A Novel Fourier Transform B-spline Method for Option Pricing*


Gareth G. Haslip, FIA PhD
Cass Business School, City University London

October 23rd 2013
Financial Engineering Workshop Series, Cass Business School

* Research conducted jointly with Vladimir K. Kaishev, Professor, Cass Business School, City University London
Executive Summary

- Efficient methods for pricing European options are an important requirement for the risk-neutral calibration of asset models to the observed implied volatility surface ("inverse problem")

- An effective option pricing method for calibration purposes should exhibit two key features:
  1. Fast calculation time across many strike prices and maturities
  2. Robustness and accuracy across the entire parameter space, moneyness, and maturities

- Our option pricing framework is called the **Fourier Transform B-spline Method (FTBS)**
  - **First application** of B-spline interpolation theory in the field of derivative pricing
  - Provides **accurate and extremely efficient** closed-form representation of the European option price under an inverse Fourier transform
  - Ideally suited to the calibration of asset price models as it provides fast and accurate pricing of options across multiple strike prices and the parameter space

- We compare the FTBS method with five other state-of-the-art option pricing methods (FFT, FRFT, IAC, COS, and CONV), which we have diligently implemented to the best of our ability in C++ using identical hardware

- FTBS method is the preferable approximation method, relative to the other methods considered, for computing option prices across multiple strikes prices for the VG process, and for the Heston model and KoBoL (CGMY) model, for precision up to the level of 1E-6
Assumptions and Asset Models Applicable to the FTBS Method

Notation and Model Assumptions

- Let \((X_t)_{t \geq 0}\) with \(X_0 = 0\) be defined on a continuous-time probability space \((\Omega, \mathcal{F}, Q)\).
- Asset price \((S_t)_{t \geq 0}\) evolves according to an exponential stochastic process: 
  \[ S_t = S_0 e^{(r-q)t + X_t} \]
  where \(r\) is the risk-free interest rate and \(q\) is the continuous income yield provided by the asset.
- Assume \(X_t\) is a semimartingale process with respect to the filtration \(\mathcal{F}\) and \(Q\) is the risk-neutral measure under the assumption of no arbitrage so that 
  \(e^{-rt} S_t\) is a martingale.
- Require closed-form expression for characteristic function 
  \(\phi_{X_t}(z) \equiv E(e^{izX_t}), z \in \mathbb{C}\).

Asset Models

- Wide range of asset models considered:

<table>
<thead>
<tr>
<th>(Jump) Diffusion Models</th>
<th>Pure Jump Lévy Processes</th>
<th>Affine Processes</th>
</tr>
</thead>
</table>
The FTBS Pricing Method for European-Style Options

- Starting point is the Fourier transform based pricing integral in the complex domain
- We utilise the Lewis-Lipton formula provided by Lewis (2001) and Lipton (2002)
- This provides a semi-closed form pricing formula for any European-style option

\[ C(T, K) = S_0 e^{-qT} - \frac{\sqrt{S_0 K e^{-2(r+q)T}}}{\pi} \int_0^\infty \text{Re} \left[ e^{iu} \phi_{X_T} \left( u - \frac{i}{2} \right) \right] \frac{du}{u^2 + \frac{1}{4}} \]  

(1)

where \( K \) is the strike price, \( T \) is the expiry time and \( k = \log \left( \frac{S_0}{K} \right) + (r - q)T \)

- This holds provided the MGF of \( X_t \) exists for all \( v \in (\alpha, \beta) \) with \( \alpha < \frac{1}{2} \) and \( \beta > 1 \)

- Using (1) the price of a European call option can be evaluated using numerical integration
  - However, this is relatively slow and is not suitable for real-time option pricing applications
  - Also integral must be truncated at suitably large number which introduces truncation error

- Note that the method is also applicable for alternatives for the contour of integration parameter
  - For simplicity of the presentation, and ease of use of the method, we have chosen to use the above form with \( v = 0.5 \)
  - This works reasonably well for a wide choice of models and parameters
  - Note that we do not consider pricing deep out-of-the-money options in this paper
The following simplified version of (1) provides the foundations for the FTBS method:

\[ C(T, K) = S_0 e^{-qT} - \frac{1}{\pi} \sqrt{S_0 K e^{-\frac{(r+q)T}{2}}} I(k) \]  

where

\[ I(k) = \int_0^1 \cos \left( \frac{1-t}{t} k \right) s_1(t) dt \]

\[ + \int_0^1 \sin \left( \frac{1-t}{t} k \right) s_2(t) dt \]

and

\[ s_1(t) = \frac{\text{Re} \left[ \phi_X \left( \frac{1-t}{t} - \frac{i}{2} \right) \right]}{1 - 2t + \frac{5}{4} t^2} \]

\[ s_2(t) = -\frac{\text{Im} \left[ \phi_X \left( \frac{1-t}{t} - \frac{i}{2} \right) \right]}{1 - 2t + \frac{5}{4} t^2} \]

**Trigonometric Function**
- Dependent on parameter \( k \)
- Recall \( k = \log \left( \frac{S_0}{K} \right) + (r - q)T \)
- Therefore dependent on strike price and time to maturity

**Characteristic Function**
- \( s_1(t) \) and \( s_2(t) \) are completely independent of option contract features contained in \( k \)
- Only dependent on choice of log-return process through the characteristic function

Integration is now carried out over the unit interval which avoids truncation error in (1)

Separation of the integral trigonometric component dependent on strike price and components which are independent of the strike price is fundamental to the FTBS method

We now use optimal B-spline interpolation to approximate \( s_1(t) \) and \( s_2(t) \)
Splines, B-splines and Divided Differences

- The function \( s(t) \) is sampled at \( \nu \) interpolation sites \( \tau = \{\tau_1, \ldots, \tau_{\nu}\} \) with \( \tau_1 = 0 \), \( \tau_{\nu} = 1 \) and \( \tau_i < \tau_{i+1}, \ i = 1, \ldots, \nu \)

- **Spline function approximation:** Let \( n, l \in \mathbb{Z}^+ \) and partition the interval \([0, 1] \in \mathbb{R} \) by the points \( 0 = t_n < \ldots < t_{n+l} < t_{n+l+1} = 1 \) called knots

- Define spline function \( \tilde{s}(t) \) of order \( n \), degree \( n - 1 \) as a piece-wise polynomial function that coincides with a polynomial of degree \( n - 1 \) between the knots \( t_n, \ldots, t_{n+l+1} \)

- The polynomial pieces are smoothly joined at the knots \( t_{n+1}, \ldots, t_{n+l} \) so that the spline is \( n-2 \) times continuously differentiable on the interval \([0, 1] \)
To introduce the set of B-spline basis functions, we add $n - 1$ additional knots at each end of the interval $[0, 1]$:

\[
\begin{array}{cccccccc}
\tau_1 &=& 0 & \tau_2 & \tau_3 & \tau_4 & \cdots & \tau_{\nu-3} & \tau_{\nu-2} & \tau_{\nu-1} & 1 = \tau_\nu \\
t_1 &=& \cdots &=& t_n &=& 0 & t_{n+1} & t_{n+2} & \cdots & t_{n+l-1} & t_{n+l} & 1 = t_{n+l+1} = \cdots = t_{2n+l}
\end{array}
\]

- Given the interpolation sites $\mathbf{\tau} = \{\tau_1, \ldots, \tau_\nu\}$, the optimal choice of internal knots $\{t_i\}_{i=n+1}^{n+l}$ is provided by De Boor (2001) as being approximately the averages of the interpolation sites:

\[
t_{n+i} = \frac{\tau_{i+1} + \cdots + \tau_{i+n-1}}{n-1}, \quad i = 1, \ldots, l, \quad (l = \nu - n)
\]

- We have tested this choice numerically and it performs as well as the theoretical optimum.
Splines, B-splines and Divided Differences

- Polynomial splines form on \( \{t_i\}_{i=1}^{2n+l} \) form a linear space of functions, an element of which is represented as:

\[
\tilde{s}(t) = \sum_{i=1}^{p} c_i M_{i,n}(t)
\]  

(3)

where \( c_i \) are constant coefficients, \( p = l + n \) and \( M_{i,n}(t) \) are B-spline basis functions

- The B-spline basis functions \( M_{i,n}(t) \) are defined on \( \{t_i\}_{i=1}^{2n+l} \) as an nth order divided difference

\[
M_{i,n}(t) = M_{i,n}(t; t_i, \ldots, t_{i+n}) = [t_i, \ldots, t_{i+n}] f(y)
\]

where \( f(y) = n \left( \max \{(y - t), 0\} \right)^{n-1} \)

- Notation \([t_i, \ldots, t_{i+n}] f(y)\) denotes the nth order divided difference calculated recursively as:

\[
[t_i, \ldots, t_{i+n}] f(t) = \frac{[t_{i+1}, \ldots, t_{i+n}] f(t) - [t_i, \ldots, t_{i+n-1}] f(t)}{t_{i+n} - t_i}
\]

where \([t_i] f(t) = f(t_i)\) and \( \{t_j\}_{j=i}^{i+n} \) are pairwise distinct (alternative formula applies otherwise)
Splines, B-splines and Divided Differences

- B-spline basis functions have a very useful properties, in particular the Peano representation of a divided difference, which is an essential tool in our methodology for computing integrals:

\[
[t_i, \ldots, t_{i+n}] f(x) = \int_{\mathbb{R}} M_{i,n}(t; t_i, \ldots, t_{i+n}) \frac{f^{(n)}(t)}{n!} dt
\]  

(4)

- Using the Peano representation we will be able to compute \( I(k) \) numerically in closed-form by replacing \( s_1(t) \) and \( s_2(t) \) by their B-spline approximants \( \tilde{s}_1(t) \) and \( \tilde{s}_2(t) \)

- We will see that the Peano representation enables us to obtain efficient and accurate approximations to option prices

- Use quadratic splines \((n = 3)\) which are known to often produce better fits than cubic splines
Overview

1. Approximate: \( \tilde{s}_1(t) = \sum_{i=1}^{p_1} c_{1,i} M_{i,3}(t) \) and \( \tilde{s}_2(t) = \sum_{i=1}^{p_2} c_{2,i} M_{i,3}(t) \)

2. Substitute: 
   \[
   \tilde{I}(k) = \int_0^1 \cos \left( \frac{1-t}{t} k \right) \tilde{s}_1(t) \, dt + \int_0^1 \sin \left( \frac{1-t}{t} k \right) \tilde{s}_2(t) \, dt
   \] (5)

3. Apply Peano’s representation (4) to compute the integrals in closed-form:

   \[
   \int_0^1 \cos \left( \frac{1-t}{t} k \right) \tilde{s}_1(t) \, dt = \int_0^1 \cos \left( \frac{1-t}{t} k \right) \sum_{i=1}^{p_1} c_{1,i} M_{i,3}(t) \, dt = \sum_{i=1}^{p_1} c_{1,i} \int_0^1 \cos \left( \frac{1-t}{t} k \right) M_{i,3}(t) \, dt
   \]

   Peano’s formula: 
   \[
   [t_{1,i}, \ldots, t_{1,i+3}] f_1(t) = \int_0^1 \frac{f_1^{(3)}(t)}{3!} M_{i,3}(t) \, dt \implies \text{Integrate three times}
   \]

   \[
   \therefore f_1(t) = \frac{1}{12} \left\{ t \left( -(k^2 - 2t^2) \cos \frac{k(t-1)}{t} - 5kt \sin \frac{k(t-1)}{t} \right) 
   + k \, Ci \left( \frac{k}{t} \right) (-6kt \cos k + (k^2 - 6t^2) \sin k) - k \, Si \left( \frac{k}{t} \right) (6kt \sin k + (k^2 - 6t^2) \cos k) \right\}
   \]

   \[
   \int_0^1 \cos \left( \frac{1-t}{t} k \right) \tilde{s}_1(t) \, dt = 6 \sum_{i=1}^{p_1} c_{1,i} [t_{1,i}, \ldots, t_{1,i+3}] f_1(t)
   \]
FTBS Pricing Formula for European Options (Main Result)

- Let \( \{t_{1,i}\}_{i=1}^{6+l_1}, \{t_{2,j}\}_{j=1}^{6+l_2} \) and \( \{c_{1,i}\}_{i=1}^{p_1}, \{c_{2,j}\}_{j=1}^{p_2} \) be the sets of knots and linear coefficients of quadratic spline interpolants, \( \tilde{s}_1(t) \) and \( \tilde{s}_2(t) \) where \( p_1 = l_1 + 3 \) and \( p_2 = l_2 + 3 \) then

\[
C'(T, K) \approx \tilde{C}'(T, K) = S_0 e^{-qT} - \frac{\sqrt{S_0 K e^{-\frac{(r+q)}{2}}} \tilde{I}(k)}{\pi}
\]

(6)

where for \( k \neq 0 \), \( \tilde{I}(k) = 6 \sum_{i=1}^{p_1} c_{1,i} [t_{1,i}, t_{1,i+1}, t_{1,i+2}, t_{1,i+3}] f_1(t, k) \)

\[
+ 6 \sum_{i=1}^{p_2} c_{2,i} [t_{2,i}, t_{2,i+1}, t_{2,i+2}, t_{2,i+3}] f_2(t, k)
\]

and if \( k = 0 \) this simplifies to \( \tilde{I}(k) = \sum_{i=1}^{p_1} c_{1,i} + \sum_{i=1}^{p_2} c_{2,i} \)

- In practice, we can use the same interpolation sites and knots for both integrals, since functions \( s_1(t) \) and \( s_2(t) \) are based on the real and imaginary parts of the same characteristic function.

- A similar result is provided in the paper for alternative choices of the contour parameter \( v \)
FTBS Error Bound for European Options (Main Result)

- The absolute error of the FTBS European option price $\tilde{C}(T, K)$ is bounded by

$$|C(T, K) - \tilde{C}(T, K)| \leq \frac{\sqrt{S_0 K}}{\pi} e^{\frac{(r+q)T}{2}} \left( \tilde{C}_1 \left\| s_1^{(3)}(t) \right\| + \tilde{C}_2 \left\| s_2^{(3)}(t) \right\| \right) \quad (7)$$

where:

$$\tilde{C}_1 = C_1 \int_0^1 |\cos \left( \frac{1-t}{t} k \right)| \, dt$$

$$\tilde{C}_2 = C_2 \int_0^1 |\sin \left( \frac{1-t}{t} k \right)| \, dt$$

$C_j^i$ are the constants obtained from the optimal spline interpolation $\tilde{s}_j(t)$ of $s_j(t)$ ($j = 1, 2$) following Gaffney and Powell (1976)

and $\|f\| = \max\{|f(t)| : 0 \leq t \leq 1\}$

- The bound in (7) converges to zero as the mesh sizes $|\tilde{t}_j| = \max_i \left( |\tilde{t}_{j,i} - \tilde{t}_{j,i-1}| \right)$ go to zero at a rate $O \left( |\tilde{t}|^3 \right)$, where $\tilde{t} = \max \left( |\tilde{t}_1|, |\tilde{t}_2| \right)$, i.e.

$$\left| C(T, K) - \tilde{C}(T, K) \right| = O \left( |\tilde{t}|^3 \right)$$

- Therefore, by increasing the number of data sites used in the interpolation, any level of accuracy can be obtained for the option price.
Computational Aspects of FTBS Method

- Extremely efficient pricing method across the quantum of strike prices:
  - Coefficients $c_{1,i}, c_{2,i}$ depend only on the choice of stochastic model and time to maturity (they are only computed once at the first strike price in the list of strike prices considered)
  - Calculation of the divided differences of $f_1, f_2$ is independent of model choice and they are pre-computed and stored in a data file at a fine resolution over all possible values of $k$
  - Computing $\tilde{I}(k)$ simply requires evaluating two sum products over (say) 50 values
- Important advantage of the FTBS method is the elimination of truncation error which comes from the necessity to truncate the limit at infinity in the original expression (1)
  - This means that it is no longer necessary to carefully identify a truncation point for each asset price model by considering how quickly the characteristic function decays to zero
- Location of interpolation sites is important to extract information from functions $s_1(t)$ and $s_2(t)$ as efficiently as possible
  - Uniform interpolation sites ensures full model and parameter independence
  - However, greater efficiency is achieved by targeting regions where functions have greater curvature
  - In practice the allocation 60% to [0, 0.2], 20% to [0.2, 0.6] and 20% to [0.6, 1] worked effectively for a large range of models
Pre-computation of the divided differences for the chosen interpolation sites is recommend to maximise efficiency. Recall: \( k = \log \left( \frac{S_0}{K} \right) + (r - q)T \)

Consider the parameter ranges: \( \log \left( \frac{S_0}{K} \right) \in [-1, 2], r - q \in [0\%, 10\%], \) and \( T \in [0, 10] \)

- This implies \( k \in [k_{\min}, k_{\max}] \) where \( k_{\min} = -1 \) and \( k_{\max} = 3 \)
- Divide \([k_{\min}, k_{\max}]\) uniformly with \( N \) points with width \( \delta = \frac{k_{\max} - k_{\min}}{N - 1} \)
- Pre-compute the divided differences:
  \[
  DvD^1_i(k_m) = [t_{1,i}, t_{1,i+1}, t_{1,i+2}, t_{1,i+3}] f_1(t, k_m)
  \]
  \[
  DvD^2_j(k_m) = [t_{2,i}, t_{2,j+1}, t_{2,j+2}, t_{2,j+3}] f_2(t, k_m)
  \]
  for \( i = 1, \ldots, p_1, j = 1, \ldots, p_2, \) and \( m = 1, \ldots, N \)
  where \( k_m = k_{\min} + (m - 1)\delta \)

- E.g. for \( \delta = 1E-5 \) and 100 interpolation sites, less than 1GB storage is required to store the divided differences for all possible combinations of contract parameters

- Alternatively, for calibration applications, the required divided differences can be evaluated at the beginning of the calibration process which takes just one to two milliseconds
Numerical Results of FTBS Method

- Performance and numerical accuracy assessed relative to state-of-the-art existing methods
- Our main numerical comparison considers the computation time, for a specified level of precision, of pricing European options across multiple strikes prices (31 options)
- As a second comparison, we consider the efficiency of the FTBS method for pricing an European option at a single strike price
- A like-for-like comparison is provided for all comparison methods – we have diligently implemented each pricing method to the best of our ability in C++, using identical hardware
  - This enables an completely objective comparison of the FTBS method’s performance

<table>
<thead>
<tr>
<th>Comparison Pricing Methods</th>
<th>Asset Price Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiple Strikes</td>
</tr>
<tr>
<td>Fast Fourier Transform of Carr and Madan (1999)</td>
<td>VG, Heston, KoBoL (CGMY)</td>
</tr>
<tr>
<td>Fractional Fast Fourier Transform (FRFT) of Chourdakis (2005)</td>
<td>-</td>
</tr>
<tr>
<td>Cosine (COS) method of Fang and Oosterlee (2008)</td>
<td>-</td>
</tr>
<tr>
<td>Convolution (CONV) method of Lord et al. (2009)</td>
<td>-</td>
</tr>
<tr>
<td>Integration-along-cut and Inverse FFT of Levendorskii and Xie (2012)</td>
<td>-</td>
</tr>
</tbody>
</table>

- Finally, to validate the robustness of the FTBS method across options of all tenors and moneyness and across the entire parameter space, an inverse calibration problem is tested
FTBS method dominates all comparison pricing methods across range of asset price models and parameter sets at precision level 1E-5 (maximum absolute error across all option prices).

Calculation times for FTBS method very competitive:
- < 1 microsecond per option for VG process
- < 2 microsecond per option for Heston model
- < 3 microsecond per option for KoBoL (CGMY) model

<table>
<thead>
<tr>
<th>Model</th>
<th>Pricing Method</th>
<th>N</th>
<th>Low Abs. Error</th>
<th>Time (ms)</th>
<th>Speed up vs. FFT</th>
<th>Bench Abs. Error</th>
<th>Time (ms)</th>
<th>Speed up vs. FFT</th>
<th>High Abs. Error</th>
<th>Time (ms)</th>
<th>Speed up vs. FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VG</td>
<td>FFT</td>
<td>11</td>
<td>1.1E-05</td>
<td>1.23</td>
<td>-</td>
<td>12</td>
<td>4.4E-06</td>
<td>2.54</td>
<td>12</td>
<td>4.8E-06</td>
<td>2.50</td>
</tr>
<tr>
<td></td>
<td>FRFT</td>
<td>8</td>
<td>8.8E-06</td>
<td>0.25</td>
<td>5</td>
<td>6</td>
<td>1.7E-05</td>
<td>0.07</td>
<td>38</td>
<td>4.8E-06</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>COS</td>
<td>50</td>
<td>2.7E-05</td>
<td>0.14</td>
<td>9</td>
<td>60</td>
<td>9.1E-06</td>
<td>0.16</td>
<td>16</td>
<td>9.7E-06</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>FTBS</td>
<td>32</td>
<td>9.9E-06</td>
<td>0.02</td>
<td>61</td>
<td>31</td>
<td>5.0E-06</td>
<td>0.02</td>
<td>128</td>
<td>7.4E-06</td>
<td>0.02</td>
</tr>
<tr>
<td>Heston</td>
<td>FFT</td>
<td>11</td>
<td>1.0E-05</td>
<td>2.86</td>
<td>-</td>
<td>10</td>
<td>1.1E-05</td>
<td>1.48</td>
<td>-</td>
<td>9</td>
<td>1.8E-05</td>
</tr>
<tr>
<td></td>
<td>FRFT</td>
<td>8</td>
<td>3.0E-06</td>
<td>0.46</td>
<td>6</td>
<td>6</td>
<td>1.0E-05</td>
<td>0.12</td>
<td>12</td>
<td>5</td>
<td>1.8E-05</td>
</tr>
<tr>
<td></td>
<td>COS</td>
<td>22</td>
<td>1.3E-05</td>
<td>0.08</td>
<td>37</td>
<td>24</td>
<td>1.7E-05</td>
<td>0.08</td>
<td>18</td>
<td>2.0E-05</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>FTBS</td>
<td>44</td>
<td>1.1E-05</td>
<td>0.06</td>
<td>45</td>
<td>22</td>
<td>1.1E-05</td>
<td>0.03</td>
<td>45</td>
<td>1.8E-05</td>
<td>0.05</td>
</tr>
<tr>
<td>KoBoL (CGMY)</td>
<td>FFT</td>
<td>9</td>
<td>1.5E-05</td>
<td>0.51</td>
<td>-</td>
<td>9</td>
<td>1.2E-05</td>
<td>0.50</td>
<td>-</td>
<td>10</td>
<td>2.2E-05</td>
</tr>
<tr>
<td></td>
<td>FRFT</td>
<td>6</td>
<td>1.8E-05</td>
<td>0.09</td>
<td>5</td>
<td>5</td>
<td>1.7E-05</td>
<td>0.05</td>
<td>9</td>
<td>5</td>
<td>8.4E-06</td>
</tr>
<tr>
<td></td>
<td>COS</td>
<td>129</td>
<td>9.7E-06</td>
<td>0.40</td>
<td>1</td>
<td>54</td>
<td>1.1E-05</td>
<td>0.17</td>
<td>3</td>
<td>15</td>
<td>8.3E-06</td>
</tr>
<tr>
<td></td>
<td>FTBS</td>
<td>67</td>
<td>9.6E-06</td>
<td>0.07</td>
<td>8</td>
<td>30</td>
<td>1.1E-05</td>
<td>0.03</td>
<td>16</td>
<td>4.2E-06</td>
<td>0.03</td>
</tr>
</tbody>
</table>
Like-for-like Comparisons (Multiple Strike Prices):
Parameter Sets “Low”, “Bench”, and “High”. Calculation time reported for pricing 31 options.
Like-for-like Comparisons (Single Strike Price):

- Comparison of precision and computation times for pricing a single European option under the VG process
  - Outperforms all comparator methods except IAC at all levels of precision considered
Risk-Neutral Calibration

- Inverse problem to identify risk-neutral model parameters from cross-sectional option price data
- Provides an effective test of robustness across options of all tenors and moneyness, and across the entire parameter space of the asset model
  - Comparison to reference calibrations of Schoutens et al. (2005) using 144 European call option prices with maturities ranging from less than one month to 5.16 years
- Calibrations using FFT and FRFT configurations of Chourdakis (2005)
  - FFT and FRFT methods demonstrated instability for Bates model over full parameter space
- Interesting to note Bates model offers two sets of parameters with very different characteristics
Conclusion

- Fourier Transform B-spline pricing method for European options has been demonstrated to provide superior precision / computation times to state-of-the-art comparison methods.

- Efficiency of FTBS method is explained by a combination of factors:
  - Optimal spline interpolation is able to reproduce arbitrary functions with a high degree of precision using only a modest number of interpolation sites (e.g. 50).
  - The Peano representation of a divided difference enables the pricing integral to be evaluated in closed-form as a linear combination of spline coefficients and divided differences.
  - The spline interpolation is independent of strike price and hence only is carried out a single time for the first strike price considered (when pricing over a quantum of strike prices).
  - The divided differences are model independent and can be pre-computed.

- These factors combined mean that computing European option prices across the quantum of strike prices becomes a three stage process:
  1. Load pre-computed divided differences into memory (e.g. at beginning of the day).
  2. Perform spline interpolation to functions $s_1(t)$ and $s_2(t)$ to obtain spline coefficients.
  3. Compute each option price as a sum product of spline coefficients and divided differences – requires just $4p$ addition and multiplication operations to compute $I(k)$. 
References

- Chourdakis, K. Option pricing using the fractional FFT. Journal of Computational Finance, 8 (2005), 1–18.
References