Option Pricing via Fast Fourier Transform: Part 2

Goals:

- Review the relationship between Characteristic Functions and Fourier Transforms of European Payoffs.
- Demonstrate how we can use Fast Fourier Transform in order to compute the Pricing Integral.
- Discuss the Practicalities of Pricing Options via FFT.

Hirsa, Chapter 2 is a very good reference for this material, and this lecture will follow the book closely.

Review: Option Pricing via Fourier Transform

- In the last lecture, we showed that we can represent the prices of a set of European call options in terms of the Characteristic Function of the density of the log of the asset price.

- In particular, if we know the Characteristic Function is known, we can price a European call option using the following formula:

$$C_T(k) = \frac{e^{-\alpha k}}{\pi} \int_0^{+\infty} e^{-i\nu k} \Psi(\nu)\, d\nu, \tag{1}$$

where $\Psi(\nu)$ has been computed to be:

$$\Psi(\nu) = \frac{e^{-\int_0^T r_u\, du}}{(\alpha + i\nu)(\alpha + i\nu + 1)} \Phi(\nu - (\alpha + 1)i), \tag{2}$$

where $\Phi(u)$ is the characteristic function of $q(s)$, the density of the log of the asset price.

Review: Option Pricing via Fourier Transform

- At the crux of solving this pricing problem is applying **Fourier Inversion** to the known Characteristic Function.
- The pricing formula in (1) still requires us to take an integral to extract the option price for all strikes.
- In the last lecture, we showed how this could be done by applying quadrature methods.
- Today we will show you how to use **Fast Fourier Transform** to compute the remaining integral, and discuss the gains in computational efficiency that we get from using this approach.

Option Pricing via Fourier Transform: Comments on Payoff Functions

- Also note that the derivation that we worked through in the last class only applies to **European Call options**. In order to extend to additional payoffs we need to go through a similar derivation, and for some payoff functions there will be no solution.
- In general, the scope of this method is limited to European payoffs. The derivations for European Calls and Puts are widely documented and require no additional work.
- Extension to non-standard payoff functions may require additional derivations, or use of another method.
- In the next slide, we will show how we could apply this technique to price European Digital Options. To do this we follow the same procedure as we followed in the last lecture for European Calls.

Fourier Transform of Digital Option Price

- In the last lecture, we showed how to derive a pricing formula for a European call using Fourier Techniques. In order to make sure we understand this derivation, let's have a look at a slight modification: European Digital Put Options.
- A European Digital Pay pays 1 if the asset is less than the strike at expiry, and 0 otherwise. That is:

$$
\begin{aligned}
D_T(K) &= \tilde{\mathbb{E}}\left[e^{-\int_0^T r_u\,du} 1_{\{S_T<K\}}\right] && (3)\\
&= \int_{-\infty}^{K} f(S)\,dS && (4)\\
&= \int_{-\infty}^{k} q(s)\,ds && (5)
\end{aligned}
$$

In the final step we changed variables, such that

$$s = \log(S_T) \tag{6}$$

$$k = \log(K) \tag{7}$$

- As we did in the last lecture, let's define a modified digital price with a damping factor applied. Remember that this damping factor ensures that our inner integral converges.

$$\begin{aligned} \tilde{D}_T(k) &= e^{\alpha k} D_T(k) && (8) \\ &= e^{\alpha k} \int_{-\infty}^{k} q(s)\, ds && (9) \end{aligned}$$

Fourier Transform of Digital Option Price

- Next, let's calculate the Fourier Transform of the modified Digital Option Price.

$$\begin{aligned}
\Psi(\nu) &= \int_{-\infty}^{+\infty} e^{i\nu k} \tilde{D}_T(k) dk \\
&= e^{-\int_0^T r_u \, du} \int_{-\infty}^{+\infty} e^{(i\nu+\alpha)k} \left\{ \int_{-\infty}^{k} q(s) \, ds \right\} dk \\
&= e^{-\int_0^T r_u \, du} \int_{-\infty}^{+\infty} q(s) \left\{ \int_{s}^{\infty} e^{(i\nu+\alpha)k} \, dk \right\} ds
\end{aligned}$$

- If our damping factor, $\alpha < 0$, then the inner integral converges.
- Taking the inner integral, we are left with:

$$\Psi(\nu) = -\frac{e^{-\int_0^T r_u \, du}}{(i\nu + \alpha)} \int_{-\infty}^{+\infty} q(s) e^{(i\nu+\alpha)s} ds \qquad (10)$$

Fourier Transform of Digital Option Price

- Factoring out an i we see the remaining integral is just the characteristic function of $q(s)$ evaluated at $\nu + \alpha i$:

$$\Psi(\nu) = -\frac{e^{-\int_0^T r_u\,du}}{(i\nu+\alpha)} \int_{-\infty}^{+\infty} q(s) e^{i(\nu-\alpha i)s} ds \tag{11}$$

$$= -\frac{e^{-\int_0^T r_u\,du}}{(i\nu+\alpha)} \Phi(\nu - \alpha i) \tag{12}$$

- In order to extract the digital option price from the Fourier Transform we just derived, we can use Fourier Inversion:

$$D_T(k) = \frac{e^{-\alpha k}}{\pi} \int_0^{\infty} e^{-i\nu k} \Psi(\nu)\, d\nu \tag{13}$$

- Here, we again used the fact that we are only interested in the real part of $D_T(K)$ and the fact that $D_T(k)$ is an even function in its real part.

Fourier Transform of Digital Option Price: Conclusions

- As in the case of the European Call price via Fourier Transform, we were able to simplify the pricing problem into calculation of a single integral.
- The remaining integral in (13) that we are approximating is an **Inverse Fourier Transform**
- We can see that, even in the case of simple payoffs, a considerable amount of work goes into deriving the Fourier Transform Pricing Formula for non-standard payoffs.

Calculating the Final Pricing Integral

- Let's return to the problem of pricing a European call via Fourier Transform, and look at the final integral in more detail.
- In the case of a European Call, we have:

$$C_T(k) = \frac{e^{-\alpha k}}{\pi} \int_0^{+\infty} e^{-i\nu k} \Psi(\nu)\, d\nu, \tag{14}$$

where $\Psi(\nu)$ has been computed to be:

$$\Psi(\nu) = \frac{e^{-\int_0^T r_u\, du}}{(\alpha + i\nu)(\alpha + i\nu + 1)} \Phi(\nu - (\alpha + 1)i) \tag{15}$$

- This remaining integral in (14) is an **Inverse Fourier Transform**
- We could approximate this integral via quadrature, as we discussed in the last class, or via pre-built FFT algorithms.

Review: Calculating the Pricing Integral via Quadrature

- In order to compute the integral in (14), we need to choose an upper bound B for our integral, a quadrature rule and define a set of nodes and node weights.
- A natural choice is to divide our interval 0 to B into N equal intervals.
- The nodes would then be defined as $\nu_i = (j-1)\Delta\nu$ and the node spacing would be defined as: $\Delta\nu = \frac{B}{N}$
- Let's use the trapezoidal rule, which makes our approximation:

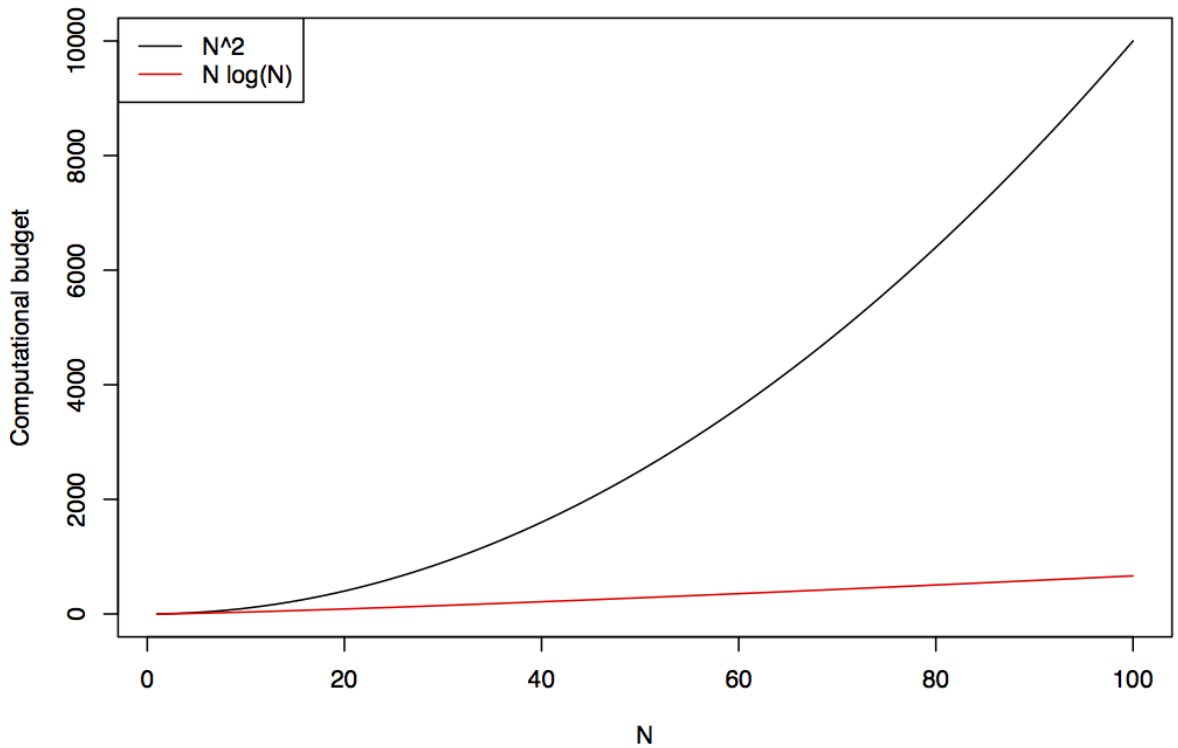
$$C_T(k) \approx \frac{e^{-\alpha k}}{\pi} \sum_{j=1}^{N} \frac{1}{2} \left[e^{-i\nu_j k} \Psi(\nu_j) + e^{-i\nu_{j+1} k} \Psi(\nu_{j+1}) \right] \Delta\nu \quad (16)$$

Calculating the Pricing Integral

- If we were to use this quadrature rule to approximate the Inverse Fourier integral, we could calculate the integral in N operations per strike.
- If we were pricing a single strike, then this method is reasonably efficient. However, in general, we are looking to generate prices for an entire volatility surface at the same time. (Why?)
- Let's say that instead of pricing a single option we wanted to price N options. The quadrature method would provide no benefit in pricing multiple options.
- As a result, pricing N options via quadrature would be $O(N^2)$.
- It turns out that pricing N options via Fast Fourier Transform is $O(N \log(N))$.

Computational Effort

A computational budget of $O(N\log(N))$ is significantly smaller than a budget of $O(N^2)$

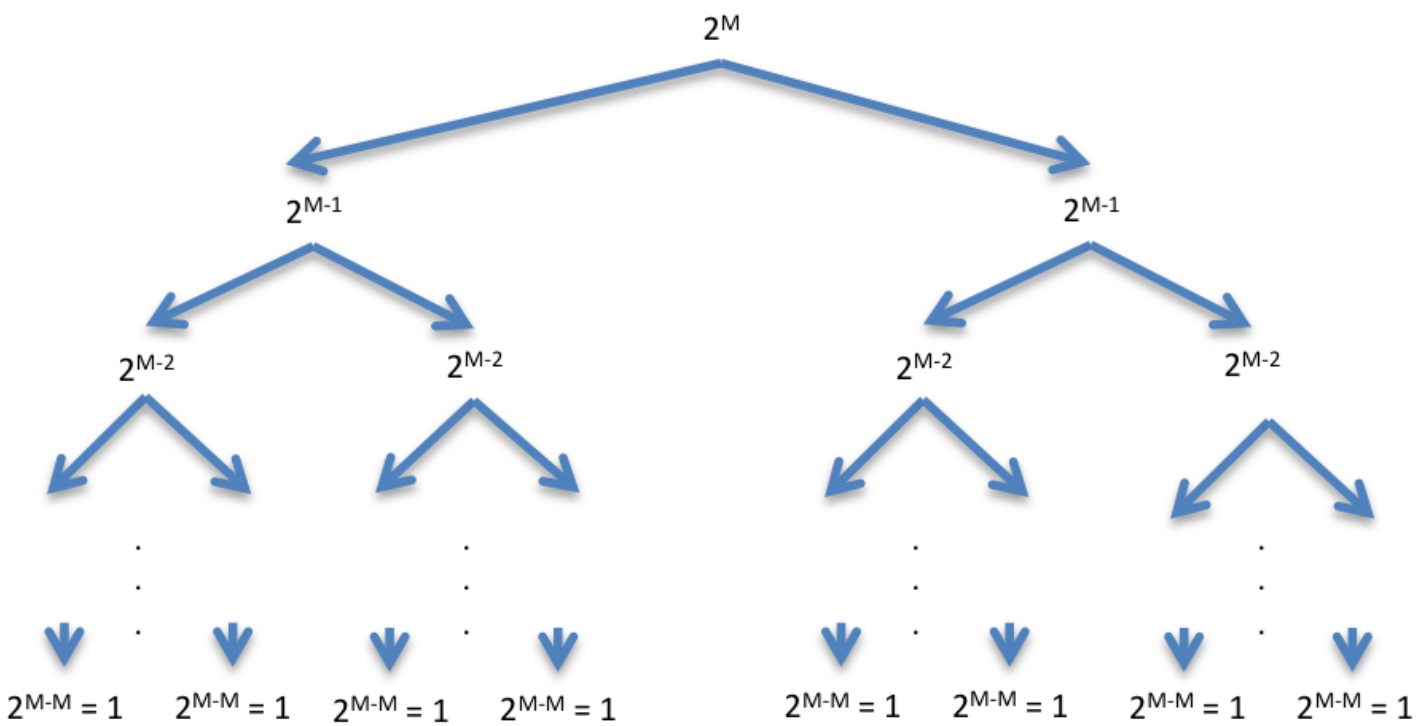


Fast Fourier Transform

- The Fast Fourier Transform (FFT) method allows us to evaluate the quadrature sum much more efficiently when we are evaluating multiple strikes.
- If we are pricing N options, it is able to do so with $N \log N$ total operations, or $\log N$ operations per strike.
- However, if we are evaluating a single strike, FFT will still require $N \log N$ operations.
- How does the FFT algorithm achieve this level of efficiency for multiple strikes?
  - The inner workings of the FFT algorithm are beyond the scope of this course (and most courses)
  - The main idea is to exploit the periodicity of the Fourier transform when evaluating the option price for multiple strikes

Fast Fourier Transform

- FFT algorithms use a "divide and conquer" approach to achieve high levels of efficiency



Fast Fourier Transform

- For our purposes it is perfectly acceptable to take the FFT algorithm as a given.
- Instead of worrying about the details of the algorithm, our job is to formulate our Fourier Inversion problem in a manner that is suitable for pre-built FFT algorithms.
- If we are able to do this, then we can take advantage of the superior efficiency of FFT algorithms to obtain prices for entire volatility surfaces with high accuracy.
- In the next few slides, we will go through how to make our problem suitable for FFT.

Calculating the Pricing Integral via Fast Fourier Transform

- Fast Fourier Transform allows us to solve problems of the following form:

$$\omega(m) = \sum_{j=1}^{N} e^{-i\frac{2\pi}{N}(j-1)(m-1)} x(j) \tag{17}$$

Where $m = 1 \ldots N$

- Recall that our Fourier Transform call pricing formula is:

$$\begin{aligned} C_T(k_m) &\approx \frac{e^{-\alpha k_m}}{\pi} \sum_{j=1}^{N} \frac{\Delta\nu}{2} \left[e^{-i\nu_j k_m} \Psi(\nu_j) + e^{-i\nu_{j+1} k_m} \Psi(\nu_{j+1}) \right] \\ &\approx \frac{e^{-\alpha k_m}}{\pi} \frac{\Delta\nu}{2} \left[F(\nu_1) + 2F(\nu_2) \ldots 2F(\nu_N) + F(\nu_{N+1}) \right], \end{aligned}$$

where in the last equation, $F(\nu_j) = e^{-i\nu_j k_m} \Psi(\nu_j)$

Calculating the Pricing Integral via Fast Fourier Transform

- Notice that the first and last terms appear once, and all other terms appear twice when we expand our sum.
- The last term, $e^{-i\nu_{N+1}k}\Psi(\nu_{N+1})$ approaches 0 exponentially and our upper bound B introduces little approximation error. As a result, we can omit this term from our approximation.
- We can then define a weight function $w_j = \frac{\Delta\nu}{2}(2 - \delta_{j-1})$, where δ_{j-1} is an indicator function equal to 1 if $j = 1$ and 0 otherwise.
- Making these substitutions, we have:

$$C_T(k_m) \approx \frac{e^{-\alpha k_m}}{\pi} \sum_{j=1}^{N} e^{-i\nu_j k_m} \Psi(\nu_j) w_j \tag{18}$$

Calculating the Pricing Integral via Fast Fourier Transform

- Next, let's define a grid of strikes that we will use in our Fourier Inversion:

$$k_m = \beta + (m-1)\Delta k \tag{19}$$

$$\beta = \ln S_0 - \frac{\Delta k N}{2} \tag{20}$$

- Note that this choice of a strike grid will ensure that the middle strike on the grid is equal to S_0, that is, it is the at-the-money spot strike.

- Additionally, remember that we have defined our nodes for ν_j as:

$$\nu_j = (j-1)\Delta\nu \tag{21}$$

$$\Delta\nu = \frac{B}{N} \tag{22}$$

- Plugging these values of k_m and ν_j into the sum on the right hand side of (18), we have:

$$C_T(k_m) \approx \frac{e^{-\alpha k_m}}{\pi} \sum_{j=1}^{N} e^{-i((j-1)\Delta\nu)(\beta+(m-1)\Delta k)} \Psi(\nu_j) w_j$$

Calculating the Pricing Integral via Fast Fourier Transform

- Simplifying and rearranging, we get:

$$C_T(k_m) \approx \frac{e^{-\alpha k_m}}{\pi} \sum_{j=1}^{N} e^{-i(j-1)(m-1)\Delta\nu\Delta k} e^{-i\nu_j\beta} \Psi(\nu_j) w_j$$

- This now looks very similar to the function in (17).
- In particular, we can see that both equations have a $-i(j-1)(m-1)$ in the first exponent.
- Further, if we set $\frac{2\pi}{N} = \Delta\nu\Delta k$ then the exponential terms are equal.
- Lastly, we can set $x(j) = e^{-i\nu_j\beta}\Psi(\nu_j)w_j$.
- After these substitutions, our problem now has the form of the original FFT equation (17).

Option Pricing via FFT

- Now that we have formulated our pricing problem in terms that FFT algorithms are able to solve, we can now use these algorithms to price N options in $O(N \log N)$.
- In particular, we have been able to setup our pricing integral, which computed the **Inverse Fourier Transform** as a call to a pre-built FFT package.

Option Pricing via FFT: Summary of Methodology

- At this point, we have derived all the results that we need in order to use FFT techniques to price European Options.
- Before moving on, let's quickly go through the steps that you should take in order to implement this technique on your own.

Option Pricing via FFT: Summary of Methodology

- We begin by choosing the parameters that correspond to the technique.
  - This means choosing N and $\Delta\nu$ and α.
  - This implicitly defines values for the rest of our model parameters, Δk and B.
  - Remember that for a call $\alpha > 0$ and for a put $\alpha < 0$.
- We should also choose the stochastic process that we want to use, and find its Characteristic Function.

Option Pricing via FFT: Summary of Methodology

- Next, we form a vector of inputs, x_i to our FFT algorithm:

$$x_j = \frac{[(2-\delta_{j-1})\Delta\nu]\, e^{-\int_0^T r_u\, du}}{2(\alpha+i\nu_j)(\alpha+i\nu_j+1)} e^{-i\left(\ln S_0 - \frac{\Delta k N}{2}\right)\nu_j} \Phi(\nu_j - (\alpha+1)i) \quad (23)$$

- Next, we call FFT with our input vector x and obtain an output vector y.

- Finally, we recover the call price

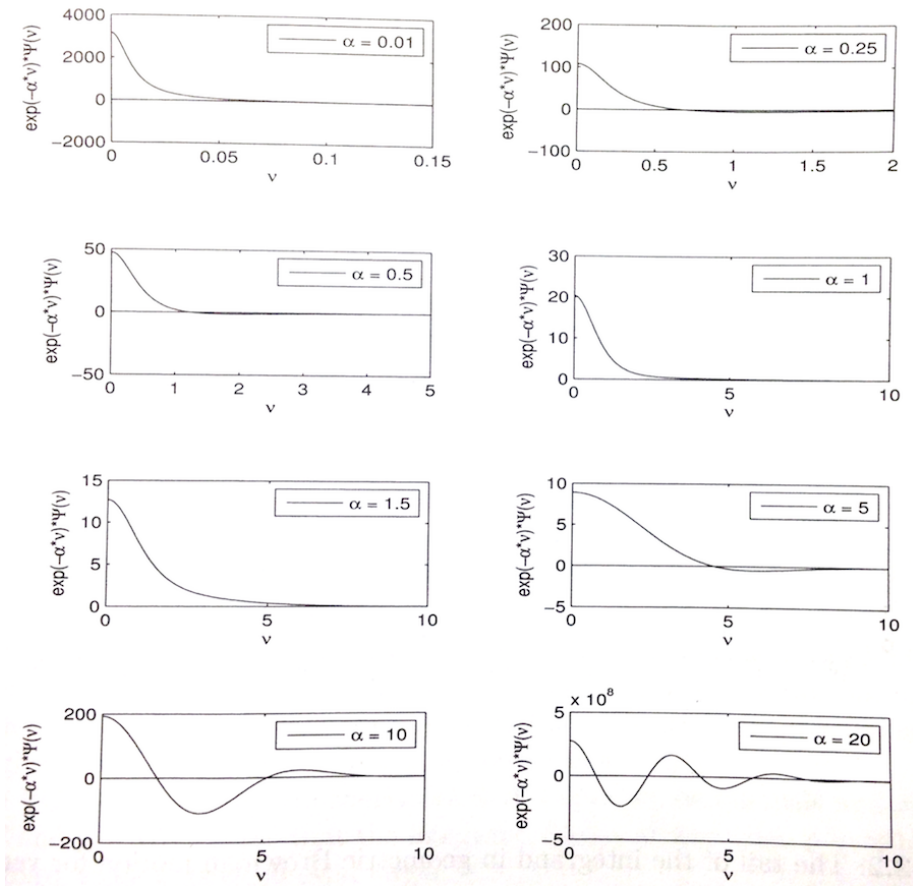
$$C_T(k_j) = \frac{e^{-\alpha\left[\ln S_0 - \Delta k\left(\frac{N}{2}-(j-1)\right)\right]}}{\pi} Re(y_j) \quad (24)$$

- This gives us a vector, $C_T(k)$ of option prices for our input strikes.

Dependence on α

- One of the model parameters in our FFT pricing technique is the damping factor, α that we introducted to ensure convergence of the inner integral.
- Ideally, we would want our choice of α to not have a significant impact on our model price.
- In reality, we find this not to be the case, and, in particular we can see that the "wrong" value for alpha will lead to significant pricing errors.
- This may be slightly counterintuitive as α is just a factor that we multiply by initially, and then remove at the end, but it is an important thing to keep in mind when implementing this technique.

Dependence on α



Plots of the functions $v \mapsto e^{-ivk}\Psi_\alpha(v)$ for different values of α. Here, s_T is normally distributed. Source: Hirsa, p. 45

Dependence on α

- In practice, this creates a challenge as we are generally working with complex stochastic process with unknown model prices and do not want our choice of α to bias our results.
- As discussed earlier in the class, an important tool for mitigating issues like this is to begin with a simplified model parameterization that leads to an analytical solution.
  - For example, you saw in your homework how to reduce the Heston model to the Black-Scholes model.
- Additionally, we can analyze our option price as a function of α and see what values of α tend to converge to the same solution.
- As you gain experience with FFT techniques, you will develop some intuition around reasonable values of α in different contexts.
- A good general rule is to use an $\alpha \approx 1$ in absolute terms.

Dependence on Additional FFT Parameters

- In addition to α, when we run our FFT algorithms the following additional parameters need to be specified:
  - $N, \quad B, \quad \Delta\nu, \quad \Delta k$
- However, we do not get to choose each of these freely.
- For example, choosing N and B defines the value for $\Delta\nu$
- Additionally, in order to keep the problem suitable for FFT we must have $\Delta k = \frac{2\pi}{\Delta\nu N}$. Therefore choosing N and $\Delta\nu$ defines Δk.
- In reality there are only two free parameters, which we typically either view as N and B or N and $\Delta\nu$.
- We also must have $N = 2^n$ in order to fit this problem into FFT algorithms.

Dependence on Additional FFT Parameters

- Choosing a higher value for N will mean we are using a tighter strike grid, and make the calculation more accurate for an entire volatility surface.
- Choosing a small value of N may be suitable if we are only interested in a single strike, and we construct our strike grid such that the desired strike is on the grid.
- We will want to adapt our values of B or $\Delta\nu$ as we modify the value of N.
  - For small N, a small value of B would lead to potential error from truncating the integral prematurely.
  - Conversely, for large N, a large value of B might result in wasted computational effort going to a region with little to no distributional mass.

Dependence on Additional FFT Parameters

- The most accurate, but also most computationally intensive results will come from high values of N combined with small $\Delta\nu$.
  - This choice of parameters will, however, calculate values for many strikes that are not realistic.
- In practice, choice of the best set of FFT parameters end up being somewhat subjective and depends on the problem you are trying to solve.

Strike Spacing Functions

- So far we have used the following strike grid in our FFT calculations:

$$k_m = \beta + (m-1)\Delta k \tag{25}$$

$$\beta = \ln S_0 - \frac{\Delta k N}{2} \tag{26}$$

- This specification ensures that the ATM strike is in our strike grid (Where?)
- If we use this grid, however, we cannot guarantee that any other strikes will appear exactly on our grid. To extract prices for these strikes, we will potentially need to rely on interpolation.
- In general, using linear interpolation is sufficient if our grid is dense enough.

Strike Spacing Functions

- Alternatively, we could simply adjust our strike spacing function to include the strike that we want to appear exactly on the grid.
- The simplest way would be to modify β to the following:

$$\beta = \ln K^* - \frac{\Delta k N}{2}, \tag{27}$$

where K^* is the strike that we would like to appear on our grid.

Implementation of Option Pricing via FFT

- An additional feature of the FFT pricing technique is that it is very general and can be used for many models and asset classes.
- In particular, there are hundreds of models with known Characteristic Functions that fit into our framework.
  - **Hirsa, Chapter 1** has a list of many useful models & their Characteristic Functions.
- Once we go through the upfront work of implementing the technique, we then have access to all of these models without having to change code internal to the technique.
- In particular, if the code is written correctly, we should only have to implement a single function which defines the characteristic function for our new stochastic process.

Option Pricing via FFT: Summary

- We can rewrite the price of a European option in terms of a Fourier transform
- Evaluating option prices in this setting reduces to the problem of inverting the Fourier transform
- We can quickly invert the Fourier transform using the method of Fast Fourier Transform
- This method exploits the periodicity of the complex exponential to reduce the computational effort necessary to invert the transform from $O(N^2)$ (naive quadrature) to $O(N \log N)$ (FFT)
- There are additional transform technique methods for Option Pricing, such as **Fractional FFT** and **COS Method**. These techniques are refinements of the FFT approach, and are discussed in detail in **Hirsa, Chapter 2**

Option Pricing via FFT: Strengths

- Provides a flexible, efficient method for pricing European options with a wide array of models.
- Prices entire volatility skews with the same effort required to produce single option price.
- Naturally lends itself to a general pricing framework with models easily substituted for each other.

Option Pricing via FFT: Weaknesses

- Difficult to extend to non-standard payoffs.
- Impossible to extend to path dependent options.
- Model prices are sensitive to parameters, α, N and B.
- Unable to accurately price highly out-of-the-money options
  - For a potential remedy to this, one can use the Saddlepoint method described in **Hirsa, Chapter 2**.