Politecnico di Milano
Mathematical Engineering
Financial Engineering

FFT and Fractional-FFT Monte Carlo for Additive Processes

June 14, 2022

Chiara Agostini
Davide Cestaro
Daniel Sima

1 Introduction

In this project, we implement a fast Monte Carlo (MC) simulation technique for additive processes following the work of Azzone and Baviera (2021b). We present an application of this simulation algorithm to a specific subfamily of the additive normal tempered stable processes (ATS), which presents a power-law scaling of the time-dependent parameters. As outlined by the work of Azzone and Baviera (2021a), the additive processes are becoming the new frontier in equity derivatives. These processes accurately replicate market-implied volatility term structures while still maintaining the parsimony of the more elementary Levy normal tempered stable processes. In contrast to the latter class of processes, ATS processes reject the *a priori* hypothesis of stationary increments which is unfeasible from a financial point of view but still consider the independency on the increments. Moreover, it has been shown that there is statistical evidence that the power-law scaling parameters are observed in the market data and therefore our numerical implementations are focused on this sub case of ATS (see e.g., Azzone and Baviera 2021a).
The rest of the report is organized as follows. In section 2, we overview the MC simulation scheme for additive processes. In section 3, we present an application of the algorithm to the European call option pricing problem and discuss the numerical result with respect to the analytic pricing method. Section 4 presents an alternative implementation of the MC simulation technique, which exploits the Fractional Fourier transform. We conclude with section 5, where we show an application of a discretely-monitoring path-dependent option.

2 Lewis-FFT algorithm

Apart from providing an accurate calibration to the implied volatility surface of equity derivatives, the pricing models based on additive processes allow an accurate and fast numerical MC scheme for path-dependent option valuation (see e.g., Azzone and Baviera 2021b). The Lewis-FFT algorithm is based on the sampling of the ATS process increments from its cumulative distribution function (CDF) $P(x)$ and it is composed of two main steps. Firstly, we compute numerically the CDF $\hat{P}(x)$ in the Fourier space by exploiting the fast Fourier transform (FFT) scheme. Secondly, knowing the CDF approximation $\hat{P}(x)$, sample from a uniform distribution in $[0,1]$ a r.v. U, and via an interpolation method, we invert the CDF to get a sample of the ATS process increments, i.e. set $X = \hat{P}^{-1}(U)$.

2.1 Additive process CDF

The Lewis (2001) formula for the CDF is based on the characteristic function, and thanks to the Lévy-Khinchine formula, the characteristic function of an ATS process f_t admits a closed-form expression, which at time t is

$$\phi_t(u) = \mathcal{L}_t\left(iu\left(\frac{1}{2}+\eta_t\right)\sigma_t^2 + \frac{u^2\sigma_t^2}{2}; k_t, \alpha\right) e^{-iu\, ln\mathcal{L}_t(\eta_t\sigma_t^2; k_t, \alpha)} \tag{1}$$

σ_t, k_t are continuous on $[0,\infty]$ and η_t is continuous on $(0,\infty)$, with $\sigma_t > 0, k_t, \eta_t \geq 0$. As in the corresponding Lévy case, the Laplace exponent is defined as

$$ln\mathcal{L}_t(u; k, \alpha) := \frac{t}{k}\frac{1-\alpha}{\alpha}\left\{1 - \left(1 + \frac{u\,k}{1-\alpha}\right)^{\alpha}\right\} \tag{2}$$

with $\alpha \in (0,1)$.
As we introduced above, for all the algorithm implementations, we focus on the subfamily of power-law scaling ATS, which are characterized by parameters of the form

$$k_t = \bar{k} t^{\beta},\ \eta_t = \bar{\eta} t^{\delta},\ \sigma_t = \bar{\sigma} \tag{3}$$

where $\bar{\sigma}, \bar{k}, \bar{\eta} \in \mathbb{R}^+$ and $\beta, \delta \in \mathbb{R}$. In a nutshell, σ_t controls the average level of the volatility surface; k_t is related to the convexity of the implied volatility surface and η_t is linked to the volatility skew.
This specific subfamily of the ATS processes has been shown to be particularly accurate for the mathematical modeling of equity derivatives (see e.g., Azzone and Baviera 2021a).
Thanks to the independent increment property of ATS processes, the characteristic function of the increments $f_t - f_s$ between times s and $t > s$ is defined as

$$\phi_{s,t}(u) = \mathbb{E}[e^{i\,u\,(f_t - f_s)}] = \frac{\mathbb{E}[e^{i\,u\,f_t}]}{\mathbb{E}[e^{i\,u\,f_s}]} \tag{4}$$

Moreover, by means of Lukacs (1972, th.3.1, p.12), the ATS characteristic function $\phi_t(u)$ is analytic in the horizontal strip delimited on the imaginary axis by two values, defined as $p_t^- > 0$ and $-(p_t^+ + 1) < -1$. Azzone and Baviera (2021b) show that under an assumption on these two pure imaginary points it is possible to determine the strip of regularity of the characteristic function of the increments $\phi_{s,t}(u)$, which is identified for values of u s.t. $Im(u) \in (-(p_t^+ + 1), p_t^-)$. We report this assumption and, thanks to that, the CDF closed-form expression can be used.

Assumption 1. *p_t^+ and p_t^- are non increasing in t*

The Additive Normal Tempered Stable (ATS) processes satisfy this assumption, the proof can be found in Appendix A.
Following the approach of Azzone and Baviera (2021b), based on Lewis (2001), and under assumption 1 we get the CDF formula (5) for an additive process and we focus on the case with $a > 0$, where the positive constant a defines the shift for the integration path to be within the horizontal analyticity strip of $\phi_{s,t}(u)$.

$$P(x) = 1 - \frac{e^{-ax}}{\pi} \int_0^{\infty} Re\left[\frac{e^{-iux}\phi_{s,t}(u - ia)}{iu + a}\right] du \tag{5}$$

2.2 Numerical approximation of the CDF

The Fourier transform above is approximated by a discrete Fourier transform. This leads to numerical errors and, as it has been shown by Azzone and Baviera (2021b), by leveraging on further assumptions on the characteristic function of the increments, it is possible to estimate an explicit bound for the numerical error and find an optimal shift a in the CDF formula. We report the assumption over here, and it can be found in Appendix A that for an ATS process this assumption is verified.

Assumption 2. *$\forall t > s \geq 0$ there exists $B > 0, b > 0$ and $\omega > 0$ such that, for sufficiently large u, the following bound for the absolute value of the characteristic function holds*

$$|\phi_{s,t}(u)| < Be^{-bu^{\omega}} \qquad \forall a \in (0, p_t^+ + 1) \tag{6}$$

If assumptions 1 and 2 hold, it can be proven that the optimal bound for the numerical error of the CDF holds selecting the shift in CDF formula $a = \frac{p_t^+ + 1}{2}$.
The approximation of the CDF is computed as

$$\hat{P}(x) := 1 - \frac{e^{-ax}}{\pi} \sum_{l=0}^{N-1} Re\left[\frac{e^{-i(l+1/2)hx}\phi_{s,t}((l+1/2)h - ia)}{i(l+1/2)h + a}\right] \cdot h \tag{7}$$

where h is the step size in the Fourier domain [1] and N is the number of points in the grid.
For the MC scheme, we need the CDF values for a large number of points in its domain, so a regular grid with step size γ is used. A computationally efficient algorithm for the numerical approximation is the Fast Fourier transform (FFT), which is characterized by the following constrains on the parameters defining the discretization grid:

$$N = 2^M \tag{8}$$

$$\gamma\, h = \frac{2\pi}{N} \tag{9}$$

where $M \in \mathbb{N}$ is the simplest choice to define grid granularity.
In the FFT algorithm, the degrees of freedom are usually two: N, which is selected by choosing M, and u_1 or γ. If you want to control how dense you want the CDF domain grid you select γ, whereas to control the truncation error you can select u_1. In this report, we follow the Azzone and Baviera (2001b) implementation, where by leveraging on the numerical CDF error bound expression, the step size in the Fourier domain grid $h = h(N)$ is selected such that the dicretization error and the range error are of the same order. The two sources of the CDF error are comparable by selecting

$$h(N) = \left(\frac{\pi(p_t^+ + 1)}{b}\frac{1}{N^\omega}\right)^{\frac{1}{\omega+1}} \tag{10}$$

where the parameters b and ω are chosen such that

$$0 < b < \frac{(1-\alpha)^{1-\alpha}}{2^\alpha \alpha}\left(\frac{t \cdot \sigma_t^{2\alpha}}{k_t^{1-\alpha}} - \frac{s \cdot \sigma_s^{2\alpha}}{k_s^{1-\alpha}}\right) \tag{11}$$

$$0 < \omega < 2\alpha \tag{12}$$

The explicit computation which allows the exploitation of the built-in Matlab function `fft` to compute the CDF can be found in Appendix B.

2.3 CDF inversion via interpolation

Starting from the formula of the CDF $P(x)$, we compute an approximation via numerical scheme $\hat{P}(x)$. As highlighted by Azzone and Baviera (2021b), to get an adequate numerical inversion of the CDF it is important to have an increasing CDF and for it to be inside the interval $[0, 1]$. To get this property, the CDF domain grid is truncated between two points $x_0 < 0$ and $x_K > 0$, and so we consider a subset of the original grid, $x_0 < ... < x_K$ with $K < N$. The final step of the simulation algorithm is to sample a uniform r.v in $[0, 1]$ U, and invert

[1] In Azzone and Baviera (2021b) we believe that in the discrete formula of the CDF there is a missing term: the infinitesimal increment du, meaning the step size h in the Fouries domain.

the CDF $\hat{P}$ using interpolation. Following Azzone and Baviera (2021b), they propose to use a spline interpolation and show how this choice improves the accuracy of the numerical pricing in terms of various error metrics with respect to the linear interpolation. In the following, we try to reproduce the results of their work, by solving the same European call option pricing problem.

3 Option pricing with FFT

In this section we show an application of the ATS processes, in particular we price 30 European call options via Lewis-FFT-S simulation method and we compare these prices with the ones obtained by Lewis' closed formula, applied to the power-law scaling ATS.
In the case we considered, the initial value of the underlying asset is $S_0 = 100€$, there is no dividend yield, and the rate curve is the same as February 15, 2008. In addition, the 30 European call options have a maturity t of one month and log-moneyness in a regular grid defined as $\sqrt{t} \cdot (-0.2, 0.2)$.

Hereinafter, the ATS index of stability α is set equal to 0.75 and all the ATS parameters are chosen as reported in Table 1.

β	δ	$\overline{k}$	$\overline{\eta}$	$\overline{\sigma}$
1	-1/2	1	1	0.2

Table 1: ATS parameters used in all numerical simulations. These selected parameters are consistent with the ones observed in market data.

3.1 Option pricing with Lewis' closed formula

First, we compute the prices using Lewis' closed formula, defined by the following expression:

$$C(x) = B(t_0, t)F_0 \left(1 - e^{-\frac{x}{2}} \int_{-\infty}^{\infty} \frac{d\xi}{2\pi} e^{-i\xi x} \phi_t \left(-\xi - \frac{i}{2}\right) \frac{1}{\xi^2 + \frac{1}{4}}\right) \tag{13}$$

where $\phi_t(u)$ is the ATS characteristic function, $x := lnF_0/K$ is the moneyness, and $B(t_0, t)$ is the discount factor between the value date and the maturity date.
The characteristic function $\phi_t(\xi)$ and ATS parameters are defined in Eq. (1) , (2) , (3).

3.2 Option pricing with Lewis-FFT-S simulation method

For the computation of call option prices with Lewis-FFT-S simulation method, a higher level of difficulty is reached. Going into it more specifically, for this computation we need to take advantage of the simulation of the increments of the log-returns of forwards, simulated through the Fast Fourier Transform (FFT).
The payoff for an European call is $[F(t, t) - K]^+$, but since it is unbounded, it is convenient to rewrite it as

$$F(t, t) - min(F(t, t), K), \tag{14}$$

where the payoff of a covered call is used. In the following, we explicit the pricing formula by exploiting this new payoff.
The general formula for an European call option is

$$C(t_0, t) = B(t_0, t) \cdot \mathbb{E}_0[(F(t, t) - K)^+], \tag{15}$$

but considering the dependence on moneyness x and introducing the forward log-return f_t, we can write

$$C(t_0, t, x) = B(t_0, t) \cdot \mathbb{E}_0[(F_0 e^{f_t} - F_0 e^{-x})^+] \tag{16}$$

Finally using the result of Eq. (14) and integrating it into the Eq. (16) , we can rewrite the price as

$$C(t_0, t, x) = B(t_0, t) \cdot F_0 \cdot (1 - \mathbb{E}_0[min(e^{f_t}, e^{-x})]), \tag{17}$$

where $f_t = ln\frac{F_t}{F_0}$ is simulated using the Lewis-FFT-S method. For the numerical pricing, we perform a first implementation by setting M equal to 12 and the number of simulation for the ATS increment equal to 10^7, the remaining parameters are all set as defined above.

3.3 Error Metrics

To analyze and compare the prices (as shown in Table 12, Appendix D) obtained with the numerical method with respect to the ones obtained by Lewis' closed formula, we use the following error metrics:

- Root-mean-square error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{N} \left(P_{num}(i) - P_{Lewis}(i)\right)^2}$$

- Maximum error

$$MAX = \max_{i=1,\dots,N} |P_{num}(i) - P_{Lewis}(i)|$$

- Mean average percentage error (MAPE)

$$MAPE = \frac{100}{n} \sum_{i=1}^{N} \left| \frac{P_{num}(i) - P_{Lewis}(i)}{P_{Lewis}(i)} \right|$$

As can be seen from the results shown in the Table 2, the RMSE and maximum error are of the order of the basis point (bp) and the MAPE is less than 0.05%.

RMSE [bp]	Maximum error [bp]	MAPE [%]
2.7709	5.1282	0.0173

Table 2: Pricing errors

3.4 Pricing errors with different number of grid points

To analyze in detail the performances of the Lewis-FFT simulation algorithm, we replicate the computation of call option prices with a different number of FFT grid points (M). In particular, we compare the computational times derived from the Lewis-FFT-S method (spline interpolation) and the Lewis-FFT-L method (linear interpolation) and their errors with reference to the Lewis' closed formula prices (independent of the change in the M parameter).

As can be seen from Table 3, although the computational times are all less than 0.2 seconds (s) for both methodologies and for all values of M in range (7-13), there is a slight difference in favor of the Lewis-FFT-L method, except for the values corresponding to M equal to 7.

M	Computational time (Spline) [s]	Computational time (Linear) [s]
7	0.133376	0.141403
8	0.139747	0.129437
9	0.143106	0.132572
10	0.146521	0.136126
11	0.151083	0.139212
12	0.156298	0.141709
13	0.159889	0.149645

Table 3: Computational time with Lewis-FFT-S method and Lewis-FFT-L method for different number of FFT grid points (M)

For further feasible evidence on the goodness of the algorithm, we analyze the errors with reference to the Lewis' closed formula prices. From Table 4 and Table 5, containing the errors with Lewis-FFT-S method and Lewis-FFT-L method respectively, we see that the errors related to the second method (Linear), are significantly higher than the first one (Spline), at least for the first three values of M. The small differences in time therefore do not justify the choice of the method using linear interpolation.

M	RMSE [bp]	Maximum error [bp]	MAPE [%]
7	11.304169	13.443769	0.075664
8	2.903766	4.773682	0.012829
9	1.905766	3.306820	0.009015
10	2.427118	4.545439	0.014638
11	2.679463	4.976047	0.016644
12	2.770908	5.128212	0.017342
13	2.802163	5.178267	0.017575

Table 4: Errors computed with Lewis-FFT-S method for different number of FFT grid points.

M	RMSE [bp]	Maximum error [bp]	MAPE [%]
7	342.224468	435.017465	2.282103
8	197.805986	254.516190	1.322016
9	114.710828	148.712040	0.766142
10	66.914369	87.657994	0.446989
11	39.407466	52.359100	0.263874
12	23.622004	32.245394	0.158710
13	14.570412	20.703120	0.098219

Table 5: Errors computed with Lewis-FFT-L method for different number of FFT grid points.

In order to better appreciate the error differences between the two methods, we analyze their trend on a logarithmic scale.

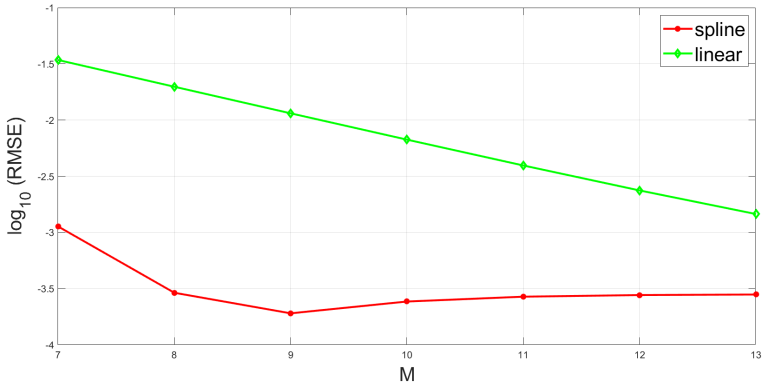

Figure 1: RMSE errors computed with Lewis-FFT-S method for different number of FFT grid points.

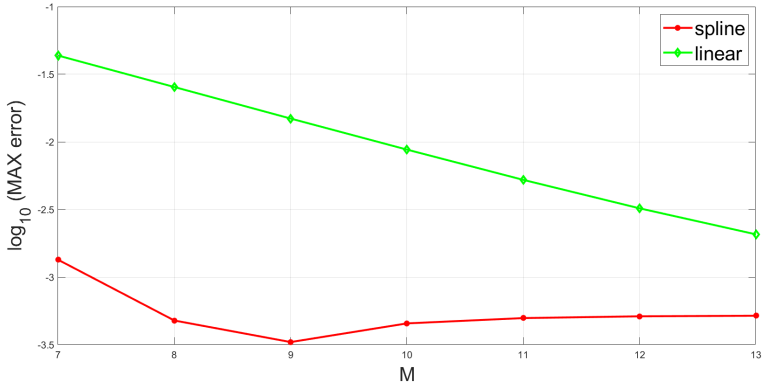

Figure 2: Maximum errors computed with Lewis-FFT-S method for different number of FFT grid points.

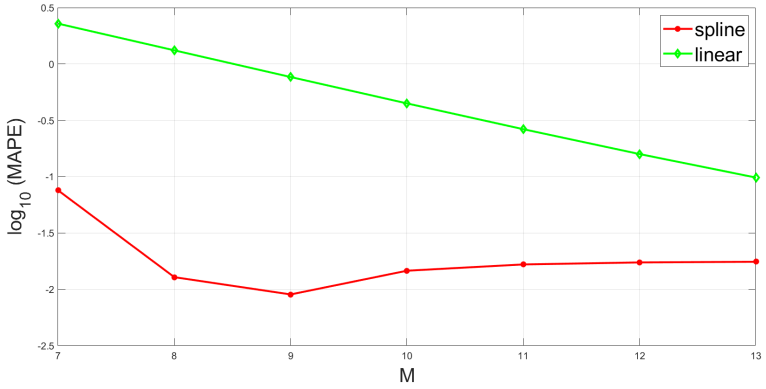

Figure 3: MAPE errors computed with Lewis-FFT-S method for different number of FFT grid points.

We can notice that for each M the spline interpolation errors are significantly below the linear interpolation errors. Moreover, spline interpolation errors improve faster than the linear interpolation errors. Considering the Spline method, the RMSE and maximum error are of the order of the basis point and the MAPE is less than 0.02%, except for M equal to 7. Instead, using the Linear method the RMSE and maximum error are of an order of 15 to 400 times higher than the basis point and the MAPE is always higher than 0.05%.

3.5 Option pricing: Conclusions

In conclusion, although the computational time is slightly less with the Lewis-FFT-L method, the error calculation is clearly in favor of the Lewis-FFT-S method. In particular, beyond a certain threshold of discretization of the FFT grid points, the errors seem not to further improve; with M equal to 9, the best results are obtained because comparison with the prices calculated by Lewis' closed formula shows smaller errors. This application of the ATS processes, for a sufficiently large number of simulations show that this numerical procedure is a valid solution of the equity derivative pricing problem.

4 Fractional FFT algorithm

In this section, we implement an alternative version of the Lewis-FFT algorithm. In particular, the main difference of this new numerical method lays on the CDF approximation, where instead of using an FFT algorithm to compute the Fourier transform, we use the Fractional Fourier transform scheme (FRFT) (see e.g., Chourdakis 2005). An explanation of the implementation of the FRFT algorithm to the specific problem of CDF approximation (Eq. (5)) can be found in appendix C, with all the computations being developed. In the following, we apply this numerical scheme (hereinafter, Lewis-FRFT) to the option pricing problem and we compare the numerical results with Lewis-FFT, in terms of errors and computational times. We conclude this section by analyzing the pros and cons of using FRFT instead of FFT in the numerical approximation of the CDF. The initial situation of the underlying asset and the various details of the call options (maturity, moneyness, etc.) are the same as those presented in Section 3.The number of simulation for the ATS increment is equal to 10^7 and the ATS parameters are as before.

4.1 Option pricing with Lewis-Fractional-FT-S simulation method

The calculations and steps for the explicit formula for the price of a European call option as a function of the moneyness and increments of the log-returns of forwards (f_t), are the same as those introduced in subsection 3.2.

The major difference of the FRFT technique from that of FFT is the introduction of a new parameter that releases the step size in the Fourier domain (h) from the step size in the CDF domain regular grid (γ), previously linked by the relationship ($\gamma\ h = \frac{2\pi}{N}$). Parameter α corresponds to a new degree of freedom and relates h and γ as follows:

$$\alpha = \frac{\gamma\, h}{2\pi}$$

By obviously imposing $\alpha = \frac{1}{N}$ (with $N = 2^M$) the values obtained by FRFT method are the same as those obtained by FFT.

In the next section we will reason on how to optimize this parameter for our purposes.

4.2 α optimization

In order to take advantage of the extra degree of constraint offered by the α parameter, we need to calibrate it. Our main goal is always to find a method for computing the price of a European call option in such a way that its error is as low as possible, always calculated by reference to Lewis' closed formula (independent on the variation of the α parameter). To get a more complete overview, finally we want to analyze the situations for all values of M in range (7-13). Therefore, in order to find the optimal value for α, we minimize the errors presented in subsection 3.3: RMSE, Maximum error and MAPE. We defined the functions of the errors as parameter α changes, and then minimized them to find the optimal value for each type of error and for each value of M (starting from $\alpha = \frac{1}{2^M}$). The results are shown below.

M	starting α	optimal α	RMSE [bp]	Maximum error [bp]	MAPE [%]	Time [s]
7	0.007812	0.004883	1.812387	3.327286	0.006932	0.135844
8	0.003906	0.003223	1.812367	3.328269	0.006931	0.141268
9	0.001953	0.002051	1.833094	3.207315	0.007591	0.139338
10	0.000977	0.001025	2.353572	4.411342	0.013987	0.146808
11	0.000488	0.000513	2.651105	4.928669	0.016434	0.151048
12	0.000244	0.000256	2.761075	5.112134	0.017269	0.159099
13	0.000122	0.000128	2.798828	5.172837	0.017551	0.166816

Table 6: Errors computed with FRFT technique for different value of M, minimizing the RMSE.

M	starting α	optimal α	RMSE [bp]	Maximum error [bp]	MAPE [%]	Time [s]
7	0.007812	0.004590	1.867516	3.136257	0.008293	0.136364
8	0.003906	0.003027	1.867946	3.138499	0.008302	0.142707
9	0.001953	0.002051	1.833094	3.207315	0.007591	0.140815
10	0.000977	0.001025	2.353572	4.411342	0.013987	0.146168
11	0.000488	0.000513	2.651105	4.928669	0.016434	0.151035
12	0.000244	0.000256	2.761075	5.112134	0.017269	0.157708
13	0.000122	0.000128	2.798828	5.172837	0.017551	0.167826

Table 7: Errors computed with FRFT technique for different value of M, minimizing the maximum error.

M	starting α	optimal α	RMSE [bp]	Maximum error [bp]	MAPE [%]	Time [s]
7	0.007812	0.005078	1.843498	3.438344	0.006733	0.137611
8	0.003906	0.003418	1.889230	3.505815	0.006784	0.141270
9	0.001953	0.002197	1.834121	3.420458	0.006768	0.143485
10	0.000977	0.001416	1.813918	3.354810	0.006887	0.139755
11	0.000488	0.000977	1.875350	3.487185	0.006764	0.142090
12	0.000244	0.000256	2.761075	5.112134	0.017269	0.157560
13	0.000122	0.000128	2.798828	5.172837	0.017551	0.172595

Table 8: Errors computed with FRFT technique for different value of M, minimizing the MAPE.

In order to better appreciate the errors, we analyze their trend on a logarithmic scale (base 10).

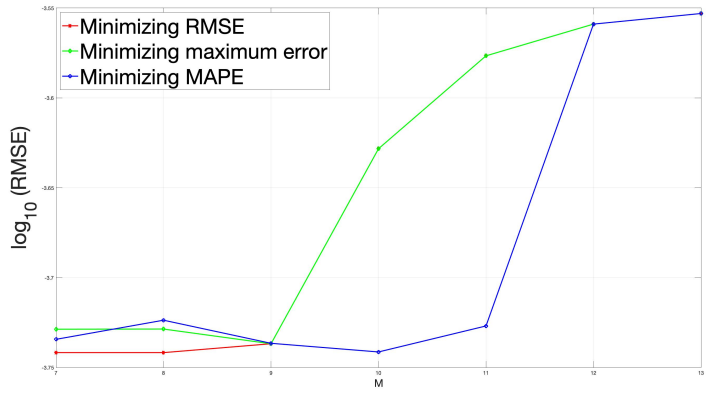

Figure 4: RMSE errors computed with FRFT technique for different value of M.

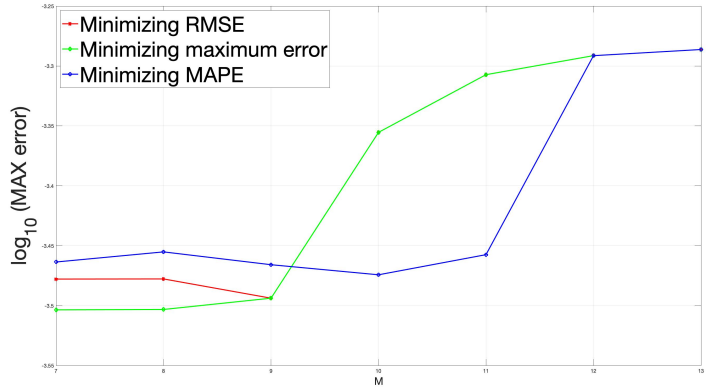

Figure 5: Maximum errors omputed with FRFT technique for different value of M.

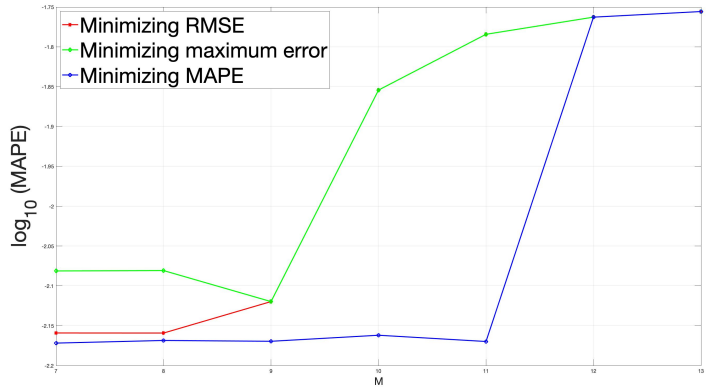

Figure 6: MAPE errors computed with FRFT technique for different value of M.

As can be seen from the graphs, the values of the errors (RMSE, Maximum error and MAPE), minimizing RMSE and the Maximum error become equal for values of M that are greater or equal to 9. Those computed by minimizing MAPE become equal to the others only when M is equal to 12 or for larger values.

4.3 Option pricing: Conclusions

Comparing the FRFT technique with the FFT (subsection 3.4), we can see that the computational times are very similar, almost all less than 0.2 seconds per simulation. Shifting our attention to the errors and comparing them with those computed by the Lewis-FFT-S method, we can easily see a decrease (especially so when M is equal to 7) and thus an improvement in our model that leads us to say that the prices of European call options with maturity of one month are more accurate with the FRFT method.

5 Example: Phoenix option

In the following, we show that the proposed technique for additive processes can be used to price path-dependent options efficiently. The case of a Phoenix option is discussed in details.

For sake of simplicity, once again we consider the case with underlying initial value $S_0 = 100$, no dividend and the interest rate curve described as 15 Feb 2008.
Moreover, let us recall the payoff P of this derivative:

$$P = \frac{1}{S_0} \cdot \sum_{j=1}^{i} max(S_{t_j} - K_1, 0) \tag{18}$$

where $0{=}t_0 < t_1 < ... < t_i < ... < t_n$ with n=6 the monitoring times (i.e. yearly reset dates over a six-year time frame), strikes $K_1 = 100$ and $K_2 = 120$. It is remarked that this exotic option is called once only and if it happens this moment corresponds to the first time the underlying overpasses the value of K_2 at a reset date [i.e $S_{t_i} > K_2$ at t_i]. There, the option expires otherwise it is never called and therefore it pays nothing.

In a nutshell, our interest is reduced to simulating a discrete sample path of the process S_t over a finite time horizon: we are only concerned about the values of the underlying on such a discrete-time grid made of reset dates only.
First, we simulate the ATS increment via LewisFFT algorithm. Then, using this framework, we are able to exploit the relationship between the forward and the underlying (i.e. $F(t,t){=}S_t$) and this allows the identification of the discrete sample of the latter. Thus, we make extensive use of the following mathematical relationships:

$$F_0 = \frac{S_0}{B(t_0, t_1)} \tag{19}$$

$$S_{t_i} = F(t_{i-1}, t_i) \cdot e^{f_{TTM_{i+1}} - f_{TTM_i}} \tag{20}$$

$$F(t_i, t_{i+1}) = S_{t_i} \cdot e^{fzr_i \cdot \delta_i} \tag{21}$$

where for every reset date t_i, with $i = 1, ..., 6$, (note that in Eq.(20) the index i varies in [0,5]),

$$fzr_i = -\frac{1}{\delta_i} \cdot log \frac{B(t_0, t_i)}{B(t_0, t_{i-1})} \tag{22}$$

it describes the forward zero rate, $B(t_0, t_i)$ denotes the discount factor, $\delta_i = t_i - t_{i-1}$ the year gap between two consecutive reset dates and $TTM_i = t_i - t_0$ the time to maturity.

The stochastic process S_t admits the following trajectory, out of a sample made of 10^4 randomly chosen simulations within a total of 10^7.

Figure 7: Underlying simulation path

Concerning the pricing problem, the obtained numerical results (in €) are as below:

Lower Bound	Price	Upper bound
0.257103	0.257265	0.257427

Table 9: 95% price confidence interval, exploiting t-student quantiles

After having performed for the first time our pricing computations, we let N_{sim} (as in the number of simulations needed for the pricing model) vary and value the corresponding prices, confidence intervals as well as the total computational time involved at each iteration. Firstly, it should be noted that a considerable increment in the computational time is directly linked to a bigger value of N_{sim}. On the other hand, this trend reverses estimating the width of the 95% confidence intervals.

In this condition a problem arises: various criteria could be selected and used to solve it. Based on this fact and parsimony principle, $N = 10^7$ was a good trade off between the two parameters. Therefore, our numerical results are all related to such a value of N_{sim}. Similarly, using the Occam's razor, a lower number of simulations could have been chosen instead but a wider confidence interval would have been the downside. Graphically speaking,

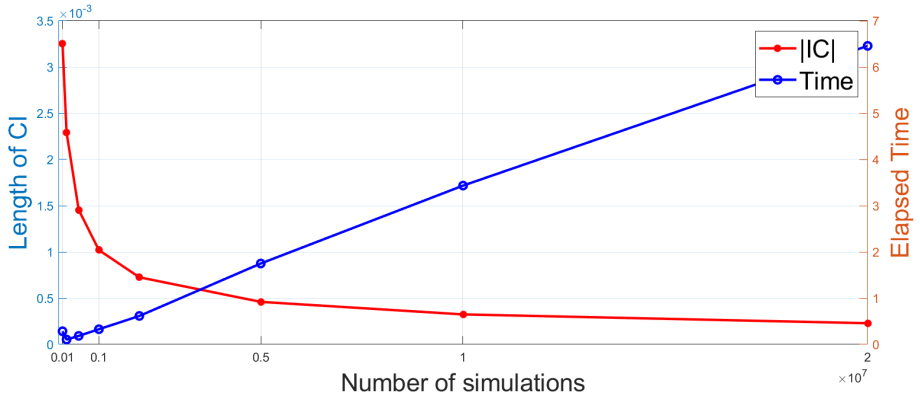

Figure 8: CI width vs Computational time

5.1 Blackmodel comparison

In order to compare the pricing and computational performance of the FFT-S algorithm, we use the Black model as a benchmark algorithm. Hence, we model the process $\{f_t\}_{t\geq 0}$ in the simplest case, exploiting Brownian motion simulations. The same financial framework, number of simulations and exotic product as above are considered.

The following numerical results are obtained:

	price [€]	CI (95%) [€]	
FFT-S	0.257265	0.257103	0.257427
Black	0.255035	0.254886	0.255183

Table 10: Prices of Phoenix option via FFT-S algorithm and Black model

One can easily verify that the two confidence intervals do not overlap.
In addition, one final observation can be made regarding the computational times:

FFT computational time [s]	Black method computational time [s]
1.398991	1.301196

Table 11: Comparison of computational times between FFT-S and Black simulations at reset dates.

It is worth noting that the speed of the FFT-S method is comparable to computing the simulation of the underlying via Black model.

6 Conclusion

In this report, we propose and analyze an application of the Lewis FFT-S method to an ATS process with power-law scaling and time-dependent parameters. According to Azzone and Baviera (2021b), our study was conducted by leveraging on the numerical efficiency of a Fast Fourier Transform algorithm and on the accuracy of the results coming from the spline interpolation technique.

We have compared the Lewis FFT-S algorithm with two alternative methods while pricing 30 European call options with one-month maturity. First, a linear interpolation has been used. The numerical results of the linear method show that lower computational times do not justify neither the magnitude of the errors induced nor the convergence of the method itself. Second, we have ensured that the results obtained via spline interpolation were consistent with the Lewis' closed formula at least at a 95% level of confidence.
In this framework, we have also implemented the FRFT algorithm and performed the pricing procedure again: computational times do not differ from the FFT-S algorithm significantly and the results are more accurate, based on the decrease of the errors. However, in terms of computational costs we find no reasons to justify and prefer the FRFT approach to the FFT-S algorithm. Therefore, the benefits of choosing the grid sizes independently are disregarded due to the higher complexity of tasks that a processor must perform. This conclusion is in agreement with Azzone and Baviera's (2021b) work. Indeed, memory allocation and the number of operations become discriminating factors. On one hand, $N \log_2 N$ operations are needed to perform a N-point FFT procedure but, on the other hand, one N-point FRFT invokes by definition three $2N$-point FFT, meaning that approximately $6N \log_2 N$ operations are implied.

For sake of honesty, we are well aware that our results in terms of errors are two orders of magnitude higher than the ones obtained in Azzone and Baviera (2021b). We did not manage to fall under the one basis point threshold, even though according to literature not only should that have been feasible but the errors could have been close to a multiple of one hundredth of a basis point.

Finally, we have verified the reliability of the simulation algorithm pricing a path-dependant exotic derivative (i.e. the Phoenix option) and comparing the results with an algorithm perfoming Brownian motion simulations. Both methods present the same order of magnitude in the computational time, meaning once again that the Lewis FFT-S algorithm is fast and produces accurate results.

7 Bibliography

1. Azzone, M., and R. Baviera. A fast Monte Carlo scheme for additive processes and option pricing. arXiv preprint arXiv:2112.08291 (2021).

2. Azzone, M. and R. Baviera. Additive normal tempered stable processes for equity derivatives and power- law scaling. Quantitative Finance 22.3 (2022): 501-518.

3. Chourdakis K., Option Pricing Using the Fractional FFT, Journal of Computational Finance, Vol. 8, No. 2, 2004, pp. 1-18.

4. Lewis, A.L., 2001. A simple option formula for general jump-diffusion and other exponential L´evy processes, Available on SSRN, ssrn.com/abstract=282110.

5. Lukacs, E., 1972. A survey of the theory of characteristic functions, Advances in Applied Probability, 4 (1), 1–37.

Appendix A

In this appendix we report for convenience the proof that, for ATS processes with $\alpha \in (0,1)$, Assumption 1 and 2 hold.
First, we recall a sufficient condition for the existence of ATS:

Theorem A. Sufficient conditions for existence of ATS
There exists an additive process $\{f_t\}_{t\geq 0}$ with the characteristic function (Eq.(1)) if the following two conditions hold.

1. *$g_1(t)$,$g_2(t)$, and $g_3(t)$ are non decreasing, where*

$$g_1(t) := (1/2+\eta_t) - \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)}$$
$$g_2(t) := -(1/2+\eta_t) - \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)}$$
$$g_3(t) := \frac{t^{1/\alpha}\sigma_t^2}{k_t^{(1-\alpha)/\alpha}} \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(\sigma_t^2 k_t)};$$

2. *Both $t\sigma_t^2\eta_t$ and $t\ \sigma_t^{2\alpha}\ \eta_t^\alpha\ /\ k_t^{1-\alpha}$ go to zero as t goes to zero.* □

Proof of assumption 1

We show that p_t^+ and p_t^- are non decreasing for an ATS process. At time t, the ATS characteristic function in equation (1) is analytic on the imaginary axis $u = -ia$, $a \in \mathbb{R}$ iff

$$a + \frac{k_t}{1-\alpha}\left(a(\frac{1}{2}+\eta_t)\sigma_t^2 - \frac{a^2\sigma_t^2}{2}\right) > 0$$

By solving the second order inequality, we get

$$g_1(t) < a < -g_2(t)$$

with $g_1(t)$ and $g_2(t)$ defined above. Hence, $p_t^+ := -g_2(t) - 1$ and $p_t^- := -g_1(t)$. Notice that $-g_2(t)$ and $-g_1(t)$ are non increasing with respect to t because of condition 1 of the previous theorem and p_t^+ and p_t^- are positive through direct inspection. Moreover, $p_t^+ \geq p_t^-$ because

$$p_t^+ - p_t^- = 2\eta_t \geq 0 \tag{23}$$

Proof of assumption 2

We observe that, by condition on $g_1(t)$ and $g_2(t)$ of theorem A, we have

$$g(t) := -(g_1(t) + g_2(t)) = \sqrt{(1/2+\eta_t)^2 + 2(1-\alpha)/(k_t\sigma_t^2)}$$

is non increasing wrt t. Hence, thanks to condition 1 on $g_3(t)$ of theorem A

$$\frac{t}{k_t^{1-\alpha}}\sigma_t^{2\alpha} \text{ is increasing in } t \tag{24}$$

We have to show that, given s and t, there exists $B > 0$, $b > 0$ and $\omega > 0$ such that, for sufficiently large u, Assumption 2 holds for the characteristic function of ATS.

We choose $log(B) > \frac{1-\alpha}{\alpha}(\frac{t}{k_t} - \frac{s}{k_s})$, $0 < b < \frac{(1-\alpha)^{1-\alpha}}{2^\alpha \alpha}(\frac{t\sigma_t^{2\alpha}}{k_t^{1-\alpha}} - \frac{s\sigma_s^{2\alpha}}{k_s^{1-\alpha}})$ and $0 < \omega < 2\alpha$.
Notice that it is possible to fix $b > 0$, because (Eq. (24)) holds. Moreover, the imaginary part of the exponent in (Eq. (1)) does not contribute to B, because the absolute value of the exponential of an imaginary quantity is unitary.
For sufficiently large u, $|\phi_{t,s}(u - ia)|$ goes to zero faster than Be^{-bu^ω} because $log\ \phi_{t,s}(u - ia)$ is asymptotic to

$$-\frac{(1-\alpha)^{1-\alpha}}{2^\alpha \alpha}\left(\frac{t \cdot \sigma_t^{2\alpha}}{k_t^{1-\alpha}} - \frac{s \cdot \sigma_s^{2\alpha}}{k_s^{1-\alpha}}\right) u^{2\alpha}$$

that is negative due to (Eq. (24)) for $\alpha \in (0, 1)$.

Appendix B

We show how to directly use the Matlab built-in `fft` function to implement the MC simulation scheme. Starting from the following Fourier transform discretization

$$I(x_k) = \int_0^\infty Re\left[\frac{e^{-iux_k}\phi_{s,t}(u - ia)}{iu + a}\right] du = \sum_{l=0}^{N-1} Re\left[\frac{e^{-i\ u_l \cdot x_k}\phi_{s,t}(u_l - ia)}{i \cdot u_l + a}\right] \cdot h$$

where the two grids, respectively in the Fourier domain and the CDF domain, are defined as

$$\begin{aligned} u_l &= \frac{1}{2}h + l \cdot h\ , \qquad l = 0,\ 1, \ldots, N-1 \\ x_k &= x_1 + k \cdot \gamma\ , \qquad k = 0,\ 1, \ldots, N-1 \end{aligned}$$

We choose the step size in the Fourier domain h as explained in section 3.2, i.e.

$$h(N) = \left(\frac{\pi(p_t^+ + 1)}{b} \cdot \frac{1}{N^\omega}\right)^{\frac{1}{\omega+1}}$$

Moreover, we impose the FFT parameters constrains and since we want to compute the CDF on a symmetric grid around zero, we get

$$\begin{aligned} N &= 2^M \\ \gamma\ h &= \frac{2\pi}{N} \\ x_1 &= -\frac{(N-1) \cdot \gamma}{2} \end{aligned}$$

Now, let's make the phase explicit in the DFT (discrete Fourier transform) by defining the function $f(u)$ as

$$f(u_l) = \frac{\phi_{s,t}(u_l - ia)}{i \cdot u_l + a}$$

Then, we can write

$$
\begin{aligned}
I(x_k) &= \sum_{l=0}^{N-1} Re\left[\frac{e^{-i\ u_l \cdot x_k}\phi_{s,t}(u_l - ia)}{i \cdot u_l + a}\right] \cdot h \\
&= \sum_{l=0}^{N-1} Re\left[e^{-i\ u_l \cdot x_k} \cdot f(u_l)\right] \cdot h \\
&= h \cdot Re\left[\sum_{l=0}^{N-1} e^{-i\ u_1 \cdot x_k} e^{-i\ l \cdot h \cdot x_1} e^{-i \cdot \frac{2\pi}{N} l \cdot k} \cdot f(u_l)\right] \\
&= h \cdot Re\left[e^{-i\ u_1 \cdot x_k} \sum_{l=0}^{N-1} e^{-i\ l \cdot h \cdot x_1} e^{-i \cdot \frac{2\pi}{N} l \cdot k} \cdot f(u_l)\right] \\
&= h \cdot Re\left[e^{-i\ u_1 \cdot x_k} \sum_{l=0}^{N-1} f_l \cdot e^{-i \cdot \frac{2\pi}{N} l \cdot k}\right] \\
&= h \cdot Re\left[e^{-i\ u_1 \cdot x_k} \cdot \texttt{fft}(f_l)\right]
\end{aligned}
$$

where f_l is the phase, which can be directly used as input of the `fft` Matlab function.

Appendix C

In this appendix, we follow the Chourdakis (2004) FRFT implementation using a similar approach as in appendix B. We want to approximate the following integral

$$
I(x_k) = \int_0^\infty Re\left[\frac{e^{-iux_k}\phi_{s,t}(u - ia)}{iu + a}\right] du = \sum_{l=0}^{N-1} Re\left[\frac{e^{-i\ u_l \cdot x_k}\phi_{s,t}(u_l - ia)}{i \cdot u_l + a}\right] \cdot h
$$

We can re-write, following the Chourdakis' notation

$$
I(x_k) = \sum_{l=0}^{N-1} Re\left[e^{-i\ u_l \cdot x_k}\psi(u_l) \cdot h\right] = Re\left[\sum_{l=0}^{N-1} e^{-i\ u_l \cdot x_k}\tilde{\psi}_l \cdot h\right]
$$

where

$$
\begin{aligned}
\psi(u) &= \frac{\phi_{s,t}(u - ia)}{iu + a} \\
\tilde{\psi}_l &= \psi(u_l) \cdot \omega_j
\end{aligned}
$$

In general, the values $\tilde{\psi}_l$ are set equal to $\tilde{\psi}_l = \psi(u_l) \cdot \omega_j$, with ω_j's implementing the integration rule. As highlighted by the Chourdakis (2004), the differences between the various integration schemes are insignificant. Therefore we use the trapezoidal rule to approximate the integral, whose integration weights are set to $\omega_j = \frac{1}{2}$ if $j = 0$ or $j = N - 1$, and $\omega_j = 1$ otherwise.
The Fourier domain grid and the CDF domain grid are respectively defined as follows:

$$
\begin{aligned}
u_l &= l \cdot h\ , \qquad l = 0,\ 1, \ldots, N-1 \\
x_k &= -((N-1)\gamma)/2 + k \cdot \gamma \qquad k = 0,\ 1, \ldots, N-1
\end{aligned}
$$

We now explicit the following sum

$$\begin{aligned}
\sum_{l=0}^{N-1} e^{-i\ u_l \cdot x_k} \tilde{\psi}_l \cdot h &= \sum_{l=0}^{N-1} e^{-i\ (-\frac{(N-1)\gamma}{2} + k\gamma)\cdot lh} \cdot \tilde{\psi}_l \cdot h \\
&= \sum_{l=0}^{N-1} e^{-i\ \gamma hkl} \cdot e^{i\ (-\frac{(N-1)\gamma}{2})\cdot lh} \cdot \tilde{\psi}_l \cdot h \\
&= \sum_{l=0}^{N-1} e^{-i\ \gamma hkl} \cdot f_l \\
&= \sum_{l=0}^{N-1} e^{-i\ kl\cdot(2\pi\alpha)} f_l \ =: D_k(\mathbf{f}, \alpha)
\end{aligned}$$

where in the last equality we highlight the parameter $2\pi\alpha = \gamma h$ which allows us to choose independently the two grid granularities. We now summarize the FRFT steps, we define the two following 2-N dimension vectors

$$\begin{aligned}
\mathbf{y} &= \left((f_l e^{-i\pi l^2 \alpha})_{l=0}^{N-1},\ (0)_{l=0}^{N-1} \right) \\
\mathbf{z} &= \left((e^{i\pi l^2 \alpha})_{l=0}^{N-1},\ (e^{i\pi (N-l)^2 \alpha})_{l=0}^{N-1} \right)
\end{aligned}$$

Then the FRFT is given by

$$D_k(\mathbf{f}, \alpha) = (e^{-i\pi k^2 \alpha})_{k=0}^{N-1} \odot D_k^{-1}(D_l(\mathbf{y}) \odot D_l(\mathbf{z}))$$

where the $\odot$ is the element by element product between two vectors and D_k^{-1} is the discrete inverse Fourier transform. For further details see Chourdakis (2004).

Appendix D

In this appendix we present the numerical values obtained with Lewis' closed formula and Lewis-FFT-S simulation method at varying moneyness values, with the respective confidence intervals.

Moneyness	Lewis' closed formula [€]	Lewis-FFT-S [€]	CI (95%) [€]	
-0.056374	0.414784	0.414957	0.411887	0.418027
-0.052487	0.470305	0.470453	0.467425	0.473480
-0.048599	0.532703	0.532839	0.529856	0.535821
-0.044711	0.602606	0.602749	0.599816	0.605683
-0.040823	0.680643	0.680796	0.677914	0.683678
-0.036935	0.767437	0.767625	0.764797	0.770452
-0.033047	0.863592	0.863831	0.861062	0.866601
-0.029159	0.969674	0.969984	0.967275	0.972693
-0.025271	1.086205	1.086600	1.083954	1.089245
-0.021383	1.213646	1.214100	1.211520	1.216680
-0.017496	1.352386	1.352870	1.350358	1.355382
-0.013608	1.502727	1.503240	1.500798	1.505682
-0.009720	1.664884	1.665395	1.663025	1.667766
-0.005832	1.838969	1.839445	1.837146	1.841743
-0.001944	2.024994	2.025414	2.023189	2.027639
0.001944	2.222867	2.223222	2.221070	2.225373
0.005832	2.432398	2.432700	2.430622	2.434777
0.009720	2.653298	2.653537	2.651532	2.655541
0.013608	2.885195	2.885387	2.883455	2.887318
0.017496	3.127636	3.127787	3.125927	3.129646
0.021383	3.380105	3.380228	3.378439	3.382017
0.025271	3.642032	3.642095	3.640375	3.643815
0.029159	3.912805	3.912817	3.911164	3.914469
0.033047	4.191789	4.191754	4.190167	4.193341
0.036935	4.478330	4.478262	4.476738	4.479786
0.040823	4.771776	4.771686	4.770223	4.773148
0.044711	5.071478	5.071365	5.069961	5.072768
0.048599	5.376807	5.376660	5.375313	5.378006
0.052487	5.687155	5.686976	5.685684	5.688269
0.056374	6.001945	6.001734	6.000494	6.002974

Table 12: Prices of the 30 call options as moneyness changes, calculated with Lewis' closed formula and Lewis-FFT-S simulation method, with confidence interval (95%)