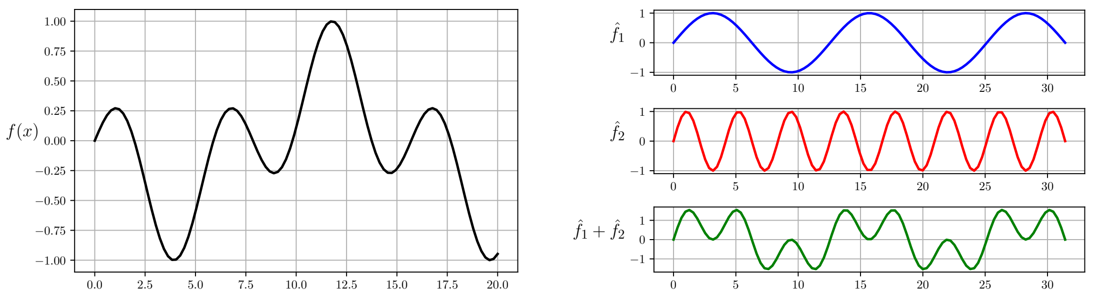
A necessary condition for the Fourier transform and its inverse to exist is that they are integrable (cf. Champeney (1987, p.49)). We call a function $f(x)$ integrable, i.e. $f \in L^1(\mathbb{R})$, if and only if

$$\int_{-\infty}^{\infty} f(x)\,dx < \infty \tag{10}$$

, where $\lim_{x\to\infty} f(x) = 0$ must be satisfied for this to hold.

2.2.2 Discrete Fourier Transformation

The discrete Fourier transformation (DFT) is a method to approximate a Fourier series on a finite interval.



(a) Sampled (unknown) function

(b) Two sinusoid frequencies and their combination

Figure 1: Fourier series approximation

More explicitly, we can approximate a function f using n observed data points $f = (f_1, ..., f_n)$ by multiplying f with the DFT-matrix W_{DFT} and the resulting Fourier transform $\mathbb{F}(f(x)) = \hat{f}(u)$ will tell us how we have to combine different frequencies of cosine and sine to reconstruct f, also illustrated in Figure (1).

$$\underbrace{\begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \hat{f}_3 \\ \vdots \\ \hat{f}_n \end{pmatrix}}_{\hat{\boldsymbol{f}} \in \mathbb{C}^n} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_n & \omega_n^2 & \dots & \omega_n^{n-1} \\ 1 & \omega_n^2 & \omega_n^4 & \dots & \omega_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_n^{n-1} & \omega_n^{2(n-1)} & \dots & \omega_n^{(n-1)^2} \end{pmatrix}}_{\boldsymbol{W_{DFT}} \in \mathbb{C}^{n \times n}} \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{pmatrix}}_{\boldsymbol{f} \in \mathbb{R}^n} \Leftrightarrow \hat{f}_j = \sum_{k=1}^{n} f_k e^{-jk2\pi i/n} \tag{11}$$

, for $j = 1, ..., n$. Here, $\omega_n = e^{-2\pi i/n}$ is the n-th unit-root and is used to compute the Fourier transform for the interval given by the n data points. Each frequency $\hat{f}_j$ includes information about the amplitude as well as the phase of the sinusoids. To obtain the desired function f we apply the inverse DFT to the frequencies as follows.

$$f_k = \frac{1}{n} \sum_{j=1}^{n} \hat{f}_j e^{jk2\pi i/n} \tag{12}$$

We see that the DFT involves n multiplications and $n - 1$ additions for each row, hence for a sequence of length n we must perform $n(2n - 1)$ computations, which is $\mathcal{O}(n^2)$ (cf. Cochran et al. (1967, p.1665)).

2.2.3 Fast Fourier Transformation

There are many different FFT algorithms and a full coverage is beyond the scope of this paper. For a thourough comparison of available FFT implementations see Soni and Kunthe (2011). A characteristic they all share is that none of them runs faster than $\mathcal{O}(nlog(n))$ and most of them have the same structure, essentially breaking the main computation into smaller pieces. In principle, there exist two main types of FFTs, namely the decimation-in-time and the decimation-in-frequency. The difference being the order of the split-and-recombination tasks. A mixture of these two, the so-called decimation-in-time-frequency, is also possible, where each version of the FFT is applied in different stages of the computation (cf. Saidi (1994, p.454)).

The work of Cooley and Tukey (1965, p.297-300) reduces the computational complexity of the DFT by using a decimation-in-time version of the FFT. This works best when assuming $n = 2^Z$, $Z \in \mathbb{N}$, which is always possible by zero-padding. We exploit the following symmetry in the DFT.

$$\hat{f}_{j+n} = \sum_{k=1}^{n} f_k e^{-2\pi i(j+n)k/n} = \sum_{k=1}^{n} f_k e^{-2\pi ijk/n} e^{-2\pi ik} = \hat{f}_j \tag{13}$$

Now, seperate even and odd indexed values and rewrite the sum of n operations into two sums of $n/2$ operations each. For this, denote even values as f_{2k} and odd values as f_{2k+1}, so that we end up with

$$\hat{f}_j = \underbrace{\sum_{k=1}^{n/2} f_{2k} e^{-2\pi i j 2k/n}}_{\text{Even Part}} + \underbrace{\sum_{k=1}^{n/2} f_{2k+1} e^{-2\pi i j (2k+1)/n}}_{\text{Odd Part}} = E_j + e^{-2\pi i j/n} O_j \tag{14}$$

Apply this divide-and-conquer approach recursively to each term until we have reduced the original sequence of length n into n subsequences of length one, which takes Z splits. We recover the DFT's original sequence by combining the following pairs

$$\hat{f}_j = E_j + e^{-2\pi i j/n} O_j \tag{15}$$

$$\hat{f}_{j+\frac{n}{2}} = E_{j+\frac{n}{2}} + e^{-2\pi i j/n} e^{-\pi i} O_{j+\frac{n}{2}} = E_j - e^{-2\pi i j/n} O_j \tag{16}$$

We end up with $\frac{n}{2}$ pairs and each of them needs one multiplication and two additions, giving us $\frac{n}{2}$ multiplications and n additions in total. On top of that, we had to perform $Z = log_2(n)$ splits. In summary, the FFT has complexity of $\mathcal{O}(n log_2(n))$.

3 Fourier-Based Option Pricing

As we have seen, the CF is identical to the Fourier transformation, meaning that we can use the inverse of the CF to perform our option valuation.

3.1 Carr and Madan Inversion

The method of Carr and Madan (1999, p.63,64) includes four main steps. First, we rewrite the option price C_T in its integral form. Second, we apply a Fourier transform on C_T. Third, we represent this Fourier transform in terms of the CF of the underlying asset S. Last, we invert the Fourier-transformed option price that now depends on the CF.

Let us consider equation (3) at time $t = 0$, with $e^k = K$, $e^s = S$ and $q_T(s)$ as the risk-neutral density of s. The option price in integral form is given as follows.

$$C_T(k) = V_\varphi(T) = e^{-rt} \int_k^\infty (e^s - e^k) q_T(s)\, ds \tag{17}$$

Since $\lim_{k\to-\infty} C_T(k) = S(0) \neq 0$, the necessary condition for (10) is violated and the Fourier transform does not exist. We can circumvent this problem by introducing a damping factor $e^{\alpha k}$, with $\alpha > 0$ (an exact value will be given later).

$$c_T(k) = e^{\alpha k} C_T(k) \Rightarrow \lim_{k \to -\infty} e^{\alpha k} C_T(k) = 0 \Rightarrow c_T(k) \in L^1 \tag{18}$$

The Fourier transform of $c_T(k)$ is given by

$$\begin{aligned}\mathbb{F}(c_T(k)) = \psi_T(\nu) &= \int_{-\infty}^{\infty} e^{i\nu k} c_T(k)\, dk \\ &= \int_{-\infty}^{\infty} e^{i\nu k} \left(\int_{k}^{\infty} e^{-rT} e^{\alpha k} (e^s - e^k) q_T(s)\, ds \right) dk \\ &= \int_{-\infty}^{\infty} e^{-rT} q_T(s) \underbrace{\left(\int_{-\infty}^{s} e^{i\nu k} (e^{\alpha k + s} - e^{\alpha k + k})\, dk \right)}_{*} ds \end{aligned} \tag{19}$$

We used Fubini's Theorem to change the order of integration since both integrals are continuous and have fixed integration bounds (cf. Champeney (1987, p.18)). Next, we solve the inner integral,

$$* = \frac{e^{\alpha s + s + i\nu s}}{\alpha^2 + \alpha + i^2 \nu^2 + i(2\alpha + 1)\nu} = \frac{e^{is(\nu - (\alpha+1)i)}}{\alpha^2 + \alpha - \nu^2 + i(2\alpha + 1)\nu} \qquad , i^2 = -1$$

Plugging it back into equation (19) yields

$$\begin{aligned}\psi_T(\nu) &= \frac{e^{-rT}}{\alpha^2 + \alpha - \nu^2 + i(2\alpha+1)\nu} \underbrace{\int_{-\infty}^{\infty} e^{is(\nu - (\alpha+1)i)} q_T(s)\, ds}_{= \mathbb{F}(q_T(s)) = \phi_T} \\ &= \frac{e^{-rT} \phi_T(\nu - (\alpha+1)i))}{\alpha^2 + \alpha - \nu^2 + i(2\alpha+1)\nu} \end{aligned} \tag{20}$$

We now have the Fourier transform of $c_T(k)$ in terms of the CF of s, which we will now invert to obtain the option price.

$$C_T(k) = \frac{e^{-\alpha k}}{2\pi} \underbrace{\int_{-\infty}^{\infty} e^{-i\nu k} \psi_T(\nu)\, d\nu}_{*_1} = \frac{e^{-\alpha k}}{\pi} \Re\left\{ \int_{0}^{\infty} e^{-i\nu k} \psi_T(\nu)\, d\nu \right\} \tag{21}$$

The second equality holds, since

$$*_1 = \int_{0}^{\infty} e^{-i\nu k} \psi_T(\nu)\, d\nu + \underbrace{\int_{-\infty}^{0} e^{-i\nu k} \psi_T(\nu)\, d\nu}_{*_2} = 2\Re\left\{ \int_{0}^{\infty} e^{-i\nu k} \psi_T(\nu)\, d\nu \right\}$$

$$
\begin{aligned}
*_2 &= \int_0^\infty e^{itk}\psi_T(-t)\,dt = \int_0^\infty \overline{e^{-itk}\psi_T(t)}\,dt = \overline{\int_0^\infty e^{-i\nu k}\psi_T(\nu)\,d\nu} \\
&= \int_0^\infty \cos(\nu k)\psi(\nu)\,d\nu + 0i\int_0^\infty \sin(\nu k)\psi(\nu)\,d\nu
\end{aligned}
$$

We know that the option price is a real function. Obviously, a real function has no imaginary part, which is why $C_T(k) \in \mathbb{R}$ implies that $C_T(k)$ is equal to its complex conjugate $\overline{C_T(k)}$ and we end up with two times the real part $\Re$.

3.2 Numerical Evaluation

In order to calculate the call price $C_T(k)$ via the FFT, we need to rewrite equation (21) in the form of equation (11) (cf. Carr and Madan (1999, p.67,68)). We use Simpson's Rule to replace the integral by a finite sum. This approximation takes three adjacent points and fits a quadratic function through them. First, we truncate the upper integration limit to $a = N\eta$, where N is the (even) number of equidistant points and η is the step size. Next, we set the (odd number of) grid points as $\nu_j = \eta(j-1)$ and with Simpson's Rule weights we get the following equation.

$$
C_T(k) \approx \frac{e^{-\alpha k}}{\pi}\Re\left\{\sum_{j=1}^{N} e^{-i\nu_j k}\psi_T(\nu_j)\frac{\eta}{3}(3+(-j)^j - \delta_{j-1})\right\} \tag{22}
$$

, where $(-j^j)$ and δ_{j-1} ensure that areas will not be covered more than once. In general, δ_n is the Kronecker delta, which is 1 for $n = 0$ and 0 otherwise.

The authors focus on at-the-money (ATM) options due to the fact that these are most actively traded. The price of an ATM option is equal or very near to that of the underlying asset. Put differently, $k \approx 0$, since this generates the same cash flow as the underlying. The strikes are then set as $k_u = -b + \lambda(u-1)$, where λ is used for even spacing, $k_u \in [-b, b]$, $b = \frac{N\lambda}{2}$ spreads them symmetrically around the exercise price and $u = 1, 2, ..., N$. Furthermore, we have to consider the so-called Nyquist relation, i.e. $\lambda\eta = \frac{2\pi}{N}$, which adjusts the sampling rate accordingly when converting the given continuous signal to a sequence of discrete strike prices. After incorporating these changes, the final approximation is given as follows.

$$
C_T(k_u) \approx \frac{e^{-\alpha k_u}}{\pi}\Re\left\{\sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)}e^{ib\nu_j}\psi_T(\nu_j)\frac{\eta}{3}(3+(-j)^j - \delta_{j-1})\right\} \tag{23}
$$

The parameters N, η and λ need to be chosen with care. Increasing N lowers the truncation error, but simultaneously demands $\lambda\eta$ to be small. To obtain many ks around $S(0)$, we would like to set a low spacing λ. At the same time, we want η to be small as well so that the integration grid is finer, leading to a better approximation of the integral.

4 Simulation Study

In this section, we will compare the approach of Carr and Madan to more common pricing methods.

4.1 Computational Modelling

We start with the implementation of the Fourier-based pricing given in equation (23), for which we first need to find the CF of the underlying stock S. From Section 2.1, we know that the BSM assumes the risk neutral density $q_T(s)$ to be a normal distribution, which gives us the following CF, where $\nu = \nu_j - (\alpha+1)i$ (for a derivation see Appendix B).

$$\phi_T(\nu) = exp\Bigg(\bigg(s + \Big(r - \frac{\sigma^2}{2}\Big)T\bigg)i\nu - \frac{\sigma^2\nu^2T}{2}\Bigg) \tag{24}$$

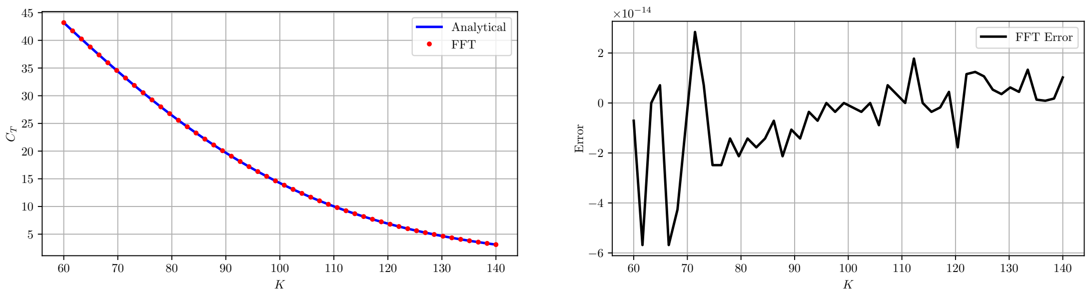
After trying out different settings, the parameters N, η and λ were chosen as suggested by the authors, i.e. $N = 4096, \eta = 0.25$ and $\lambda = 0.00613$. All that is left is choosing the damping factor α. As mentioned earlier, $\alpha > 0$ for the Fourier transform to exist. Other than that, we can set it freely. It has been shown, however, that the choice can depend on both the thickness of the tails of the underlying density and how far away the option is from being ATM. To come up with a suitable choice, it is also best to run multiple simulations with different settings (cf. Lee et al. (2004, p.24-26)). In our case, $\alpha = 3$ yielded the best results in terms of the Mean Squared Error (MSE). For the actual FFT computation we can fortunately rely on predefined libraries, here, `numpy.fft()` has been used.

We compare the FFT approach with the model of Cox et al. (1979, p.239) as well as the Monte-Carlo simulation (cf. Glasserman (2013, p.3-5)). The technical details of these two methods will not be discussed, as these should be well familiar to the reader and can be taken from the code in Appendices A3 and A4. Nonetheless, it is worth mentioning that the accuracy of both methods heavily rely on the step size or simulation size of the CRR and MC, respectively. Even more so for the CRR, which converges for a step size $M \to \infty$ to the analytical solution in (4) (cf. Cox et al. (1979, p.250-254)). In the end, we chose a step size of $M_{CRR} = 500$ for the CRR model and a simulation size $M_{MC} = 100000$ for the MC simulation.

Finally, we carried out the computations for all three models with risk-free rate $r = 0.05$, standard deviation $\sigma = 0.2$, initial stock price $S(0) = 100$, strike price $K = 100$ and Maturity $T = 1$. Moreover, the call prices are given for a total of 50 strike price levels. The Python code for all models are to be found in Appendix A.

4.2 Results

The results of the Fourier-based pricing scheme are displayed in Figure (2). At first glance, we cannot see any clear deviation from the analytical solution. In fact, we can only asses the error, which is given in absolute terms, with the help of panel (2.2). By using the FFT method, we were not only able to compute option prices



(2.1) Call Prices with analytical and FFT method

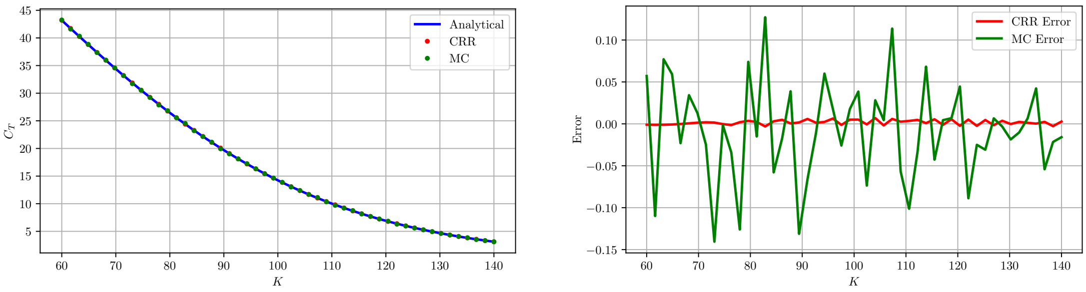
(2.2) Deviations of FFT from analytical prices

Figure 2: European call option price via FFT

with a MSE of $2.8823 \cdot 10^{-28}$, but it also took only 0.0228 seconds to do so. To put this into perspective, calculating the closed form solution takes only 0.0004 seconds.

Now, let us turn to the results of the CRR and MC approach, which are displayed in Figure (3). Assessing the performance again is only possible by looking at panel (3.2). Both methods yielded decent results, but for the CRR the error is slightly less than for the MC method. This result mainly arises since the CRR makes more parametric assumptions than the MC method. Carrying out these two alternative methods took 11.9926 seconds for the CRR and 16.2865 seconds for the MC simulation, which is in line with the common knowledge that MC simulations are computationally very expensive. As an MSE, we obtained $1.0280 \cdot 10^{-5}$ and $3.5370 \cdot 10^{-3}$ for CRR and MC respectively.

In summary, we succesfully replicated the performance of the Fourier-based pricing method and compared it to the CRR model and the MC simulation. Both competitors were not able to match the results of the FFT in terms of speed and accuracy.



(3.1) Call Prices with analytical, CRR and MC method (3.2) Deviations of CRR and MC from analytical prices

Figure 3: European call option price via CRR and MC

5 Discussion

We will now comment on the introduced Fourier-based pricing method and point out its advantages and disadvantages.

First, one of the main advantages of the Carr and Madan inversion is it being a direct application of the FFT algorithm, making it suitable for real time pricing. It is especially useful for calibration purposes where we can find unknown parameters by computing 100s of observed vanilla options for different strike levels simultaneously. There are certainly other Fourier-based approaches that lead to similar accuracy, but many cannot be written in the form of (11). Two comparable methods are that of Raible (2000) and Lewis (2001), who both handle the payoff function and underlying process individually.

Furthemore, the Carr and Madan approach is very easy to use. We only have to calculate one Fourier transform after finding the CF of the underlying process, which is possible in most cases. For example, we can calculate option prices for a variety of commonly used models, such as the Heston model (cf. Lee et al. (2004)). In fact, Carr and Madan implemented their method for a variety of Variance Gamma models.

The method, however, has a serious drawback due to the approximation in (23). First, when setting the upper integration limit to $N\eta$ for the semi-finite integral a truncation error arises. Second, by sampling the integral at equidistant points through η a discretisation error sets in. Lee et al. (2004, p.24), for instance, tried to fine-tune these parameters by strategically running multiple simulations and reporting the resulting errors. But since we have to take the Nyquist relation into account, the model suffers from an inverse relationship between N, η and λ. Increasing N lowers the truncation error, but also gives away computational power, since a lot of calculated option prices do not lie within the region of interest. Decreasing η imposes finer grid spacing, but increases λ, thus reducing pricing accuracy and vice versa. To further mitigate this problem, we can use for example a *fractioncal* FFT (FrFFT) method that generalizes the FFT. This approach basically splits the computation into three $2N$-point FFT's to produce $1N$-point FrFT, allowing us to choose λ and η independently, while speeding up calculations even further (cf. Chourdakis (2005, p.7)).

It is also not possible to price american (or even exotic) options, which can be exercised any time. These, however, are the most prominently traded types of options. There have been some implementations (cf. Lord et al. (2008)), but using e.g. MC simulations seems much more user-friendly in these situations.

In total, the Fourier-based pricing scheme introduced by Carr and Madan represents a valuable method for option pricing. It can serve as a good introduction to the topic of Fourier-based methods, while a large body of literature shows how to adjust it to more complex model specifications.

6 Conclusion

In this paper, we have revisited the basics of option pricing under the assumptions of the Black-Scholes model. Furthermore, we have introduced the Fourier Transformation and explained its numerical evaluation via the DFT. We have also shown that the DFT is a candidate for a divide-and-conquer approach that minimises its runtime complexity, which ultimately led to the FFT. We implemented the approach of Carr and Madan using the FFT and compared its performance to more simplistic approaches, including the CRR model and an MC simulation. The results confirmed the superiority of the FFT. First, it can easily be applied whenever we are able to formulate the characteristic function of a given process. Second, the FFT shows some remarkable degree of accuracy. Last, the speed of this approach is unmatched.

Fourier-based pricing, and FFT applications in general, is still an active field of research. It has been shown that extending Carr and Madan's approach can yield even better results. But, more importantly, it has not yet been proven that the FFT's complexity is in fact bounded at $\mathcal{O}(Nlog(N))$.

References

Abate, J. and W. Whitt (1992): "The Fourier-series method for inverting transforms of probability distributions," *Queueing systems*, 10, 5–87.

Bakshi, G., C. Cao, and Z. Chen (1997): "Empirical performance of alternative option pricing models," *The Journal of finance*, 52, 2003–2049.

Black, F. and M. Scholes (1973): "The Pricing of Options and Corporate Liabilities," *Journal of Political Economy*, 81, 637–654.

Carr, P. and D. Madan (1999): "Option valuation using the fast Fourier transform," *The Journal of Computational Finance*, 2, 61–73.

Champeney, D. C. (1987): *A handbook of Fourier theorems*, Cambridge University Press.

Chourdakis, K. (2005): "Option pricing using the fractional FFT," *Journal of computational finance*, 8, 1–18.

Cochran, W. T. et al. (1967): "What is the fast Fourier transform?" *Proceedings of the IEEE*, 55, 1664–1674.

Cooley, J. W. and J. W. Tukey (1965): "An algorithm for the machine calculation of complex Fourier series," *Mathematics of computation*, 19, 297–301.

Cox, J. C. and S. A. Ross (1976): "The valuation of options for alternative stochastic processes," *Journal of financial economics*, 3, 145–166.

Cox, J. C., S. A. Ross, and M. Rubinstein (1979): "Option pricing: A simplified approach," *Journal of financial Economics*, 7, 229–263.

Glasserman, P. (2013): *Monte Carlo methods in financial engineering*, vol. 53, Springer Science & Business Media.

Heston, S. L. (1993): "A closed-form solution for options with stochastic volatility with applications to bond and currency options," *The review of financial studies*, 6, 327–343.

Lee, R. W. et al. (2004): "Option pricing by transform methods: extensions, unification and error control," *Journal of Computational Finance*, 7, 51–86.

Lewis, A. L. (2001): "A simple option formula for general jump-diffusion and other exponential Lévy processes," *Available at SSRN 282110.*

Lord, R., F. Fang, F. Bervoets, and C. W. Oosterlee (2008): "A fast and accurate FFT-based method for pricing early-exercise options under Lévy processes," *SIAM Journal on Scientific Computing*, 30, 1678–1705.

Raible, S. (2000): "Lévy processes in finance: Theory, numerics, and empirical facts," Ph.D. thesis, PhD thesis, Universität Freiburg i. Br.

SAIDI, A. (1994): "Decimation-in-time-frequency FFT algorithm," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, vol. 3, III–453.

SHREVE, S. E. (2004): *Stochastic calculus for finance II: Continuous-time models*, vol. 11, Springer Science & Business Media.

SONI, M. AND P. KUNTHE (2011): "A General comparison of FFT algorithms," *Pioneer Journal Of IT & Management*.

STEIN, E. M. AND J. C. STEIN (1991): "Stock price distributions with stochastic volatility: an analytic approach," *The review of financial studies*, 4, 727–752.

ZHANG, B. AND C. OOSTERLEE (2014): "Pricing of early-exercise Asian options under Lévy processes based on Fourier cosine expansions," *Applied Numerical Mathematics*, 78, 14–30.

Appendices

A Code

A 1 Python Packages and Parameters

```python
"""
System Specifications:
  OS:  macOS Catalina (Version 10.15.7, 64-bit)
  CPU: 2.4 GHz Quad-Core Intel Core i5
  RAM: 16GB 21333 MHz LPDDR3
  Python-Version 3.9.2 64-bit
"""

import math
import numpy as np
from numpy.fft import fft
from scipy.stats import norm
import matplotlib.pyplot as plt
import cmath
import timeit

M_crr = 500
M_mc = 100000
S0 = 100.00
K = 100.00
T = 1.
r = 0.05
sigma = 0.3
i = complex(0, 1)
strikes = np.linspace(S0*0.6, S0*1.4, 50)

np.random.seed(1993)
```

A 2 Fast Fourier Transformation

```python
def BSM_analytical(S0, K, T, r, sigma):
    d_1 = (np.log(S0/K)+(r+sigma**2 /2)*T)/(sigma*np.sqrt(T))
    d_2 = (np.log(S0/K)+(r-sigma**2 /2)*T)/(sigma*np.sqrt(T))
    return S0*norm.cdf(d_1)-K*np.exp(-r*T)*norm.cdf(d_2)

def BSM_CF(v, s, T, r, sigma):
    return np.exp((s+(r-sigma**2 /2)*T)*i*v-sigma**2 *v**2 *T/2)

```

```python
def Carr_Madan_FFT(S0, K, T, r, sigma):
    k = np.log(K/S0)
    s = np.log(S0/S0)
    N = 2**12
    lam = 0.00613
    eta = 2*np.pi/(N*lam)
    b = 0.5*N*lam-k
    u = np.arange(1, N+1)
    v_j = eta*(u-1)
    alpha = 3
    v = v_j-(alpha+1)*i

    psi = np.exp(-r*T)*(BSM_CF(v, s, T, r, sigma)/(alpha**2 +alpha-v_j**2 +i*(2*alpha+1)*v_j))

    Kronecker = np.zeros(N)
    Kronecker[0] = 1
    j = np.arange(1, N+1)
    Simpson = (3+(-1)**j -Kronecker)/3

    FFT_function = np.exp(i*b*v_j)*psi*eta*Simpson
    C_T = np.exp(-alpha*k)/np.pi*(fft(FFT_function)).real

    return C_T[int((k+b)/lam)]*S0
```

A 3 Cox-Ross-Rubinstein

```python
def CRR(S0, K, T, r, sigma, M):
    dt = T/M
    beta = (math.exp(-r*dt) + math.exp((r+sigma**2)*dt))/2
    u = beta+math.sqrt(beta**2 -1)
    d = 1/u
    q = (math.exp(r*dt)-d)/(u-d)

    S = np.zeros((M+1, M+1))
    V = np.zeros((M+1, M+1))

    for i in range(1, M+2):
        for j in range(1, i+1):
            S[j-1, i-1] = S0*u**(j-1)*d**(i-j)

    def payoff_call(x):
        return np.maximum(x-K, 0)

```

```
    V[:, M] = payoff_call(S[:, M])

    for i in range(M-1, -1, -1):
        for j in range(0, i+1):
            V[j, i] = np.exp(-r*dt)*(q*V[j+1, i+1]+(1-q)*V[j, i+1])

    return V[0, 0]
```

A 4 Monte Carlo Simulation

```
def MC(S0, K, T, r, sigma, M):
    S = np.empty((M+1))
    V = np.empty((M+1))
    X = np.random.normal(0, 1, M+1)

    def payoff_call(x):
        return np.maximum(x-K, 0)

    for i in range(0, M+1):
        S[i] = S0*math.exp((r-sigma**2/2)*T+sigma*math.sqrt(T)*X[i])
        V[i] = payoff_call(S[i])

    return math.exp(-r*T)*np.mean(V)
```

B Characteristic Function

In the BSM the underlying stock also follows a GBM under the unique EMM Q, i.e. it satisfies the following stochastic differential equation.

$$dS(t) = rS(t)dt + \sigma S(t)dW(t)$$

, with local drift rate μ, local diffusion coefficient σ and Brownian motion W. We can solve this for S using Itô's formula, which can be formulated in Taylor expansion form (cf. Shreve (2004, p.147)).

$$\begin{aligned} df(X(t)) = &\left(\partial_1 f(t, X(t)) + \partial_2 f(t, X(t))\mu(t) + \frac{1}{2}\partial_{22} f(t, X(t))\sigma^2(t)\right)dt \\ &+ \partial_2 f(t, X(t))\sigma(t)dW(t) \end{aligned}$$

If we consider the underlying to be log transformed, then with $f(S(t)) = ln(S(t)) = s$ we get

$$df(S(t)) = (\mu - \frac{\sigma^2}{2})dt + \sigma dW(t)$$

One of the key assumptions of the BSM is that it imposes a normal distribution for the log return, thus we can specify the risk-neutral density of s under Q as follows.

$$q(s_T) = \frac{1}{\sqrt{2\pi\sigma^2 T}} exp\left[-\frac{\left(s(t) - s(0) - \left(r - \frac{\sigma^2}{2}\right)T\right)^2}{2\sigma^2 T}\right]$$

The CF of the normal distribution is in general of the form

$$\phi(t) = exp\left(it\mu - \frac{\sigma^2 t^2}{2}\right)$$

Combining these observations we find the CF of the underlying as in (24).

$$\phi_T(\nu) = \int_{-\infty}^{\infty} e^{i\nu s_T} q(s_T)\, ds_T = exp\left(\left(s + \left(r - \frac{\sigma^2}{2}\right)T\right)i\nu - \frac{\sigma^2\nu^2 T}{2}\right)$$

Statutory Declaration

I herewith declare that I have composed the present work myself and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The work in the same or similar form has not been submitted to any examination body and has not been published.

Place, Date

Signature