Pricing Options using Fourier Transform

Stelian Cujba Giovanni Dacchile

January 2026

1 Introduction

In this article we will explore how the Fourier Transform can be used to price options across a lot of strike prices in an efficient way following the method outlined in Carr and Madan (1999). It is important to understand that this technique doesn't represent a model in itself, but rather, an elegant way to approximate solutions to the models that follow a Lévy process, but don't have a closed-form solution or a "nice" Probability Density Function. Of course, there is always the option to apply a Monte Carlo Simulation or some Numerical Method, however, this becomes very slow when you have to compute hundreds or thousands of prices at the same time, which is exactly what the Fourier Transform solves.

We will start with an understanding of the Fourier Transform (FT) and the Discrete Fourier Transform (DFT) starting from the Fourier Series. Afterwards, we will explore what characteristic functions are, and what they have to do with Probability Density Functions. Thirdly, we get an understanding of the last puzzle piece to be able to apply the technique, which are Lévy Stochastic Processes, having as examples Geometric Brownian Motion and the Poisson Process. After exploring the background necessary to price options, we will outline and implement the framework on a simple model, the Black-Scholes Model, and we will measure its accuracy and computational speed. Since the Black-Scholes Model already has a closed-form solution, it doesn't make sense to apply this technique to price options using this model in practice, however, our goal is to outline the framework and to demonstrate the advantages of using the FT pricing technique. Furthermore, the same ideas applied on the Black-Scholes Model can be adapted to other models, taking full advantage of the technique.

You can check the implementation of the example outlined in Section 5 on GitHub (LINK).

The goal of this article is not to provide a rigorous mathematical outline of the technique, but rather, give our readers a general understanding of theory and practice. For an academically rigorous treatment of Option Pricing with Fourier Transform we recommend reading Carr and Madan (1999) and Matsuda (2004). Additionally, you can find a series of resources at the end of the article on various parts discussed, which we think can be useful to get a better understanding or intuition of different concepts. We also assume you have a basic understanding of options and statistics/probability.

2 Fourier Transform

2.1 Introduction to the Fourier Series

To present you how options can be priced through Fourier Transforms , it is essential to introduce you to the world of Fourier analysis. There are uncountable applications of Fourier analysis, for example encoding and retrieving data in the form of bits through electromagnetic waves, allowing our devices to communicate fast and accurately. However, at the purpose of this article we will mostly focus on Finance applications.

It is imperative to present you the backbone of Fourier analysis, the Fourier series:

$$f(x) = a_0 + \sum_{n=0}^{\infty} [a_n cos(n\omega_0 x) + b_n sin(n\omega_0 x)] \tag{1}$$

Here $\omega_0 = \frac{2\pi}{T}$ is the fundamental frequency that is present and it is the formula to make sure that our function has the same periodicity of sinusoidal waves. We will often refer to functions with period T=2π to simplify the equations.
The Coefficients a_0, a_n, b_n are found by orthogonality conditions and are defined by the following equations:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} \frac{1}{\sqrt{2}} f(x)\, dx \tag{2}$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) cos(nx)\, dx \tag{3}$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) sin(nx)\, dx \tag{4}$$

These coefficients above are derived under an orthonormal basis; however, when the normality condition is relaxed the convention for these coefficients may vary accordingly. Also notice that the Fourier series defined above can only approximate a restricted class of functions, namely, periodic functions (periodicity that can always be satisfied on a bounded domain) in the L^2 space.

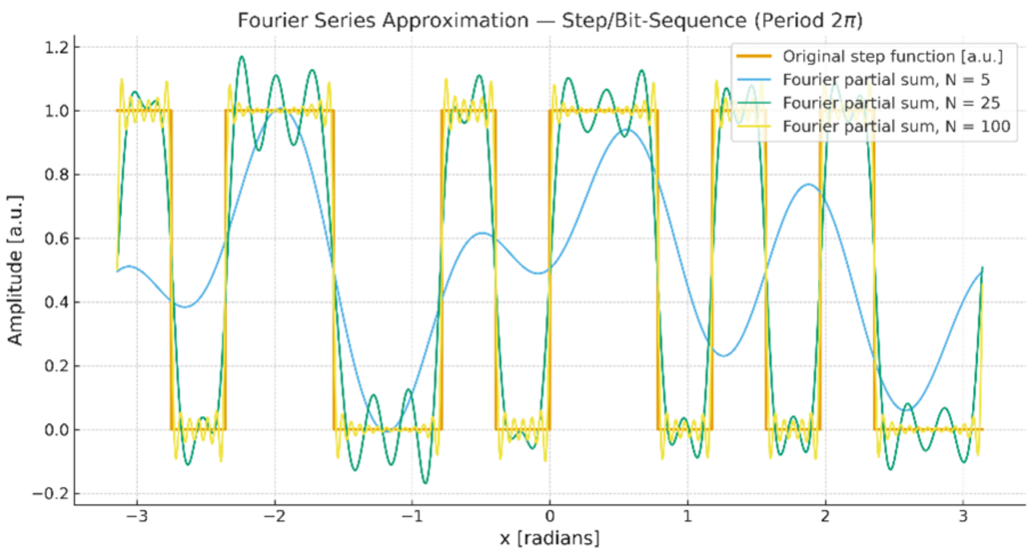


Figure 1: Approximation of a binary signal with Fourier partial sums

The Fourier Series is based on a trigonometric polynomial which resembles the Taylor series. In fact, as we can approximate functions using Taylor series, we can do the same with the Fourier series. The key difference lies in the fact that here we approximate the function with a weighted sum of its frequency's components, i.e., a sum of sinusoids of different frequencies.

2.2 From Fourier Series to Fourier Transforms and FFT

Limiting our approximation to Periodic L^2 functions may be restrictive in more the one application, fortunately through the extension of the series to the complex plane we can go from a discrete to a continuous sum which is a key component for many models of option pricing. Utilizing the notorious Euler's formula:

$$e^{i\theta} = cos(\theta) + isin(\theta) \tag{5}$$

Thus, we can enhance our initial series by meaning of compactness in the following way:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx} \tag{6}$$

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-inx}\, dx \tag{7}$$

This form of the Fourier series is nice in the fact that it provides us a nice base to explain what it does with a simple intuition. Observing the integral of the coefficients c_n we can notice how it is divided by the length over which it is integrated, that is an average. Indeed, in the inside the integral, we multiply $f(x)$ by e^{-inx}, which, according to Euler's formula, represents a rotation around the unit circle in the complex plane.

In other words, we are "wrapping" the function $f(x)$ around the circle with a frequency n. Taking the average of this wrapped function gives the "center of mass" (mean value). It turns out that this center is always close to zero except when the frequency n with which we are wrapping the graph around the circle is one of the latent frequency components of the original function. *Thus, the Fourier transform identifies which circular rotations (frequencies) produce a nonzero "center," revealing the hidden periodic components of $f(x)$.* Each c_n representing the distance between the center of mass and the origin tells us how much of the frequency n is present in the signal.

If we extend our period by imposing $T \to \infty$ and apply the principle behind Parseval Theorem we arrive at the continuous form:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega t} G(\omega)\, d\omega;\ with\ G(\omega) = \int_{-\infty}^{\infty} e^{i\omega t} f(t)\, dt \tag{8}$$

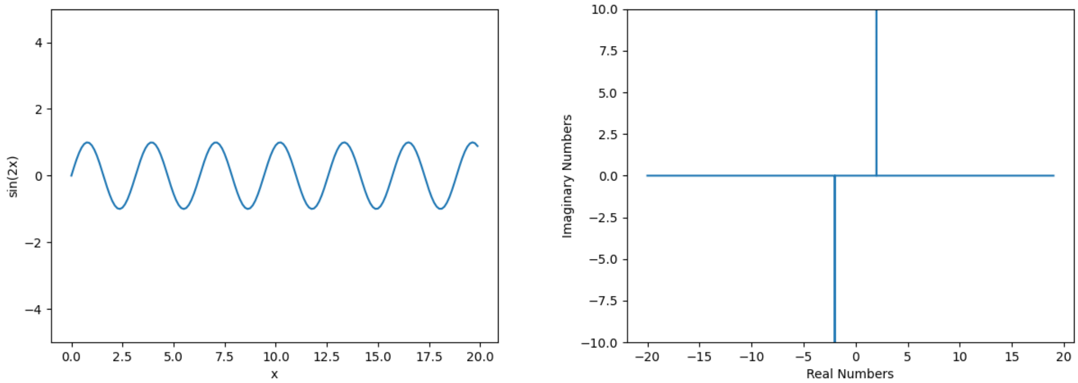
Finally, we obtained our Fourier Transform. Here the focus is not anymore on the pure approximation of a function with a sinusoidal series, but the truly outstanding result is that through the integral transformation that we are applying, we obtained basically two tools:

- $G(\omega)$, which allows us to move from a time to a frequency domain. We will shortly see how this come in handy in several situations, we call this transformation the Fourier Transform.
- The first expression, instead, is the inverse Fourier transform because it allows us to go back to the time domain.

2.3 Discrete Fourier Transform

In practice we will almost always have have finite data, think about the interest rate at different points in time or the stock price (even if it may resemble a continuous path due to the time laps of microseconds between each trade). This makes possible for us to compute numerically the Fourier transform approximating a function $f(t)$ in the time domain by a discrete sample of N points.

Let Δt be the time interval between each sample. It is trivial to express this interval as $\Delta t = T/N$,



(a) sin(2x) (b) DFT approximation of sin(2x)

Figure 2: Approximate Discrete Fourier Transform of sin(2x)

where T is the total sampling time. For example, by sampling N = 10 observation over a total sample time of T = 10 sec we will have a time interval $\Delta t = 1$ sec.

We therefore consider a discrete number of time points $t_n = n\Delta t$, with n= 0,1,. . . ,N-1 and evaluate $g(t)$ at those time points (time domain sampling). Next, we do the same for the Fourier transform by taking N samples of $G(\omega)$ from $\omega_k = k\Delta\omega$, with k= 0,1,. . . ,N-1.

We consider $\Delta\omega = \frac{2\pi}{T} = \frac{2\pi}{N\Delta t}$, since we want to have only angular frequency components that can wrap our function around the circle with an integer number of rotations k; we have $G(\omega_k) = G(k\Delta\omega) = G(k\frac{2\pi}{N\Delta t})$. A DFT does simply describe the link between these 2 sampled components:

$$\mathcal{G}(\omega_k \equiv k\frac{2\pi}{N\Delta t}) = \sum_{n=0}^{N-1} g(t_n \equiv n\Delta t)exp(i\omega_k t_n) \tag{9}$$

$$g(t_n \equiv n\Delta t) = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{G}(\omega_k \equiv k\frac{2\pi}{N\Delta t})exp(-i\omega_k t_n) \tag{10}$$

2.4 Applications Overview

When dealing with models such Heston model, Variance Gamma, NIG (normal inverse gaussian) or other stochastic volatility models the CF $\phi(u)$ is known, but the PDF (probability density function) $f(x)$ is often unknown in closed form or difficult to manage.

There are Models like the Lewis or Carr-Madan that retrieve all the necessary information that we need from the PDF through the Inverse Fourier Transform of the CF, computing prices in the frequency domain and bringing them back to the real domain without even trying to manage the PDF in a closed form. A second important feature of this approach is that it builds the basis to apply numerical methods still based on the inverse Fourier transform such FFT and COS method that greatly enhance computational efficiency. We will see that to work in the frequency domain it will be necessary to apply the Fourier transform also to the payoff function of the option.

In finance, beyond option pricing, there are many applications of Fourier transforms and series. We can mention the spectral analysis of financial time series: a method used to decompose price data in frequency components through the approximation of the log-returns pattern via a DFT (Discrete Fourier Transform). This method makes it possible to empirically approximate a spectral density function (the distribution of the frequencies rather than the log-returns themselves), furthermore analyzing the spectral density it is possible to uncover patterns especially by means of periodicity. It is therefore used often for intraday patterns.

The same kind of analysis can be used to filter out unwanted frequencies (noise) while analyzing the evolving price pattern for example in algo-trading strategies.

Lastly, we can mention a further application that resemble the methods seen in option pricing but applied to risk management. Indeed, we can also use Fourier inversion of characteristic functions to enable efficient and accurate Value-at-Risk (VaR) or Expected Shortfall computation.

3 Characteristic Functions

A characteristic function is essentially the Fourier Transform of a probability distribution. The function itself contains all the information about a random variable, in a similar way how a Probability Density Function (PDF) does. For a random variable X, the characteristic function is:

$$\phi(\omega) = \mathcal{F}[P(X)] = \int_{-\infty}^{\infty} e^{i\omega X} P(X)\, dx = E[e^{i\omega X}] \tag{11}$$

Thus, while the PDF tells us "how probable" each outcome is, the characteristic function does the same, but in the frequency domain. In many option pricing models, it is difficult to manage, or sometimes even impossible, to get the PDF of the asset's log-price in a closed form. However, in many cases, it's much easier to derive the characteristic function, and such, when pricing options, instead of integrating over the unknown density function, we can integrate over the Inverse Fourier Transform of the characteristic function, which we can derive analytically.

Using the Taylor series expansion of an exponential function $e^{i\omega x}$ centered around the point x=0 we get that:

$$e^{i\omega x} = e^{i\omega 0} + i\omega e^{i\omega x}(x-0) + \frac{1}{2!}(i\omega)^2 e^{i\omega x}(x-0)^2 + ... \tag{12}$$

$$e^{i\omega x} = 1 + i\omega x + \frac{1}{2!}(i\omega x)^2 + \frac{1}{3!}(i\omega x)^3 ... \tag{13}$$

And thus, a characteristic function can be rewritten as:

$$\phi(\omega) = \mathcal{F}[P(X)] = \int_{-\infty}^{\infty} e^{i\omega X} P(X)\, dx = \int_{-\infty}^{\infty} (1 + i\omega x + \frac{1}{2!}(i\omega x)^2 + \frac{1}{3!}(i\omega x)^3 ...) P(X)\, dx \tag{14}$$

To be noted, some important properties about characteristic functions are:

- They always exist (unlike moment generating functions)
- A characteristic function uniquely determines the distribution i.e. there exists a one-to-one relationship between distribution functions and characteristic functions
- The probability density function(if it exists) can be obtained by an Inverse Fourier Transform of the characteristic function:

$$P(x) = \mathcal{F}^{-1}[\phi(\omega)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega X} \phi(\omega)\, d\omega \tag{15}$$

The last property is the backbone of Fourier-based option pricing, where we replace the often-unknown PDF, with the usually known characteristic function.

To get a better understanding of why a characteristic function uniquely determines a distribution we can use the "Cumulant Generating Function" which is defined as the natural log of the characteristic function:

$$\Psi_X(\omega) = ln[\phi_X(\omega)] \tag{16}$$

And the n-th cumulant being:

$$cummulant_n = \frac{1}{i^n} \frac{\partial^n \Psi(\omega)}{\partial \omega^n} \tag{17}$$

where ω is centered at 0. These are just the n-th partial derivatives of the Cumulant Generating Function divided by i^n. Knowing these cumulants we can determine the mean, variance, skewness, excess kurtosis and so on, of the random variable X by:

$$\text{Mean} = cummulant_1 \qquad \text{Variance} = cummulant_2$$
$$\text{Skewness} = \frac{cummulant_3}{(cummulant_2)^{3/2}} \qquad \text{Kurtosis} = \frac{cummulant_4}{(cummulant_2)^2}$$

4 Lévy Processes

A Stochastic Process, X, is a Lévy Process if it is right-continuous with left limits (cadlag) / adapted(non-anticipating), and possesses the following properties:

- Independent Increments
- Stationary Increments: X_{t+h}–X_t has the same distribution as X_h. In other words, the distribution of increments does not depend on t.
- $X_0 = 0$
- X_t is continuous in probability: $\lim_{h\to\ 0} P(|X_{t+h} - X_h| \geq \epsilon) = 0$

Brownian Motion Process

Lévy processes are mainly discontinuous (jump) processes; however, this is not true in a strict sense. A Lévy process can have a continuous sample path. The only example of such a process is the Standard Brownian Motion process. A standard Brownian Motion process is a Lévy Process defined on $\mathbb{R}$ such that:

- $B_t \sim Normal(0, t)$
- $X_t(\omega)$ is continuous in t for every ω of Ω

Poisson Processes and Compound Poisson processes

Let $(\tau_i)_{i\geq 1}$ be a sequence of independent exponential random variables with a parameter λ and $T_n = \sum_{i=1}^{n} \tau_i$. Thus, a Poisson process with intensity lambda is:

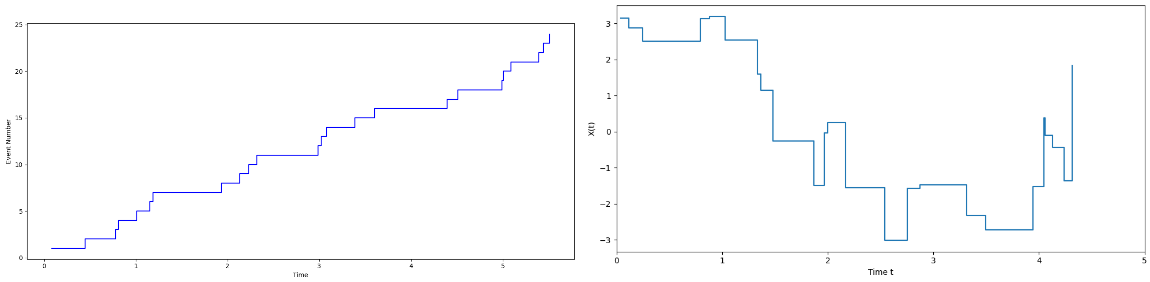
$$N_t = \sum_{n\geq 1} 1_{t\geq T_n} \tag{18}$$

The function N_t counts the number of random times (T_n) at which the specified event occurs during a time period between 0 and t, where $(T_n$–$T_{n-1})$ is an i.i.d. sequence of exponential variables. Therefore, each possible N_t is represented as a non-decreasing piecewise constant function.

We also have a more general version of the Poisson Process called a compound Poisson Process, defined as:

$$X_t = \sum_{i=1}^{N} Y_i \tag{19}$$

where Y_i are i.i.d jump sizes with the probability density function f.
Since a Compound Poisson Process simplifies to a Poisson Process if Y = 1 (constant jump size of 1), a compound Poisson Process has jump sizes that are independent and identically distributed with the probability density function f.



(a) Poisson Process with $\lambda = 3$ and T = 5

(b) Compound Poisson Process with jumps sampled from a Normal Distribution

Figure 3: Poisson Processes

5 Option Pricing with Fourier Transform: Black-Scholes Example

In an arbitrage-free market, prices of assets can be computed as expected payoffs at maturity under $\mathbb{Q}$, the risk neutral probability measure, discounted by a risk-free interest rate r:

$$S_t = e^{-r(T-t)} E^{\mathbb{Q}}[S_T] \tag{20}$$

Let K be a strike price and T be the maturity of the option. We will work with call options for simplicity, but a similar process can be applied for put options. The price of a plain vanilla call option is:

$$C(t, S_t) = e^{-r(T-t)} E^{\mathbb{Q}}[(S_T - K)^+] \tag{21}$$

Using $\mathbb{Q}$ as a probability density function of a terminal asset S_T, we can rewrite the call price as:

$$C(t, S_t) = e^{-r(T-t)} \int_K^{\infty} (S_T - K)\mathbb{Q}(S_T) dS_T \tag{22}$$

In Black-Scholes we assume that S_T is a log-normal random variable, for which we have the density function. Therefore, BS option pricing comes down to a simple integration problem since all parameters are known:

$$C_{BS}(t, S_t) = e^{-r\tau} \int_K^{\infty} (S_T - K) \frac{1}{S_T\sqrt{2\pi\sigma^2\tau}} \exp\left[-\frac{\left\{ \ln S_T - \left(\ln S_t + \left(r - \frac{1}{2}\sigma^2 \right) \tau \right) \right\}^2}{2\sigma^2\tau} \right] dS_T \tag{23}$$

However, for general Lévy processes, $\mathbb{Q}(S_T)$ cannot be expressed using special functions of mathematics or isn't known, therefore, we can't price plain vanilla options using this method. To get around this, a potential solution is to use the fact that for general exponential Lévy processes, the characteristic functions are always known in closed forms or can be expressed in terms of special functions of mathematics, even though their probability density functions are not. We have also seen previously that there is a direct relationship between probability density functions and the characteristic functions. So, all we have to do is to rewrite the above equations in terms of characteristic functions instead of probability density.

To simplify our equations, without loss of generality, assume t=0. We can also apply a change of variable from S_T to $ln(S_T)$:

$$C(T,K) = e^{-rT} \int_{\ln K}^{\infty} \left(e^{\ln S_T} - e^{\ln K}\right) \mathbb{Q}(\ln S_T)\ d\ln S_T \tag{24}$$

To simplify notation, from now on, let $s_T = ln(S_T)$ and $k = lnK$. We know that a characteristic function of s_T is a Fourier Transform of its density function $\mathbb{Q}(s_T)$:

$$\phi_t(\omega) = \mathcal{F}[\mathbb{Q}(s_T)]\,(\omega) = \int_{-\infty}^{\infty} e^{i\omega s_T}\, \mathbb{Q}(s_T)\, ds_T \tag{25}$$

If we use the BS model, with the normal density given by (23), and replace it into the characteristic function formula (11), the call pricing function is given by:

$$C(T,k) = e^{-rT}\, \mathbb{E}^{\mathbb{Q}}[e^{s_T}] \tag{26}$$

and if we apply the Martingale property, we realize that the standard call option pricing is not square-integrable, which is necessary for the Fourier Transform. This is due to the log-strike approaching a constant at $-\infty$ which is not zero. To solve this problem, Carr-Madan(1999) uses a dampening factor and introduces the modified call price as:

$$C_{mod}(T,k) = e^{\alpha k} C(T,k) \tag{27}$$

where C_{mod} is expected to satisfy the integrability condition by choosing an $\alpha > 0$. Then, considering a FT of the modified call price:

$$\Psi_T(\omega) = \int_{-\infty}^{\infty} e^{i\omega k}\, C_{\mathrm{mod}}(T,k)\, dk \tag{28}$$

and thus, we can obtain the call price C(T,k) by an inverse Fourier Transform of $\Psi_T(\omega)$:

$$C(T,k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega k}\, \Psi_T(\omega)\, d\omega \tag{29}$$

Now, Carr-Madan derive an analytical expression of the modified FT in terms of the characteristic function getting to:

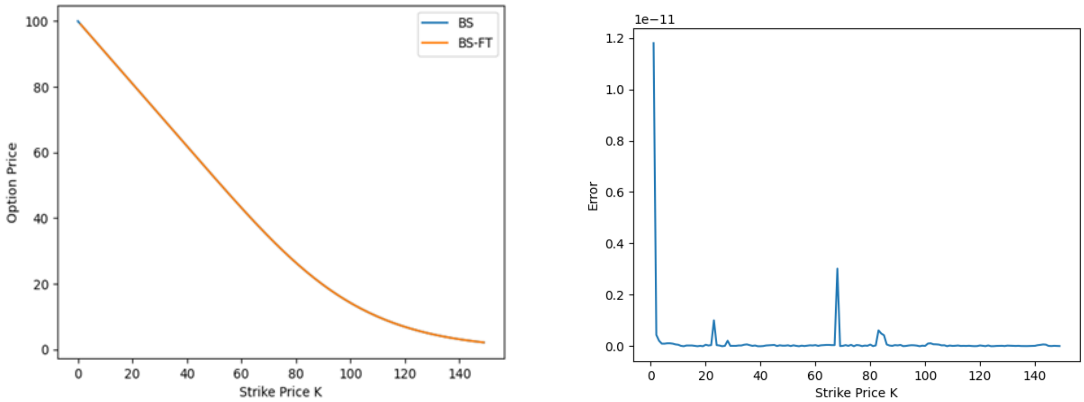
$$\Psi_T(\omega) = \int_{-\infty}^{\infty} e^{i\omega k} e^{\alpha k} e^{-rT} \int_{k}^{\infty} \left(e^{S_T} - e^{k}\right) \mathbb{Q}(S_T)\, dS_T\, dk = \frac{e^{-rT}\, \phi_T(\omega - (\alpha+1)i)}{\alpha^2 + \alpha - \omega^2 + i(2\alpha+1)\omega} \tag{30}$$

And thus, a call pricing function is obtained as:

$$C(T,k) = \frac{e^{-\alpha k}}{2\pi} \int_{-\infty}^{\infty} e^{-i\omega k}\, \frac{e^{-rT}\, \phi_T(\omega - (\alpha+1)i)}{\alpha^2 + \alpha - \omega^2 + i(2\alpha+1)\omega}\, d\omega \tag{31}$$

Where ϕ is a characteristic function of a log terminal stock price s_T

Now, we can implement the formula obtained above using a decay rate parameter, alpha, of 1. As we can see below, the BS-FT call price and the original BS call price are very close (errors in



(a) Call prices obtained using BS and BS-FT

(b) Pricing errors between the theoretical price (BS) and the obtained price (BS-FT)

Figure 4: The parameters used to price the options were: $S_0 = 100$, $\sigma = 0.3$, r = 0.05 and T = 1.

the magnitude of 10^{-11}), and this makes sense because, after all, the BS-FT formula is just the frequency representation of the original BS formula.

If we consider different options of the decay parameter, α, it should lead to the same theoretical results, and is used only with the goal of making the price integrable, and Matsuda(2004) determined that it doesn't have much importance as long as $\alpha \in \mathbb{R}^+$

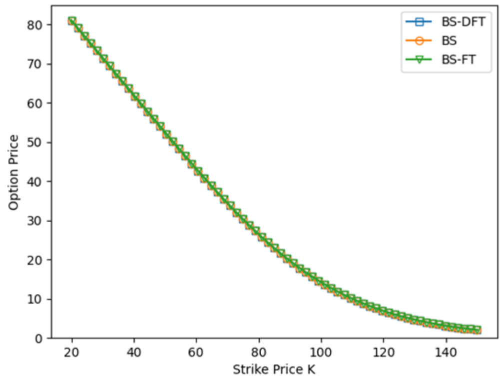
Additionally, if we consider FT option pricing for near maturity, deep OTM and ITM calls, since their prices approach their intrinsic value, this makes the Inverse Fourier Transform integrand highly oscillatory, making the integration process slow and difficult. There is a solution around this described in the Carr-Madan paper, however, this is beyond this article's scope.

Lastly, we can implement another improvement for this technique. Even though the numerical integration for near maturity options causes no big pricing errors, it makes the evaluation of the FT price very slow, and if we want to calculate the price of hundreds or thousands of options, this speed becomes an issue (which is the same problem the Monte-Carlo simulation runs into). To solve this issue, we will implement the Discrete Fourier Transform to approximate the equation. We will follow the DFT described in Matsuda (2004).

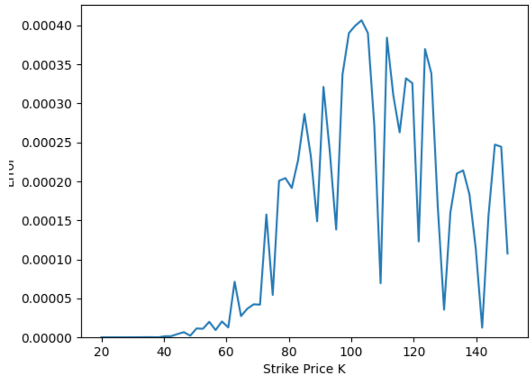
After implementing the DFT using an N=4096, which is the number of samples we use to estimate the FT, we get similar prices obtained by the other 2 methods described above (in the magnitude of 10^{-4}) The higher errors are due to the approximation and can be further minimized by the choice of the grid size. The choice of N influences directly the grid size, and in turn, the accuracy of the approximation. Thus, it is recommended to keep a small grid size (bigger N), even in cases where you don't need to price N options. In these cases, you can just "snap" the strike prices you are interested onto the grid, getting a good approximation of the option price.

The Manual Implementation of the DFT has a time complexity of $O(N^2)$, which can also be a further bottleneck to the efficiency of the algorithm. However, we can take advantage of the Fast Fourier Transform algorithm, which is an optimized implementation of the DFT with a time complexity of $O(NlogN)$. (For the implementation we have used the fft() function in numpy)

When we investigate the CPU time required for these operations, we observe a significant improvement over the numerical integration method, since in the DFT method, we compute N (4096) option prices at the same time. Additionally, the implementation of the Fast Fourier Transform



(a) Call prices obtained using BS and BS-FT and BS-DFT



(b) Pricing errors between the theoretical price (BS) and the obtained price (BS-DFT)

Figure 5: The parameters used to price the options were: $S_0 = 100$, $\sigma = 0.3$, r = 0.05 and T = 1.

improves the CPU time even more. The results using the Python implementation on our GitHub, ran on a personal computer are:

BS	0.06s
BS-FT	87.08s
BS-DFT	26.22s
BS-DFT-FFT	0.19s

Figure 6: CPU times required to price 4096 options

As we can see, there is no real advantage in implementing any of this, if we consider the basic Black-Scholes model, since this model already has a closed-form solution that can be computed analytically.

However, the framework described in this article naturally extends to other models such as Heston, Variance-Gamma, Merton-JD or Normal Inverse Gaussian. These models usually don't have available probability densities in closed form, but the characteristic function is known. As a result, the pricing formula we discussed remains unchanged in structure, the modification required being to substitute the characteristic function of the model. This being another advantage of the Fourier Transform method, as once the FFT infrastructure is implemented, switching between the models requires minimal effort while maintaining high computational efficiency.
Even though there are always other methods you can use to price such options, like a Monte Carlo simulation or a Numerical Approximation, the DFT-FFT method outpaces most of them since we can price a multitude of options "in-one-go", which takes relatively little processing power.

Resources

- 3Blue1Brown series on Fourier Transform on YouTube (Link)
- About the Fast Fourier Transform Algorithm on YouTube (Link)
- Section 5 and 8.6, 8.7 on DFT maths in Matsuda paper (Link)
- Implementation of everything described in the paper (Link)

Sources

- "Elias Stein, Rami Shakarchi: Fourier Analysis, an introduction. Princeton University Press, 2002"
- "Statistical inference. George Casella , Roger L. Berger"
- "Oosterlee, C. W., Grzelak, L. A. (2019). Mathematical Modeling and Computation in Finance"
- "Peter Carr, Dilip. B. Madan, Option Valuation using the fast Fourier transform, 1999"
- "Max Matsuda, Introduction to Option Pricing with Fourier Transform: Option Pricing with Exponential Lévy Models (2004)"