Fast Fourier Transform on Option Pricing

Haokai Hu, Xuan Ji, Chenyu Wu

Abstract

Fourier Transform method is widely used in current financial mathmatic. It simplify the calculation when dealing with characteristic functin of log stock price. The Fast Fourier Transform is a further improve of orginal fourier transform method, it heavily reduced the complexity of numerous options pricing. We simulate the option pricing processes with different methods on python, and conclude Fast Fourier Transform is a efficient method on option pricing.

1. Mathematical Background

I Black-Scholes Model on Pricing Options:

There are 4 assumptions when we use B-S model on option pricing:
1. Option Price follows brownian motion. (Set μ as expected return rate, σ as volatilitym and dW as standard brownian motion process)

$$\frac{dS}{S} = \mu dt + \sigma dW \tag{1}$$

2. The volatility and risk-free rate is constant in option's maturity
3. There are no transaction costs in buying the option.
4. No dividends are paid out during the life of the option.

The option pricing function under Black-Scholes model:

$$C(S,T) = SN(\frac{ln(\frac{S}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}) - Ke^{-r(T-t)}N(\frac{ln(\frac{S}{K}) + (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}) \tag{2}$$

*$N(x)$ here is standard normal distribution

Since the log stock price is assumed normal distributed, we can have:

$$ln(S_T) \sim N(lnS_t + (r - \frac{\sigma^2}{2})(T-t), \sigma\sqrt{T-t}) \tag{3}$$

$$q_T(S) = \frac{1}{\sqrt{2\pi}S_t\sigma\sqrt{T-t}} e^{(-\frac{(lnS_t-u)^2}{2\sigma^2(T-t)})} \tag{4}$$

*function(4) is the probability density function of S_t, which follows lognormal distribution

$$q(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{(-\frac{(lnx-\mu)^2}{2\sigma^2})} \tag{5}$$

The call option price is defined (under risk neutral measure):

$$C = e^{-r(T-t)}E[max(S_T - K), 0] \tag{6}$$

$$\begin{aligned} E[max(S_T - K), 0] &= \int_{-\infty}^{\infty} max(s-K, 0) q_T(s) ds \\ &= \int_{K}^{\infty} (s-K) q_T(s) ds \\ &= (s-K)\frac{1}{\sqrt{2\pi}s\sqrt{T-t}} e^{(-\frac{(lns-\mu)^2}{2\sigma^2})} ds \end{aligned} \tag{7}$$

*set u as lns, we shall have:

$$\begin{aligned} E[max(S_T - K), 0] = &\int_{lnK}^{\infty} \frac{e^u}{\sqrt{2\pi}\sigma\sqrt{T-t}} e^{-\frac{(u-\mu)^2}{2\sigma^2(T-t)}} du \\ &- \int_{lnK}^{\infty} \frac{K}{\sqrt{2\pi}\sigma\sqrt{T-t}} e^{-\frac{(u-\mu)^2}{2\sigma^2(T-t)}} du \end{aligned} \tag{8}$$

Then we bring back the μ to function(8)

$$E[max(S_T - K), 0] = E(S_T)N(\frac{lnE(S_T) - lnK + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}})$$
$$-KN(\frac{lnE(S_T) - lnK - \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}} \quad (9)$$

with $E(S_T) = Se^{r(T-t)}$, we can get function(2)

II Fourier Transform:

There are two fourier transform applications on optioon pricing. One is fourier transforming the PDF of log stock price and integrate it to get the CDF of the log stock price. The other one is doing fourier transform on the stock price and inverse the fourier transform to get characteristic function of option price.

1. Characteristic Function
Characteristic function is the fourier transform of probability density function.

$$\Phi_X(u) = E[e^{iuX}] = \int_{-\infty}^{\infty} e^{iuX} f_x(X)dx \quad (10)$$

$$e^{iuX} = cos(uX) + isin(uX) \quad (11)$$

From equation(11) we know the characteristic function is symmetric.($\bar{\Phi_X}(u) = \Phi_X(-u)$) This property is important when we introduced the factorization method of Fast Fourier Transform.

Fourier Transform can be seen as a vector on complex plane, which defined as: $z = a + ib$. The conjugate transpose of fourier transform is $\bar{z} = a - ib$. The real part is $R = (z + \bar{z})/2$, and imaginary part is $I = (z - \bar{z})/2$. Then we implement the characteristic function.

$$R[\Phi_X(u)] = \frac{1}{2}[\Phi_X(u) + \Phi_X(-u)]$$
$$I[\Phi_X(u)] = \frac{1}{2i}[\Phi_X(u) - \Phi_X(-u)] \quad (12)$$

Here we find the real part of characteristic function is always even, and imaginary part is always odd.

2. Inverse Fourier Transform on Characteristic Function
Similar to the B-S model, there are 4 assumptions when we use Inverse Fourier Transform to price options.

a. The market is risk neutral

b. No tax or other trading fee exist

c. Expected return rate is risk-free rate

d. No dividened in options

Call option price under risk neutral measure:

$$\begin{aligned} C &= e^{-rT} E^Q[(S_T - K)^+] \\ &= e^{-rT} \int_K^{\infty} (S_T - K) q_T(S) dS_T \end{aligned} \quad (13)$$

Since $E(S_T) = S_0 e^{rT}$, we know the cumulative distribution function $C_T(k)$is not integrable. We add a variable α here and we get:

$$c_T(k) = e^{\alpha k} C_T(k) \quad (14)$$

Now we get the Fourier Transform of $c_T(k)$

$$
\begin{aligned}
\Psi_T(u) &= \int_{-\infty}^{\infty} e^{iuk} c_T(k) dk \\
&= \int_{-\infty}^{\infty} e^{\alpha k} e^{-rT} (e^s - e^k) q_T(s) ds dk \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s) (\int_{-\infty}^{\infty} e^{\alpha k} e^{iuk} (e^s - e^k) dk) ds \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s) (e^s \int_{-\infty}^{s} e^{(\alpha+iu)k} dk - \int_{-\infty}^{s} e^{(\alpha+iu+1)k} dk) ds \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s) (\frac{e^s}{\alpha+iu} [e^{(\alpha+iu)k}]_{-\infty}^{s} - \frac{1}{\alpha+iu+1} [e^{(\alpha+iu+1)k}]_{-\infty}^{s}) ds \\
&= \int_{-\infty}^{\infty} e^{-rT} q_T(s) (\frac{e^{(\alpha+iu+1)s}}{\alpha+iu} - \frac{e^{(\alpha+iu+1)s}}{\alpha+iu+1}) ds \\
&= e^{-rT} \int_{-\infty}^{\infty} q_T(s) \frac{e^{(\alpha+iu+1)s}}{(\alpha+iu)(\alpha+iu+1)} ds \\
&= e^{-rT} \frac{\Phi_T(u - i(\alpha+1))}{(\alpha+iu)(\alpha+iu+1)} ds \qquad (15)
\end{aligned}
$$

$$
* \int_{-\infty}^{\infty} q_T(s) e^{(\alpha+iu+1)s} ds = \int_{-\infty}^{\infty} q_T(s) e^{i(u-i(\alpha+1))s} ds = \Phi_T(u - i(\alpha+1))
$$

Now we want to do inverse fourier transform on equation(15), based on the definition of fourier transform inversion:

$$
F^{-1}(\hat{f}) = f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\hat{u}) e^{iux} du \qquad (16)
$$

So, the call optin price obtained by:

$$
C_T(k) = \frac{e^{(-\alpha k)}}{2\pi} \int_{-\infty}^{\infty} e^{-iuk} \Phi_T(u) du \qquad (17)
$$

Since we know the real part of fourier transform is even, we can get:

$$
C_T(k) = \frac{e^{(-\alpha k)}}{\pi} \int_{0}^{\infty} R[e^{-iuk} \Phi_T(u)] du \qquad (18)
$$

The value of α is determined by $E^Q[S_T^{1+\alpha}] < \infty$

III Fast Fourier Transform

The fast fourier transform method is based on the fourier transform method we introduced in the previous section, but here we will use discrete fourier transform.
1. Standard discrete fourier transform:

$$
w_k = \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(k-1)} x(j) \quad for\ k = 1, \ldots, N \qquad (19)
$$

2. DFT pricing call options:

In order to use fast fourier transform, N must be even. Now we can transform function (19) to:

$$
C_T(k) \approx \frac{e^{-\alpha k}}{\pi} \sum_{j=1}^{N} e^{-iu_j k} \Phi_T(u_j) \eta \qquad (20)
$$

$N\eta$ is the upper limit of the integration, N is the number of segments of strike price, η is the interval of integration and λ is the regular spacing size of strike price. Then we have:

$$
\begin{aligned}
k_u &= -b + \lambda(u-1) \\
v_j &= (j-1)\eta \\
b &= \frac{N\lambda}{2} \qquad (21)
\end{aligned}
$$

Bring equation (22) back to function (21). Now we obtain the Call option price function under discrete fourier transform inverse

$$\begin{aligned}
C_T(k) &\approx \frac{e^{(-\alpha k_u)}}{\pi}\sum_{j=1}^{N} e^{-iv_j[-b+\lambda(u-1)}\Phi_T(v_j)\eta \\
&\approx \frac{e^{(-\alpha k_u)}}{\pi}\sum_{j=1}^{N} e^{-i\lambda\eta(j-1)(u-1)e^{ibv_j}}\Phi_T(v_j)\eta \\
&\approx \frac{e^{(-\alpha k_u)}}{\pi}\sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(u-1)e^{ibv_j}}\Phi_T(v_j)\eta \qquad (22)
\end{aligned}$$

$*\frac{2\pi}{N} = \lambda\eta$

3. FFT introduction:

First we introduce the equation $z^n = 1$, the solutions z are the "nth roots of unity", which means there are n evenly spaced points around the unit circle in the complex plane. (Introduction to Linear Algebra, Gilbert Strang) Then we consider the complex number $\omega = e^{i\theta}$ and its polar form $e^{i\theta} = cos(\theta) + isin(\theta)$. Now, for a n by n fourier matrix, it contains powers of $\omega = e^{\frac{2\pi i}{n}}$.

$$F_n c = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \dots & \dots & \dots & \ddots & \dots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{bmatrix}_{n*n} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \dots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \dots \\ y_{n-1} \end{bmatrix} = y \qquad (23)$$

From function(23), we know the fourier matrix times a vector c will take n^2 multiplications, since the fourier matrix has n^2 entries. To reduce the multiplication, we will do factorization on the fourier matrix because the symmetric property on ω. (Factorization will generate zero entries in a matrix, which can be skipped during multiplication) Here we will directly introduce the factorization method in FFT.

First we want to prove the symmetric property of ω on complex plane.

$$\begin{aligned}
\omega_n^k &= e^{\frac{2\pi k i}{n}} \\
\omega_{2n}^{2k} &= cos(2\pi\frac{2k}{2n}) + isin(2\pi\frac{2k}{2n}) \\
&= cos(2\pi\frac{k}{n}) + isin(2\pi\frac{k}{n}) = \omega_n^k \qquad (24)
\end{aligned}$$

And Elimination Lemma:

$$\begin{aligned}
\omega_n^{k+\frac{n}{2}} &= cos(2\pi\frac{k+\frac{n}{2}}{n}) + isin(2\pi\frac{k+\frac{n}{2}}{n}) \\
&= cos(2\pi\frac{k}{n}+\pi) + isin(2\pi\frac{k}{n}+\pi) \\
&= -cos(2\pi\frac{k}{n}) - isin(2\pi\frac{k}{n}) \\
&= -\omega_n^k \qquad (25)
\end{aligned}$$

Now, set we have a n terms polynomial function $A(x)$, and $n = 2^l$

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{n-1}x^{n-1} \qquad (26)$$

Then we serperate its even and odd parts

$$\begin{aligned}
A(x) &= (a_0 + a_2x^2 + a_4x^4 + \cdots + a_{n-2}x^{n-2}) + (a_1x + a_3x^3 + a_5x^5 + \cdots + a_{n-1}x^{n-1}) \\
&= (a_0 + a_2x^2 + a_4x^4 + \cdots + a_{n-2}x^{n-2}) + x(a_1 + a_3x^2 + a_5x^4 + \cdots + a_{n-1}x^{n-2} \qquad (27)
\end{aligned}$$

$$\begin{aligned}
A_1(x) &= (a_0 + a_2x + a_4x^2 + \cdots + a_{n-2}x^{\frac{n-2}{2}} \\
A_2(x) &= (a_1 + a_3x + a_5x^2 + \cdots + a_{n-2}x^{\frac{n-2}{2}} \\
A(x) &= A_1(x^2) + xA_2(x^2) \qquad (28)
\end{aligned}$$

For $0 \le k \le \frac{n}{2} - 1$

$$\begin{aligned} A(\omega_n^k) &= A_1(\omega_n^{2k}) + \omega_n^k A_2(\omega_n^{2k}) \\ A(\omega_n^k) &= A_1(\omega_{\frac{n}{2}}^k) + \omega_n^k A_2(\omega_{\frac{n}{2}}^k) \end{aligned} \tag{29}$$

For $\frac{n}{2} \le k + \frac{n}{2} \le n - 1$

$$\begin{aligned} A(\omega_n^{k+\frac{n}{2}}) &= A_1(\omega_n^{2k+n}) + \omega_n^{k+\frac{n}{2}} A_2(\omega_n^{2k+n}) \\ A(\omega_n^{k+\frac{n}{2}}) &= A_1(\omega_{\frac{n}{2}}^k) - \omega_n^k A_2(\omega_{\frac{n}{2}}^k) \end{aligned} \tag{30}$$

$**\omega_n^{2k+n} = \omega_n^{2k}\omega_n^n = \omega_n^{2k} = \omega_{\frac{n}{2}}^k, \quad \omega_n^{k+\frac{n}{2}} = -\omega_n^k$ Based on function (29) and (30), we find if we know the solution of $A_1(x)$ and $A_2(x)$ on $\omega_{\frac{n}{2}}^0, \omega_{\frac{n}{2}}^1, \omega_{\frac{n}{2}}^2, \ldots, \omega_{\frac{n}{2}}^{\frac{n}{2}-1}$, we can get the value of $A(x)$ in O(n) time. $A_1(x)$ and $A_2(x)$ here are half-size of orginial $A(x)$, so we can replicate this process multiple times to reduce the complexity of fourier matrix multiplication. For $n = 2^l$, we can reduce the time complexity from $O[N^2]$ to $O[N * log_2 N]$

IV Simpson's Rule

Definition:

$\int_a^b f(x)dx \approx$
$\frac{b-a}{3}(f(x_0) + 4[f(x_1) + f(x_3) + f(x_5) + \ldots] + 2[f(x_2) + f(x_4) + f(x_6) + \ldots] + f(x_n))$

*n must be even.

Explanation: This method is introduced to better approximate the integration compared with generally used Mid-Ordinate rule and Trapezium rule. The reason we define the simpson's rule function ourselves is the package offered by python will generate η automatically, in order to better invetigate FFT ability on pricing the optino, we want to choose η mannually.