INVERSE TRANSFORM METHOD FOR SIMULATING LEVY PROCESSES AND DISCRETE ASIAN OPTIONS PRICING

Zisheng Chen
Liming Feng

Xiong Lin

University of Illinois at Urbana-Champaign
Department of Industrial and
Enterprise Systems Engineering

Gresham Investment Management
New York, NY, USA

ABSTRACT

The simulation of a Lévy process on a discrete time grid reduces to simulating from the distribution of a Lévy increment. For a general Lévy process with no explicit transition density, it is often desirable to simulate from the characteristic function of the Lévy increment. We show that the inverse transform method, when combined with a Hilbert transform approach for computing the cdf of the Lévy increment, is reliable and efficient. The Hilbert transform representation for the cdf is easy to implement and highly accurate, with approximation errors decaying exponentially. The inverse transform method can be combined with quasi-Monte Carlo methods and variance reduction techniques to greatly increase the efficiency of the scheme. As an illustration, discrete Asian options pricing in the CGMY model is considered, where the combination of the Hilbert transform inversion of characteristic functions, quasi-Monte Carlo methods and the control variate technique proves to be very efficient.

1 INTRODUCTION

Lévy processes have been used to model the dynamics of various financial variables (Boyarchenko and Levendorskiĭ 2002, Cont and Tankov 2004, Schoutens 2003). One needs to simulate a Lévy process when Monte Carlo methods are used for the pricing and risk management of derivative securities in Lévy models. Standard techniques can be used when the transition density of the Lévy process is known explicitly and admits a simple form. Otherwise, one may simulate a Lévy process from its Lévy density, which characterizes the behavior of jumps. For an infinite activity Lévy process, one may discard small jumps when there are not too many such jumps (Cont and Tankov 2004) or approximate the small jump component by a Brownian motion and simulate from a jump diffusion process (Asmussen and Rosiński 2001). One may also compute infinite series representations of a Lévy process and simulate from such a series representation (Bondesson 1982, Rosiński 2001). When the Lévy process results from a Brownian subordination, it suffices to simulate the subordinator and a standard normal random variable. For example, see Rydberg (1997) for the case of simulating normal inverse Gaussian processes (Barndorff-Nielsen 1998). Finally, although the transition density of a general Lévy process is often not known explicitly, due to the celebrated Lévy-Khintchine formula for infinitely divisible distributions, the characteristic functions of many popular Lévy processes are known. Thus, it suffices to simulate from a given Lévy characteristic function.

One method to simulate from a characteristic function is to approximate the quantile function by solving a non-linear integro-differential equation (Shaw and McCabe 2009). One may also follow Devroye (1981), where a simple bound for the probability density is obtained and an acceptance-rejection method is then used. In this approach, to generate each replicate, an inverse Fourier transform integral needs to be computed. The computational cost could thus be high. Alternatively, one may invert the characteristic function, compute and tabulate the values of the cumulative distribution function on an appropriately

444

determined grid and then use the inverse transform method (Glasserman and Liu 2007, Glasserman and Liu 2010). To generate a replicate, one generates a uniform variate and interpolate to find the solution of the equation where the uniform variate equals the cdf. It is shown in Glasserman and Liu (2010) that the bias due to linear interpolation decays quadratically in terms of the step size of the grid over which the values of the cdf are tabulated. Alternatively, higher order schemes such as the Hermite interpolation might be used to reduce the interpolation bias (Hörmann and Leydold 2003).

An important step when using the inverse transform method is to fast and accurately compute the values of the cdf on the grid. It is shown in Glasserman and Liu (2010) that the bias of the method is proportional to the approximation error of the characteristic function inversion scheme. To more reliably control this source of bias, we use the Hilbert transform method of Feng and Lin (2011), which allows us to compute the probabilities almost exactly with explicit and tight error estimates. In a full-length paper (Chen, Feng, and Lin 2011), we describe how to precisely control the bias of the inverse transform method. In this paper, we mainly exhibit the effectiveness of the method.

One of the advantages of the inverse transform method is that it can be easily used together with quasi-Monte Carlo methods. Quasi-Monte Carlo methods can be attractive since they increase the convergence order from $1/2$ for standard Monte Carlo simulation to nearly 1. In Benth, Groth, and Kettler (2006), normal inverse Gaussian processes are simulated using quasi-Monte Carlo methods and the Brownian subordination construction of such processes. Quasi-Monte Carlo methods have also been applied to the simulation of variance gamma processes (Madan, Carr, and Chang 1998) in Avramidis and L'Ecuyer (2006), Ribeiro and Webber (2003), where special structures of such processes are exploited. See Fu (2007) for a review of simulating variance gamma processes. We exhibit the effectiveness of quasi-Monte Carlo methods when combined with the inverse transform method. In particular, we consider randomized quasi-Monte Carlo methods (L'Ecuyer 2004, L'Ecuyer and Lemieux 2002) where error estimates can be obtained more easily. The performance of the method is illustrated with the pricing of discrete Asian options in the CGMY model (Carr, Geman, Madan, and Yor 2002). We use discrete geometric Asian options as control variates. For geometric Asian options, we present a Hilbert transform method which can be used to price the options almost exactly. The combination of the Hilbert transform inversion of characteristic functions, quasi-Monte Carlo methods and the control variate technique proves to be very efficient. Comparison of the inverse transform method with alternative approaches as well as more options pricing applications are reported in Chen, Feng, and Lin (2011).

2   THE INVERSE TRANSFORM METHOD

Suppose we are interested in the expectation of a function f that depends on the values of a Lévy process $X = \{X_t, t \geq 0\}$ on a uniform time grid with step size Δ over a finite time horizon $[0, T]$, where $\Delta = T/d$ for some positive integer d:

$$\mathbb{E}[f(X_\Delta, \cdots, X_{d\Delta})]. \tag{1}$$

We simulate N paths of the Lévy process: $\{(X_\Delta^n, \cdots, X_{d\Delta}^n) : 1 \leq n \leq N\}$ and then estimate the above expectation by the following:

$$\frac{1}{N} \sum_{n=1}^{N} f(X_\Delta^n, \cdots, X_{d\Delta}^n).$$

Due to the independent and stationary increment properties of Lévy processes, $X_{j\Delta} - X_{(j-1)\Delta}, j = 1, 2, \cdots, d$ are i.i.d. and have the same distribution as X_Δ. It thus suffices to simulate from the characteristic function of the Lévy increment X_Δ. In the following, we first describe how the cdf of a Lévy increment can be computed fast and accurately from its characteristic function using a Hilbert transform approach. The inverse transform method for simulating the Lévy increment is then detailed.

2.1 Hilbert Transform Representation for the Cdf

Feng and Lin (2011) presents a Hilbert transform method for the inversion of characteristic functions for computing cdf's. The Hilbert transform representation can be discretized using simple rules. For a wide class of analytic characteristic functions, the method exhibits exponentially decaying discretization errors. More specifically, the Hilbert transform of an integrable function f is defined by the following Cauchy principal value integral (Stein and Weiss 1971):

$$\mathscr{H}f(x) = \frac{1}{\pi}p.v.\int_{\mathbb{R}}\frac{f(y)}{x-y}dy, \quad x \in \mathbb{R}.$$

Then the cdf $F(x)$ of a continuous distribution with an integrable characteristic function $\phi(\xi)$ can be expressed in terms of the Hilbert transform of ϕ (Theorem 1). Moreover, if the distribution admits exponential moments in an interval containing the origin, ϕ is analytic in a horizontal strip containing the real line in the complex plane. Denote the strip by $\mathscr{D}_{(d_-,d_+)} = \{z \in \mathbb{C} : \Im(z) \in (d_-,d_+)\}$ for some $d_- < 0 < d_+$, where $\Im(z)$ denotes the imaginary part of z. Further assume that ϕ has reasonable tail behaviors and integrability as below:

$$\int_{d_-}^{d_+}|\phi(\xi+iy)|dy \to 0, \quad \xi \to \pm\infty, \tag{2}$$

$$\int_{\mathbb{R}}(|\phi(\xi+id_-)|+|\phi(\xi+id_+)|)d\xi < +\infty. \tag{3}$$

Then the Hilbert transform representation for $F(x)$ can be discretized using a simple rule highly accurately:

$$F_h(x) = \frac{1}{2}+\frac{i}{2}\sum_{m=-\infty}^{\infty}e^{-ix(m-1/2)h}\frac{\phi((m-1/2)h)}{(m-1/2)\pi}, \quad h > 0.$$

The discretization error is of the order $O(\exp(-\pi d_0/h))$ and hence converges to zero exponentially in $1/h$. Here h is the discretization step size and $d_0 = 2\min(-d_-,d_+)$. The resulting infinite series is further truncated at a truncation level $M \geq 1$. Denote the resulting finite sum by $F_{h,M}(x)$:

$$F_{h,M}(x) = \frac{1}{2}+\frac{i}{2}\sum_{m=-M}^{M}e^{-ix(m-1/2)h}\frac{\phi((m-1/2)h)}{(m-1/2)\pi}, \quad h > 0, M \geq 1. \tag{4}$$

The truncation error depends on the tail behavior of the characteristic function. In many popular Lévy models used in finance, the characteristic function satisfies the following for some constants $\kappa,c,\nu > 0$:

$$|\phi(\xi)| \leq \kappa\exp(-c|\xi|^\nu), \quad \xi \in \mathbb{R}. \tag{5}$$

The exponential tail behavior of the characteristic function corresponds to the smoothness of the probability density function. When (5) is satisfied, the truncation error also decays exponentially in terms of Mh. We may thus select $h = h(M)$ so that the discretization and truncation errors are proportional:

$$h(M) = (\pi d_0/c)^{1/(1+\nu)}M^{-\nu/(1+\nu)}, \quad M \geq 1. \tag{6}$$

Then the total error decays to zero exponentially in terms of M, which represents the computational cost of approximating $F(x)$. In summary, we have the following:

Theorem 1 (Feng and Lin 2011) Let $F(x)$ and $\phi(\xi)$ be the cdf and the characteristic function of a continuous distribution. Suppose that $\phi \in L^1(\mathbb{R})$. Then

$$F(x) = \frac{1}{2}-\frac{i}{2}\mathscr{H}(e^{-i\xi x}\phi(\xi))(0).$$

If ϕ is analytic in a stripe $\mathscr{D}_{(d_-,d_+)}$ and satisfies (2) and (3), then there exists a constant C independent of h such that

$$|F(x) - F_h(x)| \leq \frac{Ce^{-\pi d_0/h}}{1 - e^{-\pi d_0/h}}, \quad h > 0,$$

where $d_0 = 2\min(-d_-, d_+)$. If ϕ also satisfies (5), then there exists $C > 0$ independent of M such that

$$|F(x) - F_{h(M),M}(x)| \leq C\exp(-c^{1/(1+v)}(\pi d_0 M)^{v/(1+v)}), \quad M \geq 1.$$

When $\mathscr{D}_{(d_-,d_+)}$ is asymmetric, one may shift the line of integration and further increase the rate of convergence. Moreover, the error estimates in Theorem 1 admit explicit expressions. Precise error control is therefore possible (Feng and Lin 2011, Chen, Feng, and Lin 2011). In derivatives pricing applications, the price process of the underlying asset is modeled by the exponential of a Lévy process. For the asset itself to be priced, certain exponential moments of the corresponding Lévy process exist. Consequently, the characteristic function is analytic (Feng and Linetsky 2008a). For many popular Lévy models, the transition probability density is smooth and the characteristic function admits exponentially decaying tails. $F_{h(M),M}(x)$ thus provides a highly accurate approximation to the cdf $F(x)$. This is illustrated in the numerical section.

Finally, due to the e^{-ixmh} term, the fast Fourier transform can be used to compute multiple values of the cdf simultaneously very efficiently. More specifically, over an arbitrary interval $[x_0, x_{2M}]$, given a uniform grid $\{x_0, x_0 + \eta, \cdots, x_0 + 2M\eta = x_{2M}\}$, the computation of $\{F(x_0 + m\eta), m = 0, 1, \cdots, 2M\}$ can be reduced to a Toeplitz matrix vector multiplication which can be implemented using the fast Fourier transform with $O(M\log M)$ operations only. See the appendices of Feng and Linetsky (2008a), Feng and Linetsky (2008b) for more details. We would like to remark that, to use the fast Fourier transform based on the Toeplitz matrix vector multiplication, the step sizes $h > 0$ and $\eta > 0$ can be arbitrary. This is in contrast to a standard implementation of the fast Fourier transform method, where h and η have to satisfy $h\eta = 2\pi/(2M+1)$, which can be inconvenient and restrictive.

2.2 The Inverse Transform Method

To simulate a Lévy increment with cdf $F(x)$, it suffices to simulate a uniformly distributed random variable U on $(0,1)$ and compute $F^{-1}(U)$. Since F^{-1} usually does not admit a closed form expression, we will proceed as below. First, we take a large enough interval $[x_0, x_K]$ so that $\max(F(x_0), 1 - F(x_K)) < \varepsilon$ for a given tolerance level $\varepsilon > 0$ (the determination of such an interval will be detailed below). Let $\eta = (x_K - x_0)/K$ for a positive integer K, and $x_k = x_0 + k\eta, 0 \leq k \leq K$. We compute the values $F_k = F(x_k)$ and store the pairs (x_k, F_k) in the following matrix:

$$\begin{pmatrix} x_0 & x_1 & \cdots & x_K \\ F_0 & F_1 & \cdots & F_K \end{pmatrix}. \tag{7}$$

For each generated U, which is between 0 and 1, we use binary search to find $0 \leq k \leq K - 1$ so that $F_k \leq U < F_{k+1}$. $F^{-1}(U)$ is then approximated by the following linear interpolation:

$$F^{-1}(U) \approx x_k + \frac{x_{k+1} - x_k}{F_{k+1} - F_k}(U - F_k).$$

If $0 < U < F_0$ or $F_K \leq U < 1$, we may solve the following equation numerically using a standard root finding algorithm such as the Newton-Raphson method to obtain an approximation to $F^{-1}(U)$:

$$F(x) = U.$$

However, when ε is small, it is rare to get $0 < U < F_0$ or $F_K \leq U < 1$. For instance, in our numerical example, we take $\varepsilon = 10^{-8}$. It takes on average $1/(2\varepsilon)$, or 50 million, iterations to get $0 < U < F_0$ or

$F_K \leq U < 1$. Therefore, one may simply use $F^{-1}(U) \approx x_0$ if $0 < U < F_0$ or $F^{-1}(U) \approx x_K$ if $F_K \leq U < 1$. We will follow this in our implementation. We also would like to remark that it takes roughly $\log_2(K)$ iterations to locate the interval that contains U. For convenience, in our implementation, we let K be a power of 2. Then it takes exactly $\log_2(K)$ iterations to locate the interval through a binary search.

As for determining x_0 and x_K, we denote μ and σ the expected value and the standard deviation of the distribution $F(x)$. They can be computed from the given characteristic function ϕ as follows:

$$\mu = -i\phi'(0), \quad \sigma = ((\phi'(0))^2 - \phi''(0))^{1/2},$$

where ϕ' and ϕ'' are the first and second order derivatives of ϕ. We then define x_0 and x_K as follows:

$$x_0 = \max(\mu + j\sigma : F(\mu + j\sigma) < \varepsilon), \quad x_K = \min(\mu + j\sigma : 1 - F(\mu + j\sigma) < \varepsilon).$$

Since for any integer j, $F(\mu + j\sigma)$ can be computed very fast and accurately, it is usually very easy to compute x_0 and x_K defined above. One starts with $F(x)$ for $x = \mu$, and then keep increasing or decreasing x by an amount of σ until x_0 or x_K is obtained. The above procedure is very simple. It is possible to determine x_0 and x_K for any given bias tolerance level. This will be shown in Chen, Feng, and Lin (2011).

We recognize that the above interpolation procedure will introduce bias. It is shown in Glasserman and Liu (2010) that the bias is of the order $O(\eta^2)$. One may thus select η accordingly for a required level of accuracy. Additional bias may be introduced if the probabilities in (7) are not computed accurately. However, due to the exponential converging nature of the scheme (4), we are able to compute the probabilities in (7) almost exactly. Moreover, this needs to be done once only at the very beginning of the simulation procedure, and can usually be done fast with the help of the fast Fourier transform, when necessary. Therefore, when implementing the above method, we are mainly concerned about the interpolation bias. In the numerical section, we exhibit the selection of η and its impact on the quality of the simulation outcome. In Chen, Feng, and Lin (2011), we further show how to determine η for any given bias tolerance level.

2.3 Randomized Quasi-Monte Carlo Methods

One advantage of the inverse transform method is that it can be easily used together with quasi-Monte Carlo methods. Suppose we want to compute an expectation which can be reduced to the following form for d i.i.d. uniform random variables on $(0,1)$, $U^1, \cdots, U^d$:

$$\mathbb{E}[f(F_1^{-1}(U^1), \cdots, F_d^{-1}(U^d))], \tag{8}$$

where F_j^{-1}, $1 \leq j \leq d$, are certain inverse cumulative distribution functions. For example, we have seen that the expectation (1) reduces to the above. When estimating the expectation above using quasi-Monte Carlo methods, instead of using i.i.d. $d-$dimensional uniform variables, a deterministic sequence $\{x_1, \cdots, x_{N_0}\}$ in $[0,1]^d$ is used and the following estimation is obtained:

$$\frac{1}{N_0} \sum_{n=1}^{N_0} f(F_1^{-1}(x_n^1), \cdots, F_d^{-1}(x_n^d)), \tag{9}$$

where $(x_n^1, \cdots, x_n^d) = x_n \in [0,1]^d$, $1 \leq n \leq N_0$. In contrast to the characteristic $1/2$ order of convergence for Monte Carlo methods, the convergence of quasi-Monte Carlo approximation may be accelerated to $O(1/N_0^{1-\varepsilon})$ for an arbitrarily small $\varepsilon > 0$. This could be attractive in applications.

On the other hand, we also note that while error estimates are rather straightforward to obtain for Monte Carlo methods, it is harder to obtain reliable error estimates for quasi-Monte Carlo methods. This has motivated randomized quasi-Monte Carlo methods, which effectively turn quasi-Monte Carlo methods into variance reduction techniques. One way to randomize a quasi-Monte Carlo method is to use the so called digital shift (L'Ecuyer 2004, L'Ecuyer and Lemieux 2002). More specifically, for a given quasi-Monte

Carlo sequence $\{x_1, \cdots, x_{N_0}\}$ in $[0,1]^d$, we generate d i.i.d. uniform random variables $U^1, \cdots, U^d$, and then use them to generated the following shifted sequence $\{x_1^U, \cdots, x_{N_0}^U\}$:

$$x_n^U = (x_n^1 \oplus U^1, \cdots, x_n^d \oplus U^d), \quad 1 \le n \le N_0,$$

where $\oplus$ refers to the exclusive-or operator. From this shifted sequence, we obtain an approximation of the form (9). When the above procedure is repeated L times, we obtain L i.i.d. variables. The sample mean of these L variables is then used to estimate the expectation in (8). The standard error computed from these L variables provides an estimate of the error of the randomized quasi-Monte Carlo approximation. In total, LN_0 replicates have been generated to obtain the above estimate. Finally, in our implementation, we have used Sobol's sequence to generate $\{x_1, \cdots, x_{N_0}\}$, which has been shown to be effective in financial applications (Glasserman 2004).

3 DISCRETE ASIAN OPTIONS IN LÉVY MODELS

We illustrate the method by considering the pricing of discretely monitored Asian options. We assume a geometric Lévy model in a given risk neutral measure, where the asset price is governed by

$$S_t = S_0 e^{X_t}.$$

Here S_0 is the current asset price, and $\{X_t, t \ge 0\}$ is a Lévy process starting at the origin at time 0. Consider a fixed strike Asian call option with maturity T and strike price K. The payoff of such an option is given by

$$\max(0, A_T - K),$$

where A_T is the average asset price over the time period $[0, T]$. We assume that the average asset price is computed based on the asset price monitored discretely on a uniform grid of $[0, T]$ with monitoring interval $\Delta = T/d$. Here $d > 0$ is the number of monitoring intervals. That is,

$$A_T = \frac{1}{d+1} \sum_{k=0}^{d} S_{k\Delta}.$$

Then, from the risk neutral pricing theory, the price of the call is given by

$$V = e^{-rT} \mathbb{E}[\max(0, A_T - K)],$$

where r is the continuously compounded risk free interest rate.

3.1 Geometric Asian Options

Most traded Asian options assume arithmetic averaging described above. Closed form solutions usually do not exist for such contracts. However, if the average is computed in the geometric sense, one may obtain a semi-analytic expression for the option price that can be computed very fast and accurately. They serve as excellent control variates for the pricing of arithmetic Asian options. The price of a discretely monitored geometric Asian call option with maturity T and strike K is given by the following:

$$V^g = e^{-rT} \mathbb{E}[\max(0, A_T^g - K)], \quad A_T^g = \left(\Pi_{k=0}^d S_{k\Delta}\right)^{1/(d+1)}.$$

Let $Y = \ln(A_T^g/S_0) = \frac{1}{d+1} \sum_{k=0}^{d} X_{k\Delta}$. We further rewrite Y in terms of independent Lévy increments (recall that $X_0 = 0$):

$$Y = \frac{1}{d+1} \sum_{k=1}^{d} k(X_{(d-k+1)\Delta} - X_{(d-k)\Delta}).$$

Denote the characteristic function of the Lévy process by $\phi_t(\xi) = \mathbb{E}[e^{i\xi X_t}]$. Then for any $1 \le k \le d$, the characteristic function of the Lévy increment $X_{(d-k+1)\Delta} - X_{(d-k)\Delta}$ is given by

$$\mathbb{E}[e^{i\xi(X_{(d-k+1)\Delta} - X_{(d-k)\Delta})}] = \phi_\Delta(\xi).$$

Consequently, the characteristic function of Y is given by

$$\phi_Y(\xi) = \mathbb{E}[e^{i\xi Y}] = \Pi_{k=1}^d \phi_\Delta\left(\frac{k\xi}{d+1}\right).$$

Therefore,

$$
\begin{aligned}
V^g &= e^{-rT}\mathbb{E}[(S_0 e^Y - K)\mathbf{1}_{\{Y>\ln(K/S_0)\}}] \\
&= e^{-rT}S_0\mathbb{E}[e^Y\mathbf{1}_{\{Y>\ln(K/S_0)\}}] - Ke^{-rT}\mathbb{P}(Y > \ln(K/S_0)).
\end{aligned}
\tag{10}
$$

Obviously, the second term above can be approximated using (4) highly accurately:

$$\mathbb{P}(Y > \ln(K/S_0)) \approx \frac{1}{2} - \frac{i}{2}\sum_{m=-M}^{M} e^{-i\ln(K/S_0)(m-1/2)h}\frac{\phi_Y((m-1/2)h)}{(m-1/2)\pi}. \tag{11}$$

Using a standard measure change technique, the first term can also be turned into a probability and hence admits a Hilbert transform representation. The following is shown in Feng and Lin (2011):

Theorem 2 (Feng and Lin 2011) Let ϕ be the characteristic function of a continuous random variable X. Suppose that ϕ is analytic in a strip $\mathscr{D}_{(d_-,d_+)}$, $\beta \in (d_-, d_+)$, and $\phi(\cdot + i\beta) \in L^1(\mathbb{R})$, then

$$\mathbb{E}\left[e^{-\beta X}\mathbf{1}_{\{X\le\theta\}}\right] = \frac{\phi(i\beta)}{2} - \frac{i}{2}\mathscr{H}(e^{-i\xi\theta}\phi(\xi+i\beta))(0).$$

If ϕ satisfies (2) and (3), we have the following approximation and error estimate:

$$F_h^{\beta,\theta} = \frac{\phi(i\beta)}{2} + \frac{i}{2}\sum_{m=-\infty}^{\infty} e^{-i\theta(m-1/2)h}\frac{\phi((m-1/2)h+i\beta)}{(m-1/2)\pi},$$

$$\left|\mathbb{E}\left[e^{-\beta X}\mathbf{1}_{\{X\le\theta\}}\right] - F_h^{\beta,\theta}\right| \le \frac{Ce^{-\pi d_0/h}}{1 - e^{-\pi d_0/h}}, \quad h > 0,$$

where $d_0 = 2\min(\beta - d_-, d_+ - \beta)$ and $C > 0$ is a constant independent of h. If $\phi(\cdot + i\beta)$ further satisfies (5), $h = h(M)$ is selected according to (6), and $F_{h,M}^{\beta,\theta}$ is the finite sum approximation of $F_h^{\beta,\theta}$ with truncation level $M \ge 1$, then

$$\left|\mathbb{E}\left[e^{-\beta X}\mathbf{1}_{\{X\le\theta\}}\right] - F_{h,M}^{\beta,\theta}\right| \le C\exp(-c^{1/(1+v)}(\pi d_0 M)^{v/(1+v)}), \quad M \ge 1,$$

for some constant C independent of M.

Notice that

$$\mathbb{E}\left[e^{-\beta X}\mathbf{1}_{\{X>\theta\}}\right] = \phi(i\beta) - \mathbb{E}\left[e^{-\beta X}\mathbf{1}_{\{X\le\theta\}}\right].$$

Taking $\beta = -1$ and $\theta = \ln(K/S_0)$, we obtain the following approximation to the first term in (10):

$$\mathbb{E}[e^Y\mathbf{1}_{\{Y>\ln(K/S_0)\}}] \approx \frac{\phi_Y(-i)}{2} - \frac{i}{2}\sum_{m=-M}^{M} e^{-i\ln(K/S_0)(m-1/2)h}\frac{\phi_Y((m-1/2)h-i)}{(m-1/2)\pi}. \tag{12}$$

For those Lévy models with characteristic functions that decay exponentially, the pricing error of the above scheme decreases to zero exponentially in terms of the truncation level M. Consequently, the price of the geometric Asian option can be computed highly accurately. In the example considered in Section 4, we obtain the geometric Asian option price almost exactly in a very small amount of time.

3.2 Control Variates

To compute $V = e^{-rT} \mathbb{E}[\max(0, A_T - K)]$, we simulate d Lévy increments and hence a Lévy path $\{X_{\Delta}^n, \cdots,$ $X_{d\Delta}^n\}$ and compute $V_n = \max(0, A_T - K)$ and $V_n^g = \max(0, A_T^g - K)$. Moreover, the discrepancy between V_n^g and $\mathbb{E}[\max(0, A_T^g - K)]$ can be used to adjust V_n in the following way so that the resulting quantity is closer to $\mathbb{E}[\max(0, A_T - K)]$:

$$V_n^b := V_n + b(\mathbb{E}[\max(0, A_T^g - K)] - V_n^g).$$

Here b is a constant to be determined. We repeat the above for $n = 1, 2, \cdots, N$, and obtain the following estimator to $\mathbb{E}[\max(0, A_T - K)]$:

$$\bar{V}_N^b := \frac{1}{N} \sum_{n=1}^{N} V_n^b.$$

By the law of large numbers, $\bar{V}_N^b$ converges to $\mathbb{E}[\max(0, A_T - K)]$. Moreover, let σ_a denote the standard deviation of $\max(0, A_T - K)$, σ_g the standard derivation of $\max(0, A_T^g - K)$, and ρ the correlation coefficient of the two random variables. When b is selected according to $b = \rho \sigma_a / \sigma_g$, the variance of $\bar{V}_N^b$ is reduced to

$$\text{var}(\bar{V}_N^b) = (1 - \rho^2) \text{var}(\bar{V}_N),$$

where $\bar{V}_N = \frac{1}{N} \sum_{n=1}^{N} V_n$ is the estimator of $\mathbb{E}[\max(0, A_T - K)]$ without using a control variate. When ρ is large, the performance improvement of the above scheme over a standard scheme without control variates is significant. Since the true value of b is usually unknown to us, an estimate of b is obtained by simulating a predetermined number (e.g., 1000) of pairs of (V_n, V_n^g) and replacing the correlation coefficient and standard deviations by sample correlation coefficient and sample standard deviations.

4 NUMERICAL RESULTS

We illustrate the method by considering the pricing of an arithmetic Asian call option in the CGMY model (Carr, Geman, Madan, and Yor 2002). Computations are done using C++ in Windows XP on a Lenovo laptop T61p with Intel Core 2 Duo 2.5GHz CPU and 3GB RAM. The maturity of the contract is 6 months ($T = 0.5$). The average is computed based on either monthly monitoring ($d = 6$) or weekly monitoring ($d = 26$). The current asset price and the strike price are $S_0 = K = 100$. The risk free interest rate is $r = 5\%$. The underlying asset does not pay dividends. The parameters of the CGMY model are the same as those in Feng and Linetsky (2008a):

$$\hat{C} = 4, \hat{G} = 50, \hat{M} = 60, \hat{Y} = 0.7.$$

The explicit expression of the transition probability density of a general CGMY process is not available. However, the characteristic function of such a process $X = \{X_t, t \geq 0\}$ admits a simple form and is given by

$$\phi_t(\xi) = \exp(i\hat{\mu}t\xi - t\hat{C}\Gamma(-\hat{Y})(\hat{M}^{\hat{Y}} - (\hat{M} - i\xi)^{\hat{Y}} + \hat{G}^{\hat{Y}} - (\hat{G} + i\xi)^{\hat{Y}})),$$

where $\hat{\mu} = r - \hat{C}\Gamma(-\hat{Y})((\hat{M} - 1)^{\hat{Y}} - \hat{M}^{\hat{Y}} + (\hat{G} + 1)^{\hat{Y}} - \hat{G}^{\hat{Y}})$ is the drift of the CGMY process, and $\Gamma(\cdot)$ is the gamma function.

4.1 Computing the Probabilities

We first compute the endpoints of the interval $[x_0, x_K]$ on which the values of the cdf of the Lévy increment are tabulated. We assume $\varepsilon = 10^{-8}$. Note that the expected value and the standard deviation of X_T can be easily computed to be (see formulas in Section 2.2)

$$\mathbb{E}[X_{0.5}] = 0.015, \quad \text{stdev}(X_{0.5}) = 0.141.$$

Using the independent and stationary increment properties of Lévy processes, the expected value and the standard deviation of the Lévy increment X_Δ with d monitoring intervals on $[0, 0.5]$ are given by

$$\mu_\Delta = \mathbb{E}[X_\Delta] = \frac{1}{d}\mathbb{E}[X_{0.5}], \quad \sigma_\Delta = \text{stdev}(X_\Delta) = \frac{1}{\sqrt{d}}\text{stdev}(X_{0.5}).$$

It can be easily checked that, for any $\Delta > 0$, ϕ_Δ is analytic in a horizontal strip and satisfies the exponential tail condition (5) with

$$d_- = -\hat{M}, \quad d_+ = \hat{G}, \quad v = \hat{Y}, \quad c_\Delta = 2\Delta\hat{C}|\Gamma(-\hat{Y})\cos(\pi\hat{Y}/2)|. \tag{13}$$

For $d = 6$ and $\Delta = T/6$, we compute the probabilities $F_\Delta(\mu_\Delta + j\sigma_\Delta)$ for $j = 0, -1, \cdots, -8$ and find that

$$F_\Delta(\mu_\Delta - 8\sigma_\Delta) < \varepsilon = 10^{-8}, \quad \mu_\Delta - 8\sigma_\Delta \approx -0.458.$$

Similarly, we compute $F_\Delta(\mu_\Delta + j\sigma_\Delta)$ for $j = 1, \cdots, 7$ to find that

$$1 - F_\Delta(\mu_\Delta + 7\sigma_\Delta) < \varepsilon = 10^{-8}, \quad \mu_\Delta + 7\sigma_\Delta \approx 0.405.$$

We thus take $[x_0, x_K] = [-0.458, 0.405]$ when $d = 6$ with monthly monitoring. Similarly, for weekly monitoring with $d = 26$, we find the interval to be $[x_0, x_K] = [-0.331, 0.277]$. In our implementation, we take $M = 150$ so that the probabilities in (7) are accurate up to 12 digits after the decimal point. The purpose is to focus on the error due to the interpolation and the simulation itself. When $K = 16$ (corresponding to an interpolation bias of about 1 cent), it takes 0.001 second to produce all the $K + 1$ probabilities. When $K = 128$ (corresponding to an interpolation bias of about 0.01 cent), it takes 0.007 second to produce all 129 probabilities. In this example, since K does not need to be very large to guarantee a practical accuracy, we didn't use the fast Fourier transform. However, if in a particular application, K has be to larger, the fast Fourier transform can be used to greatly reduce the computational time (Chen, Feng, and Lin 2011).

4.2 Computing Geometric Asian Option Prices

When the corresponding geometric Asian option is used as a control variable, we need to compute the exact geometric Asian option price. This again can be computed highly accurately and fast using the method described in Section 3.1. We first note that ϕ_Y defined in Section 3.1 is again analytic in a horizontal strip and satisfies the exponential tail condition (5) with

$$d_-(Y) = d_-(d+1)/d, \quad d_+(Y) = d_+(d+1)/d, \quad v = \hat{Y}, \quad c(Y) = c_\Delta \sum_{k=1}^{d} k^v/(d+1)^v, \tag{14}$$

where $d_\pm$ and c_Δ are given in (13). We use (11) and (12) to compute the two terms in (10). By Theorems 1 and 2, we select $h = h(M)$ according to (6), (14) and $d_0 = 2\min(-1 - d_-(Y), d_+(Y))$. The geometric Asian option prices can be computed with an accuracy of up to 12 digits after the decimal point with only $M = 50$. For the monthly monitored contract ($d = 6$), the price is 3.604561644590 and it takes 0.001 second. For the weekly monitored contract ($d = 26$), the price is 3.698573792716 and it takes 0.005 second.

4.3 Interpolation Bias

In the following, we assume that the errors contained in the probabilities in (7) and the geometric Asian call prices are negligible and focus on the interpolation bias. For $K = 16, 32, 64, 128$, we increase the simulation sample size and observe the pricing error. Benchmark prices for the arithmetic Asian options have been computed using the Fourier transform method (Benhamou 2002, Carverhill and Clewlow 1990) to be $V = 3.71933798$ for monthly monitoring and $V = 3.80167775$ for weekly monitoring. We use the

Monte Carlo scheme with control variate (MC-CV). This scheme is fast and accurate. Very soon we observe that the pricing error does not decrease any more by increasing the sample size. That is when the interpolation bias starts to dominate. We consider the case with monthly monitoring ($d = 6$). In Table 1, the first column refers to sample sizes that are used. Columns with label "Error" are absolute pricing errors (the difference between the Monte Carlo estimate and the benchmark price). Computational times for the Monte Carlo scheme will be reported later when different methods are compared. Columns with label "SE" refer to the standard error of the Monte Carlo estimate. Suppose N replicates $\{V_1^b, V_2^b, \cdots, V_N^b\}$ are generated. The SE is defined by

$$\text{SE}(V_1^b, V_2^b, \cdots, V_N^b) = \sqrt{\frac{\sum_{n=1}^{N}(V_n^b - \bar{V}_N^b)^2}{N(N-1)}}, \quad \bar{V}_N^b = \frac{1}{N}\sum_{n=1}^{N} V_n^b.$$

Note that in Table 1, SE is not necessarily a good indicator of the absolute pricing error since the interpolation bias is not accounted for in its definition. On the other hand, when SE is very small, the pricing error is mainly due to the interpolation bias. We are thus able to observe the interpolation bias by looking at the bottom of the table. We find that when $K = 16$ ($\eta = (X_K - X_0)/K = (0.405 + 0.458)/16 \approx 0.054$), the interpolation bias is about 9×10^{-3}. This implies an interpolation bias of the form $c\eta^2 \approx 3\eta^2$ in this example. It can be seen from the numbers in bold that when K doubles, the interpolation bias roughly decreases by 4 times, which is consistent with Glasserman and Liu (2010). In particular, in this example, a fairly small K will be sufficient for practical purpose.

Table 1: Interpolation bias ($d = 6$). Error: absolute pricing error; SE: standard error, it should not be used as an indicator of the pricing error unless the interpolation bias is negligible.

N	$K = 16$		$K = 32$		$K = 64$		$K = 128$	
	SE	Error	SE	Error	SE	Error	SE	Error
1000	3.4E-3	4.0E-3	3.9E-3	1.1E-2	3.7E-3	5.5E-3	3.7E-3	3.2E-3
4000	2.0E-3	1.0E-2	1.8E-3	1.4E-3	1.8E-3	2.1E-3	1.8E-3	1.6E-3
16000	9.8E-4	8.4E-3	9.3E-4	4.3E-3	8.6E-4	7.0E-4	8.4E-4	2.7E-4
64000	4.9E-4	9.5E-3	4.5E-4	2.4E-3	4.3E-4	2.8E-4	4.3E-4	2.2E-4
256000	2.5E-4	9.4E-3	2.2E-4	2.3E-3	2.2E-4	6.5E-4	2.2E-4	2.2E-4
1024000	1.2E-4	8.9E-3	1.1E-4	2.0E-3	1.1E-4	8.0E-4	1.1E-4	1.0E-4
4096000	6.1E-5	8.9E-3	5.5E-5	2.1E-3	5.4E-5	6.6E-4	5.3E-5	1.7E-4
16384000	3.1E-5	**8.7E-3**	2.8E-5	**2.3E-3**	2.7E-5	**6.2E-4**	2.7E-5	**1.4E-4**

4.4 Simulation Errors

Finally, we investigate the simulation error. To have a clear comparison, we make the interpolation bias small by taking a large $K = 1024$. Based on the previous section, with this K, the interpolation bias is about 2×10^{-6} and is hence negligible compared to simulation errors. We compare the standard Monte Carlo scheme (MC), the randomized quasi-Monte Carlo scheme (RQMC), the Monte Carlo scheme with control variate (MC-CV), and the randomized quasi-Monte Carlo scheme with control variate (RQMC-CV). For MC-CV, we generate 1000 pairs of arithmetic and geometric Asian call payoffs to estimate the parameter b. For simplicity, these 1000 pairs are not reused. For RQMC-CV, since the number of batches L is typically small (between 5-25 according to L'Ecuyer (2004)), we directly use the L pairs of arithmetic and geometric Asian option prices that we obtain, estimate the parameter b, and adjust the arithmetic Asian option prices accordingly. For each scheme, we increase the sample size N and record the standard error ("SE"), absolute pricing error ("Error"), and computational time ("CPU"). For a better comparison, the computational time does not include the time to compute the probabilities in (7) and geometric Asian option prices (see Sections 4.1 and 4.2 instead). For MC and MC-CV, $N = 1000, 4000, \cdots, 1024000$ refers

Table 2: Discrete Asian call options in the CGMY model. MC: standard Monte Carlo; RQMC: randomized quasi-Monte Carlo; MC-CV: Monte Carlo with control variate; RQMC-CV: randomized quasi-Monte Carlo with control variate; N: total number of Lévy paths generated; Error: absolute pricing error; CPU: computational time in seconds (not including the time for computing the probabilities in (7) and the geometric Asian option prices; see Sections 4.1 and 4.2); SE: standard error.

		MC			RQMC			MC-CV			RQMC-CV		
	N	SE	Error	CPU	SE	Error	CPU	SE	Error	CPU	SE	Error	CPU
$d=6$	1000	2E-1	2E-1	0.002	4E-2	6E-3	0.001	4E-3	2E-3	0.003	2E-3	1E-4	0.004
	4000	8E-2	2E-1	0.004	2E-2	3E-2	0.004	2E-3	2E-3	0.006	5E-4	1E-3	0.006
	16000	4E-2	2E-2	0.012	7E-3	2E-3	0.014	9E-4	5E-4	0.015	2E-4	2E-4	0.016
	64000	2E-2	2E-3	0.046	2E-3	7E-4	0.051	4E-4	3E-4	0.053	1E-4	1E-4	0.055
	256000	1E-2	7E-3	0.177	5E-4	4E-4	0.199	2E-4	4E-4	0.206	4E-5	9E-6	0.214
	1024000	5E-3	4E-3	0.707	2E-4	9E-5	0.795	1E-4	9E-5	0.818	3E-5	9E-6	0.867
$d=26$	1000	2E-1	2E-1	0.004	1E-1	6E-2	0.005	3E-3	3E-3	0.007	2E-3	3E-3	0.006
	4000	8E-2	3E-2	0.013	2E-2	2E-2	0.013	2E-3	2E-3	0.016	8E-4	5E-4	0.016
	16000	4E-2	4E-3	0.046	1E-2	1E-2	0.049	8E-4	1E-3	0.051	6E-4	3E-4	0.055
	64000	2E-2	3E-2	0.176	2E-3	2E-3	0.195	4E-4	5E-4	0.196	3E-4	2E-4	0.211
	256000	1E-2	1E-3	0.698	1E-3	1E-3	0.771	2E-4	9E-5	0.780	1E-4	3E-4	0.834
	1024000	5E-3	2E-3	2.853	6E-4	8E-4	3.046	1E-4	4E-5	3.097	4E-5	5E-5	3.353

to the sample size (the total number of Lévy paths generated) and the SE is computed as in the previous section. For RQMC and RQMC-CV, we use $L = 10$ batches and for each batch we use a sequence of length $N_0 = 100, 400, \cdots, 102400$. Therefore, $N = LN_0$ again refers to the total number of Lévy paths that are generated. The SE is computed from the L estimates we obtain.

We observe that the computational times are close among different methods (all of them use the same number of Lévy paths). The standard Monte Carlo scheme is less accurate in general. The randomized quasi-Monte Carlo method greatly increases the convergence, in particular for $d = 6$. The improvement in the performance is less significant for $d = 26$. It is well known that quasi-Monte Carlo methods worsen when the dimension of the problem increases. Dimension reduction schemes such as bridge sampling (Avramidis and L'Ecuyer 2006, Fu 2007, Ribeiro and Webber 2003) might be used to reduce the effective dimension of the problem and can be implemented for Lévy processes where such schemes apply. We also observe that when geometric Asian options are used as control variates, the performance of the simple Monte Carlo scheme greatly improves. The MC-CV scheme is even better than the RQMC scheme. Finally, the RQMC scheme with control variate outperforms all other methods.

In summary, the inverse transform method combined with quasi-Monte Carlo approximation and variance reduction techniques proves to be very fast and accurate, at least for the discrete Asian options pricing example considered in this paper. With less than 1 second, one obtains an accuracy (measured by SE in Table 2) of about $3 \times 10^{-5} \sim 4 \times 10^{-5}$ for the monthly or weekly monitored Asian options we consider.

ACKNOWLEDGMENTS

REFERENCES

Asmussen, S., and J. Rosiński. 2001. "Approximation of small jumps of Lévy processes with a view towards simulation". *Journal of Applied Probability* 38 (2): 482–493.

Avramidis, A. N., and P. L'Ecuyer. 2006. "Efficient Monte Carlo and quasi-Monte Carlo option pricing under the variance-gamma model". *Management Science* 52 (12): 1930–1944.

Barndorff-Nielsen, O. E. 1998. "Process of Normal Inverse Gaussian Type". *Finance and Stochastics* 2 (1): 41–68.

Benhamou, E. 2002. "Fast Fourier transform for discrete Asian options". *Journal of Computational Finance* 6 (1): 49–68.

Benth, F., M. Groth, and P. Kettler. 2006. "A quasi-Monte Carlo algorithm for the normal inverse Gaussian distribution and valuation of financial derivatives". *International Journal of Theoretical and Applied Finance* 9:843–867.

Bondesson, L. 1982. "On simulation from infinitely divisible distributions". *Advances in Applied Probability* 14 (4): 855–869.

Boyarchenko, S., and S. Levendorskiĭ. 2002. *Non-Gaussian Merton-Black-Scholes Theory*. Singapore: World Scientific.

Carr, P., H. Geman, D. B. Madan, and M. Yor. 2002. "The fine structure of asset returns: an empirical investigation". *Journal of Business* 75 (2): 305–332.

Carverhill, A. P., and L. J. Clewlow. 1990. "Flexible convolution". *Risk* 3:25 – 29.

Chen, Z., L. Feng, and X. Lin. 2011. "Simulating Lévy processes from their characteristic functions and financial applications". *Working paper, University of Illinois at Urbana-Champaign*.

Cont, R., and P. Tankov. 2004. *Financial Modelling with Jump Processes*. Boca Raton, Florida: Chapman & Hall/CRC.

Devroye, L. 1981. "On the computer generation of random variables with a given characteristic function". *Computers & Mathematics with Applications* 7 (6): 547–552.

Feng, L., and X. Lin. 2011. "Inverting analytic characteristic functions and financial applications". *Working paper, University of Illinois at Urbana-Champaign*.

Feng, L., and V. Linetsky. 2008a. "Pricing discretely monitored barrier options and defaultable bonds in Lévy process models: a fast Hilbert transform approach". *Mathematical Finance* 18 (3): 337–384.

Feng, L., and V. Linetsky. 2008b. "Pricing options in jump-diffusion models: an extrapolation approach". *Operations Research* 56 (2): 304–325.

Fu, M. C. 2007. "Variance-gamma and Monte Carlo". In *Advances in Mathematical Finance*, edited by M. C. Fu, R. A. Jarrow, J. Yen, and R. J. Elliott, 21–35. Boston: Birkhäuser.

Glasserman, P. 2004. *Monte Carlo Methods in Financial Engineering*. Berlin, Germany: Springer-Verlag.

Glasserman, P., and Z. Liu. 2007, December. "Sensitivity estimates from characteristic functions". In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 932–940. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Glasserman, P., and Z. Liu. 2010. "Sensitivity estimates from characteristic functions". *Operations Research* 58 (6): 1611–1623.

Hörmann, W., and J. Leydold. 2003. "Continuous random variate generation by fast numerical inversion". *ACM Transations on Modeling and Computer Simulation* 13:347–362.

L'Ecuyer, P. 2004, December. "Quasi-Monte Carlo methods in finance". In *Proceedings of the 2004 Winter Simulation Conference*, edited by R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

L'Ecuyer, P., and C. Lemieux. 2002. "Recent advances in randomized quasi-Monte Carlo methods". In *Modeling Uncertainty: an Examination of Stochastic Theory, Methods, and Applications*, edited by M. Dror, P. L'Ecuyer, and F. Szidarovszki, 419–474. Boston: Kluwer Academic Publishers.

Madan, D. B., P. P. Carr, and E. C. Chang. 1998. "The variance gamma process and option pricing". *European Finance Review* 2 (1): 79–105.

Ribeiro, C., and N. Webber. 2003. "Valuaing path-dependent options in the variance-gamma model by Monte Carlo with a gamma bridge". *Journal of Computational Finance* 7:81–100.

Rosiński, J. 2001. "Series representations of Lévy processes from the perspective of point processes". In *Lévy Processes – Theory and Applications*, edited by O. Barndorff-Nielsen, T. Mikosch, and S. Resnick, 401–415. Boston: Birkhäuser.

Rydberg, T. H. 1997. "The normal inverse Gaussian Lévy process: simulation and approximation". *Stochastic Models* 13 (4): 887–910.

Schoutens, W. 2003. *Lévy Processes in Finance: Pricing Financial Derivatives*. Hoboken, New Jersey: John Wiley & Sons.

Shaw, W. T., and J. McCabe. 2009. "Monte Carlo sampling given a characteristic function: quantile mechanics in momentum space". *Available from http://arxiv.org/abs/0903.1592*.

Stein, E. M., and G. Weiss. 1971. *Introduction to Fourier Analysis on Euclidean Spaces*. Princeton, New Jersey: Princeton University Press.

AUTHOR BIOGRAPHIES

ZISHENG CHEN is a Ph.D. student in the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign.

LIMING FENG is an assistant professor in the Department of Industrial and Enterprise Systems Engineering at the University of Illinois at Urbana-Champaign. His research interests are in financial engineering, applied probability and stochastic modeling. He can be reached at fenglm@illinois.edu.

XIONG LIN obtained his Ph.D. in 2010 from the Department of Mathematics at the University of Illinois at Urbana-Champaign.