Simulating Lévy processes from their characteristic functions and financial applications*

Zisheng Chen† **Liming Feng**‡ **Xiong Lin**§

This version: 7/30/2011, ACM Transactions on Modeling and Computer Simulation

Abstract

The simulation of a discrete sample path of a Lévy process reduces to simulating from the distribution of a Lévy increment. For a general Lévy process with exponential moments, the inverse transform method proposed in Glasserman and Liu 2010 [24] is reliable and efficient. The values of the cumulative distribution function (cdf) are computed by inverting the characteristic function and tabulated on a uniform grid. The inverse of the cumulative distribution function is obtained by linear interpolation. In this paper, we apply a Hilbert transform method for the characteristic function inversion. The Hilbert transform representation for the cdf can be discretized using a simple rule highly accurately. Most importantly, the error estimates admit explicit and computable expressions, which allow us to compute the cdf to any desired accuracy. We present an explicit bound for the estimation bias in terms of the range of the grid where probabilities are tabulated, the step size of the grid, and the approximation error for the probabilities. The bound can be computed from the characteristic function directly and allows one to determine the size and fineness of the grid and numerical parameters for evaluating the Hilbert transforms for any given bias tolerance level in one dimensional problems. For multidimensional problems, we present a procedure for selecting the grid and the numerical parameters that is shown to converge theoretically and works well practically. The inverse transform method is attractive not only for Lévy processes that are otherwise not easy to simulate, but also for processes with special structures that could be simulated in different ways. The method is very fast and accurate when combined with quasi-Monte Carlo schemes and variance reduction techniques. The main results we derived are not limited to Lévy processes and can be applied to simulating from tabulated cumulative distribution functions in general and characteristic functions in our analytic class in particular.

Keywords:

Lévy process, randomized quasi-Monte Carlo method, inverse transform method, Hilbert transform, analytic characteristic function, discrete Asian option, control variates

*This research was partially supported by the National Science Foundation under grants CMMI-0927367, CMMI-1029846

†Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, *E-mail:* zchen27@illinois.edu.

‡Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, *E-mail:* fenglm@illinois.edu.

§Department of Mathematics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, *E-mail:* afxlin9@gmail.com.

1 Introduction

Lévy models have seen many applications, from modeling the input of water into a dam [34], to modeling the reserve of an insurance company [27], to modeling the dynamics of financial variables [10, 14, 39]. One often needs to compute numerically a quantity that is associated with the path of a Lévy process where a closed-form expression is not available. One of the popular methods is Monte Carlo simulation [2, 22]. In derivatives pricing applications, Monte Carlo methods are attractive in that they handle multidimensional problems better and often do not require significant modifications when the payoff structure of the contract changes.

Standard techniques can be used when the transition probability density or the cumulative distribution function (cdf) 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. A general Lévy process may have an infinite number of jumps, most of them being small, over an arbitrary finite time horizon, making the simulation of such a process often non-trivial. One approach is to discard these small jumps. Such an approach is accurate only when there are not too many small jumps [14]. Alternatively, under certain conditions, the small jump component of a Lévy process may be approximated by a Brownian motion. Consequently, the Lévy process is approximated by a jump diffusion process and then simulated. [3] studies conditions under which this approach is valid. One may also compute infinite series representations of a Lévy process and simulate from such a series representation [9, 37]. When the Lévy process results from a Brownian subordination, it suffices to simulate the subordinator and a standard normal random variable (see [38] for the case of simulating a normal inverse Gaussian process [5] using this approach).

In this paper, we are interested in simulating a discrete sample path of a Lévy process over a finite time horizon. This may arise from situations where only discrete values of the process are concerned (e.g., discrete lookback and Asian options considered in this paper), or we are interested in a functional of the complete path of the Lévy process that is difficult to analyze and needs to be approximated by a function of discrete values of the process. By the independent increment property of Lévy processes, to simulate a discrete Lévy sample path, it suffices to simulate independent Lévy increments. Although the transition probability density of a general Lévy process is often unknown explicitly, due to the celebrated Lévy-Khintchine formula for infinitely divisible distributions, the characteristic function of the Lévy process is often known. In such a case, our task reduces to simulating from an explicit characteristic function of a certain distribution.

One method to simulating from a given characteristic function is to approximate the quantile function by solving a non-linear integro-differential equation [40]. One may also follow [15], where a simple bound for the probability density is derived from its characteristic function and an acceptance-rejection method is used. In this approach, to generate each realization, an inverse Fourier transform integral needs to be computed. The computational cost could thus be high. Alternatively, one may invert the characteristic function, compute the cumulative distribution function and then use an inverse transform method[1]. To use the inverse transform method directly, one needs to solve a nonlinear equation where the cdf equals a uniformly distributed random variable on $(0, 1)$. It will be again computationally intensive if such a root finding

[1]The inverse transform method refers to the approach where one simulates from a distribution $F(\cdot)$ by generating $F^{-1}(U)$, where F^{-1} is the inverse of F, and U is uniformly distributed on $(0, 1)$. This is not to be confused with the inversion of the characteristic function to compute the cdf, which is another key step in our method.

procedure needs to be conducted for each realization generated and the cdf itself needs to be computed by numerically inverting its characteristic function. A much easier approach is to tabulate the numerically computed values of the cdf on a large and fine enough grid and then interpolate to find the solution of the previously mentioned equation. This is the method that is used in [23, 24], where it is shown that the bias due to linear interpolation is quadratic in η, where η is the step size of the grid over which the values of the cdf are tabulated (higher order schemes such as the Hermite interpolation might be used to reduce the interpolation bias [25]).

More specifically, when studying sensitivity estimates from characteristic functions using the likelihood ratio method, [23] and [24] investigate the estimation bias that is introduced by the above interpolation procedure and by the Fourier series method of [1] that is used to compute the probabilities through a Laplace transform inversion. The discretization of the inverse Laplace transform integral leads to an infinite series, which is then truncated. For a class of distributions that admit exponential moments, it is shown that the total bias consists of the previously mentioned quadratic term $O(\eta^2)$, a term of the form $O(\exp(-c/h))$ for some $c > 0$, where h is the discretization step size used for the inversion of the Laplace transform, and a term due to the truncation of the infinite series. This representation allows one to update the step size η of the grid, the step size h for the Laplace transform inversion and the truncation level for the infinite series so that their contributions to the total estimation bias decrease proportionally.

In this paper, we extend the above analysis by presenting an explicit bound for the total estimation bias. The bound is in terms of the known characteristic function and can be computed directly. This allows us to precisely compute the interpolation step size η, the step size h for the characteristic function inversion, the truncation level, as well as the range of the grid for any given bias tolerance level for one dimensional problems. One does not need to update these parameters and re-run the simulation. Moreover, when the bias is expressed using the big O notation as in [23, 24], one only knows the convergence rate of each component, but not the magnitude. It is thus not clear how to choose η, h, the range of the grid, and the truncation level so that their contributions to the total bias are well balanced. It is also not clear how to come up with an appropriate initial guess of these parameters. These difficulties will be eased with the explicit bound for the bias which we will present. For multidimensional problems, a precise bias control such as the above is difficult. We provide a procedure for selecting the grid and numerical parameters for characteristic function inversion that is shown to converge theoretically and works well practically.

Our analysis depends on a Hilbert transform method for the inversion of analytic characteristic functions developed in [17]. For a class of characteristic functions that are analytic in a horizontal strip containing the real axis, the Hilbert transform representation can be discretized using very simple rules highly accurately, with errors decaying exponentially in $1/h$, where h is the step size used to approximate the Hilbert transform. Moreover, the error estimates admit explicit expressions that can be computed directly. This in turn allows us to derive an explicit bound for the estimation bias for the previously discussed inverse transform method. In financial applications, an asset price is often modeled by the exponential of a certain stochastic process. For the asset price itself to be priced with a finite moment, the corresponding stochastic process must have certain exponential moments. This naturally leads to the analyticity of the characteristic function [30]. The Hilbert transform method then allows one to compute the values of the cdf to any desired accuracy level. Moreover, the Hilbert transform method is very easy to implement, and, when desirable, can be implemented using the fast Fourier transform to compute multiple values of the cdf simultaneously in a very efficient way.

One advantage of the inverse transform method for simulating a Lévy process is that it does not depend on particular structures of the Lévy process. For example, Kou's jump diffusion process [26] can be simulated by generating a Brownian motion, a Poisson process and independent double exponential jump sizes. A normal inverse Gaussian process can be simulated by generating a Brownian motion and an inverse Gaussian process. These approaches are of course highly valuable since, in contrast to the inverse transform method, they do not introduce bias due to the approximation of the distribution functions. On the other hand, the CGMY process [11] does not admit such simple structures. However, the inverse transform method only depends on the characteristic function and hence is applicable to the CGMY process. Moreover, the inverse transform method is attractive not only for processes that are otherwise difficult to simulate, but also for processes such as Kou's process and the NIG process that could be simulated in other ways. Our numerical examples show that for practical levels of accuracy, the inverse transform method could be a few times faster.

The inverse transform method can be easily coupled with quasi-Monte Carlo methods. Quasi-Monte Carlo methods can be attractive since they increase the order of convergence from $1/2$ for standard Monte Carlo methods to nearly 1. In [7], 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 [31] in [4, 36], where special structures of such processes are exploited (see [21] for a review of simulating variance gamma processes). A nice feature of the inverse transform method is that the dimension of the problem stays minimal. This is important for quasi-Monte Carlo methods since their performance decreases as the dimension increases. For example, to simulate d increments of a normal inverse Gaussian process using its Brownian subordination representation, one needs $3d-$dimensional sequences. This is in contrast to $d-$dimensional sequences required by the inverse transform method. Our numerical results show that the inverse transform method could be many times better than the Brownian subordination approach, even when an inverse Gaussian bridge [35] is used.

Quasi-Monte Carlo methods could be very attractive when combined with appropriate variance reduction techniques. This is illustrated with the pricing of discrete arithmetic Asian options in the CGMY model. In this type of applications, discrete Asian options where the average is in the geometric sense serve as excellent control variates. For a discrete geometric Asian option, we present a Hilbert transform based method which can be used to price the option highly accurately in almost no time. When such control variates are used, the randomized quasi-Monte Carlo method becomes very fast and accurate and outperforms the standard Monte Carlo method by tens of thousands of times.

The contribution of the paper is two-fold. Theoretically, we derive explicit bounds for the estimation bias for simulating from tabulated distribution functions in general and analytic characteristic functions in particular. These explicit bounds lead to convenient ways to determine the grid where probabilities are tabulated and numerical parameters for approximating the probabilities. Numerically, we exhibit the effectiveness of the inverse transform method for simulating Lévy processes with analytic characteristic functions and for options pricing in Lévy models, in particular, when it is combined with quasi-Monte Carlo methods and variance reduction techniques.

The remaining of the paper is organized as follows. In Section 2, we present the Hilbert transform method for inverting a characteristic function to compute the values of a cdf. In particular, we present explicit and computable expressions for the error estimates which allow

us to compute probabilities to any desired accuracy. In Section 3, we describe the inverse transform method for simulating from an approximated and tabulated cdf on a uniform grid, and briefly describe quasi-Monte Carlo methods that will be used in the paper. In Section 4, we first derive a bound for the estimation bias in terms of the interpolation step size, the range of the grid where probabilities are stored, and the approximation error for computing the probabilities on the grid. No specific method is assumed regarding how the probabilities are approximated. Then for a class of characteristic functions that are of interest to us, we present a bound for the estimation bias in terms of the known characteristic function. We also give results for multidimensional problems. In Section 5, we describe three representative Lévy processes that will be tested and discuss options pricing in Lévy models. In particular, we present a Hilbert transform method for pricing discrete geometric Asian options, which will be used as control variates in pricing more commonly traded arithmetic Asian options. In Section 6, we verify our theoretical results, compare the performance of the inverse transform method with commonly used alternative schemes for the representative Lévy processes, and present numerical results exhibiting the effectiveness of the inverse transform method. Section 7 concludes.

2 Hilbert transform method for characteristic functions inversion

[17] presents a Hilbert transform method for the inversion of a characteristic function for computing various quantities, including the cumulative distribution function. The Hilbert transform representation can be discretized using simple rules. For a wide class of analytic characteristic functions that naturally arise in financial applications, the method exhibits exponentially decaying errors. More importantly, the error estimates admit explicit and computable expressions, which allow us to compute the values of a cdf to any desired accuracy level. The Hilbert transform of an integrable function f is defined by the following Cauchy principal value integral [41]:

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

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. Then the Hilbert transform representation can be discretized using simple rules with exponentially decaying errors. More specifically, let $\mathcal{D}_{(d_-,d_+)} = \{z \in \mathbb{C} : \Im(z) \in (d_-, d_+)\}$ for some $-\infty < d_- < 0 < d_+ < +\infty$, where $\Im(z)$ denotes the imaginary part of z. We say that ϕ is in $H(\mathcal{D}_{(d_-,d_+)})$ if it is analytic in $\mathcal{D}_{(d_-,d_+)}$, absolutely integrable on $\mathbb{R}$, and satisfies:

$$\int_{d_-}^{d_+} |\phi(\xi + iy)| dy \to 0, \quad \xi \to \pm\infty, \tag{2.1a}$$

$$||\phi||^{\pm} := \lim_{\epsilon \to 0+} \int_{\mathbb{R}} |\phi(\xi + i(d_{\pm} \mp \epsilon))| d\xi < +\infty. \tag{2.1b}$$

Then the Hilbert transform representation for the cdf $F(x)$ can be approximated by the following:

$$F_{h,M}(x) = \frac{1}{2} + \frac{i}{2} \sum_{m=-M}^{M} e^{-ix(m-1/2)h} \frac{\phi((m-1/2)h)}{(m-1/2)\pi}, \quad h > 0, M \geq 1. \tag{2.2}$$

The discretization error $|F(x) - F_{h,\infty}(x)|$ is of the order $O(\exp(-c/h))$ and hence converges to zero exponentially in $1/h$. Here h is the discretization step size. For a finite truncation level $M \geq 1$, the truncation error $|F_{h,\infty}(x) - F_{h,M}(x)|$ depends on the tail behavior of the characteristic function. In many applications, the characteristic function satisfies the following for some constants $\kappa, c, \nu > 0$:

$$|\phi(\xi)| \leq \kappa \exp(-c|\xi|^{\nu}), \quad \xi \in \mathbb{R}. \tag{2.3}$$

The exponential tail behavior of the characteristic function corresponds to the smoothness of the probability density function. When (2.3) is satisfied, $|F_{h,\infty}(x) - F_{h,M}(x)|$ is of the order $O(e^{-c(Mh)^{\nu}})$ and decays exponentially in terms of Mh. On the other hand, when the characteristic function decays polynomially, so will the truncation error. Suppose the following holds for some $\kappa, \nu > 0$:

$$|\phi(\xi)| \leq \frac{\kappa}{|\xi|^{\nu}}, \quad \xi \in \mathbb{R}. \tag{2.4}$$

Then the truncation error $|F_{h,\infty}(x) - F_{h,M}(x)|$ is of the order $O((Mh)^{-\nu})$. The following theorem summarizes the above and presents an explicit expression for the error estimate when $\phi \in H(\mathcal{D}_{(d_-,d_+)})$ and satisfies (2.3) or (2.4):

Theorem 1. *Let $F(x)$ and $\phi(\xi)$ be the cdf and the characteristic function of a continuous distribution. Suppose that $\phi \in H(\mathcal{D}_{(d_-,d_+)})$. Then*

$$F(x) = \frac{1}{2} - \frac{i}{2} \mathcal{H}(e^{-i\xi x}\phi(\xi))(0). \tag{2.5}$$

For any $a \in (d_-, 0)$,

$$1 - F(x) = e^{ax} \int_{\mathbb{R}} \frac{e^{-ix\xi}\phi(\xi + ia)}{2\pi i(\xi + ia)} d\xi. \tag{2.6}$$

For any $a \in (0, d_+)$,

$$F(x) = e^{ax} \int_{\mathbb{R}} \frac{e^{-ix\xi}\phi(\xi + ia)}{-2\pi i(\xi + ia)} d\xi. \tag{2.7}$$

If ϕ satisfies (2.3) for some $c, \nu, \kappa > 0$, then for any $h > 0$ and integer $M \geq 1$,

$$\begin{aligned} |F(x) - F_{h,M}(x)| \quad \leq \quad & \frac{e^{-2\pi|d_-|/h + xd_-}}{2\pi|d_-|(1 - e^{-2\pi|d_-|/h})} ||\phi||^- + \frac{e^{-2\pi d_+/h + xd_+}}{2\pi d_+(1 - e^{-2\pi d_+/h})} ||\phi||^+ \\ & + \frac{\kappa}{2\pi} \Big(\frac{1}{M} + \frac{4}{\nu c (Mh)^{\nu}} \Big) e^{-c(Mh)^{\nu}}. \end{aligned} \tag{2.8}$$

If ϕ satisfies (2.4) for some $\nu, \kappa > 0$, then for any $h > 0$ and integer $M \geq 1$,

$$\begin{aligned} |F(x) - F_{h,M}(x)| \quad \leq \quad & \frac{e^{-2\pi|d_-|/h + xd_-}}{2\pi|d_-|(1 - e^{-2\pi|d_-|/h})} ||\phi||^- + \frac{e^{-2\pi d_+/h + xd_+}}{2\pi d_+(1 - e^{-2\pi d_+/h})} ||\phi||^+ \\ & + \frac{\kappa}{2\pi} \Big(\frac{1}{M} + \frac{2}{\nu} \Big) \frac{1}{(Mh)^{\nu}}. \end{aligned} \tag{2.9}$$

Proof. See [17] for the proof. □

Remark 1.

1. The alternative representations (2.6) and (2.7) exhibit the exponential tail of $1 - F(x)$ for positive x and that of $F(x)$ for negative x. They are useful when we analyze the bias of the inverse transform method.

2. Schemes for computing the alternative representations (2.6) and (2.7) can be found in [17]. They may further speed up the convergence in some cases. For simplicity, we didn't present these alternative schemes since the approximation (2.2) that corresponds to (2.5) is sufficient in most of our applications.

In derivatives pricing applications, the price process of the underlying asset is modeled by the exponential of a stochastic process. For the asset itself to be priced, certain exponential moments of the corresponding stochastic process exist. Consequently, the characteristic function is analytic [30]. In many models, the transition probability density is smooth and the characteristic function admits exponentially decaying tails. $F_{h,M}(x)$ thus provides a highly accurate approximation to the cdf $F(x)$. The explicit bounds (2.8) and (2.9) further allow us to compute the cdf to any desired accuracy level.

Due to the e^{-ixmh} term, the fast Fourier transform can be used to compute multiple values of the cdf simultaneously. More specifically, given any evenly spaced sequence $\{x_0, x_0+\eta, \cdots, x_0+(N-1)\eta\}$, the computation of $\{F(x_0 + n\eta), n = 0, 1, \cdots, N-1\}$ can be reduced to a Toeplitz matrix vector multiplication which can be implemented using the fast Fourier transform with $O(N \log N)$ operations only. See the appendices of [18, 19] 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 $h\eta = 2\pi/N$, which can be inconvenient and restrictive.

3 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 finite time grid $0 = t_0 < t_1 < \cdots < t_d = T$ for some positive integer d:

$$\mathbb{E}[f(X_{t_1}, \cdots, X_{t_d})]. \tag{3.10}$$

Sometimes, we are only concerned about the values of a Lévy process on such a discrete time grid (e.g., pricing a weekly monitored Asian option in a Lévy model). Other times, we use the above to approximate an expectation of the form $\mathbb{E}[f(X_s, 0 \leq s \leq T)]$ that is difficult to compute directly. Suppose Monte Carlo methods are used to evaluate the above expectation (3.10). We simulate N paths of the Lévy process: $\{(X^n_{t_1}, \cdots, X^n_{t_d}) : 1 \leq n \leq N\}$ and then estimate the above expectation by the following:

$$\frac{1}{N}\sum_{n=1}^{N} f(X^n_{t_1}, \cdots, X^n_{t_d}).$$

Due to the independent and stationary increment properties of Lévy processes, $X_{t_j} - X_{t_{j-1}}, 1 \leq j \leq d$, are independent. Moreover, $X_{t_j} - X_{t_{j-1}}$ has the same distribution as X_{Δ_j}, where $\Delta_j = t_j - t_{j-1}, 1 \leq j \leq d$. Therefore, to simulate a path of the Lévy process, it suffices to simulate independent Lévy increments, denoted by $Y_j, 1 \leq j \leq d$, from the distributions of $X_{\Delta_j}, 1 \leq j \leq d$. Note that a Lévy process starts from the origin at time 0: $X_0 = 0$. We therefore let $X_{t_j} = \sum_{k=1}^{j} Y_k, 1 \leq j \leq d$. Due to the celebrated Lévy-Khintchine formula for infinitely divisible distributions, the characteristic function of a Lévy process often admits a closed-form expression. It thus suffices to simulate from the characteristic functions of the Lévy increments. In the following, we first describe the inverse transform method for simulating from an approximated and tabulated cdf. We then briefly describe quasi-Monte Carlo methods. Bias analysis is left to Section 4.

3.1 The inverse transform method from tabulated probabilities

To simulate from a strictly increasing continuous distribution function $F(x)$, it suffices to simulate a uniformly distributed random variable U on $(0, 1)$ and compute $F^{-1}(U)$. In general, F^{-1} does not admit a closed form expression. In many applications, even $F(x)$ is not known explicitly and needs to be approximated. In such cases, we take a large enough interval $\mathcal{X} = [x_0, x_K]$ (the determination of such an interval for a given bias tolerance level will be studied in Section 4). Let $\eta = (x_K - x_0)/K$ for a positive integer K, and $x_k = x_0 + k\eta, 0 \leq k \leq K$. Denote $F_k = F(x_k), 0 \leq k \leq K$. We approximate F_k by $\hat{F}_k$, and store the pairs $(x_k, \hat{F}_k)$ in the following matrix:

$$\begin{pmatrix} x_0 & x_1 & \cdots & x_K \\ \hat{F}_0 & \hat{F}_1 & \cdots & \hat{F}_K \end{pmatrix}. \tag{3.11}$$

In particular, we insure that $\{\hat{F}_0, \cdots, \hat{F}_K\}$ is a strictly increasing nonnegative sequence. For the class of distributions we focus on in this paper, we are able to compute the cdf to any desired accuracy. If the monotone requirement is violated, we increase the accuracy of the approximation scheme.

For each generated U, which is between 0 and 1, we use binary search to find $0 \leq k \leq K-1$ so that $\hat{F}_k \leq U < \hat{F}_{k+1}$. We then return the following as an approximation to $F^{-1}(U)$:

$$x_k + \frac{x_{k+1} - x_k}{\hat{F}_{k+1} - \hat{F}_k}(U - \hat{F}_k).$$

If $0 < U < \hat{F}_0$, we return x_0. Similarly, if $\hat{F}_K \leq U < 1$, we return x_K. As can be seen easily, we are simulating from the following distribution instead:

$$\hat{F}(x) = \begin{cases} 0, & x < x_0 \\ \hat{F}_{k-1} + \frac{\hat{F}_k - \hat{F}_{k-1}}{\eta}(x - x_{k-1}), & x_{k-1} \leq x < x_k, 1 \leq k \leq K \\ 1, & x \geq x_K \end{cases}. \tag{3.12}$$

Note that when $\hat{F}_0 > 0$, the above distribution has a probability mass at x_0. Similarly, when $\hat{F}_K < 1$, there is a probability mass at x_K. We also remark that it takes roughly $\log_2(K)$ iterations to locate the interval that contains U.

Remark 2.

- It is not necessary that $\hat{F}_0 \approx 0$ and $\hat{F}_K \approx 1$. For example, when one is evaluating $\mathbb{E}[f(X)]$ and $f(x) = 0$ for $x \geq a$, one may simply take $x_K = a$, although $\hat{F}(a)$ could be far below 1. Moreover, in such a case, one only needs to simulate U that is uniform between 0 and $\hat{F}(a)$ to avoid getting many zeros and scale the final estimation by $\hat{F}(a)$. In our numerical examples in Section 6, we didn't implement this feature. But it obviously could be used whenever applicable to reduce the variance.

Obviously, the above procedure will introduce bias. There are three sources of bias: (1) the truncation of the domain of the original distribution from $\mathbb{R}$ to $[x_0, x_K]$; (2) the approximation of the original cdf by a piecewise linear function; (3) the approximation of the probabilities on the grid. In Section 4, we present bounds for the total bias of the above scheme.

3.2 Randomized quasi-Monte Carlo methods

The inverse transform method can be easily coupled 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 $U_1, \cdots, U_d$ on $(0, 1)$:

$$\mathbb{E}[f(U_1, \cdots, U_d)]. \tag{3.13}$$

When estimating the expectation (3.13) 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:

$$\frac{1}{N_0} \sum_{n=1}^{N_0} f(x_n^1, \cdots, x_n^d), \tag{3.14}$$

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 standard 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, while error estimates are rather straightforward to obtain for standard Monte Carlo methods, it is much 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 [22]. One way to randomize a quasi-Monte Carlo method is to use the so called digital shift [28, 29]. 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 use them to generate the following shifted sequence $\{x_1^U, \cdots, x_{N_0}^U\}$:

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

where $\oplus$ refers to the exclusive-or operator. From this shifted sequence, we obtain an approximation of the form (3.14). 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 (3.13). The standard error computed from these L variables provides an estimate of the error of the quasi-Monte Carlo approximation. In total, $N = LN_0$ realizations have been generated to obtain the above estimate. This method will be used in our numerical experiments. 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 [22].

4 Estimation bias

In this section, we analyze the bias introduced by simulating from $\hat{F}$ instead of the original distribution F. In the first subsection, we present a bound for the bias in terms of the truncated domain $\mathcal{X} = [x_0, x_K]$, fineness of the grid on $\mathcal{X}$, and the approximation error for the probabilities. In particular, the result is independent of how the probabilities are approximated. Any method of preference can be used. In the second subsection, we consider distributions with characteristic functions in our analytic class, and present the bound for the bias that can be computed from the given characteristic function directly. In the third subsection, we follow the approach of [24] and study the multidimensional case.

4.1 The general case

Suppose we are evaluating $\mathbb{E}[f(X)]$ for a continuous random variable X with distribution $F(x)$ and density $p(x)$. Instead of simulating from $F(x)$ precisely, we simulate from $\hat{F}(x)$ defined in (3.12) with the following density:

$$\hat{p}(x) = \hat{F}_0 \cdot \delta(x - x_0) + (1 - \hat{F}_K) \cdot \delta(x - x_K) + \frac{1}{\eta} \sum_{k=1}^{K} (\hat{F}_k - \hat{F}_{k-1}) \cdot \mathbf{1}_{(x_{k-1}, x_k)}(x),$$

where $\delta(\cdot)$ is the Dirac Delta function, and $\mathbf{1}_A(x)$ is the indicator function which takes value one if $x \in A$ and zero otherwise. Denote

$$\begin{aligned} |\mathcal{X}| &= x_K - x_0, \\ ||f||_{\mathcal{X}} &= \sup_{x \in \mathcal{X}} |f(x)|, \\ ||f'||_{\mathcal{X}} &= \operatorname{ess\,sup}_{x \in \mathcal{X}} |f'(x)|, \\ ||p'||_{\mathcal{X}} &= \operatorname{ess\,sup}_{x \in \mathcal{X}} |p'(x)|, \\ E_{\mathcal{X}} &= \max_{0 \le i \le K} |F(x_i) - \hat{F}_i|. \end{aligned}$$

Then we have the following upper bound for the estimation bias.

Theorem 2. *Consider a continuous random variable X with cumulative distribution function F and density p. Let $\{x_0, x_1, \cdots, x_K\}$ be a uniform grid with step size $\eta = (x_K - x_0)/K$ for some positive integer K. Let $\hat{X}$ be a random variable with distribution function $\hat{F}$ defined in (3.12). Suppose f is differentiable in $\bigcup_{i=1}^{K}(x_i, x_{i-1})$ except at n^f points, and $||f'||_{\mathcal{X}} < \infty$, $||f||_{\mathcal{X}} < \infty$. p is bounded on $\mathcal{X}$, differentiable in $\bigcup_{i=1}^{K}(x_i, x_{i-1})$, and $||p'||_{\mathcal{X}} < \infty$. Then*

$$\begin{aligned} |\mathbb{E}[f(X)] - \mathbb{E}[f(\hat{X})]| \le\ & |f(x_0)|F(x_0) + |f(x_K)|(1 - F(x_K)) + \Big(\int_{-\infty}^{x_0} + \int_{x_K}^{\infty} \Big) |f(x)|p(x)dx \\ & + \frac{1}{K^2} ||f'||_{\mathcal{X}} ||p'||_{\mathcal{X}} |\mathcal{X}|^3 \\ & + \Big(|f(x_0)| + |f(x_K)| + 2(K + n^f)||f||_{\mathcal{X}} + 2||f'||_{\mathcal{X}}|\mathcal{X}| \Big) E_{\mathcal{X}}. \end{aligned} \tag{4.15}$$

Proof. The bias is given by

$$\begin{aligned}
\mathbb{E}[f(X)] - \mathbb{E}[f(\hat{X})] &= \int_{\mathbb{R}} f(x)p(x)dx - \int_{\mathbb{R}} f(x)\hat{p}(x)dx \\
&= \int_{\mathbb{R}} f(x)p(x)dx - \int_{x_0}^{x_K} f(x)\hat{p}(x)dx - f(x_0)\hat{F}_0 - f(x_K)(1-\hat{F}_K) \\
&= \sum_{k=1}^{K}\int_{x_{k-1}}^{x_k} f(x)(p(x)-\hat{p}(x))dx + \Big(\int_{-\infty}^{x_0} + \int_{x_K}^{\infty}\Big) f(x)p(x)dx \\
&\quad -f(x_0)\hat{F}_0 - f(x_K)(1-\hat{F}_K). \qquad (4.16)
\end{aligned}$$

If $f(x)$ is differentiable in (x_{k-1}, x_k), then for any $x \in (x_{k-1}, x_k)$, by the mean value theorem, there exists $\xi_k(x) \in (x_{k-1}, x)$ such that

$$\begin{aligned}
\Big|\int_{x_{k-1}}^{x_k} f(x)(p(x)-\hat{p}(x))dx\Big| &= \Big|\int_{x_{k-1}}^{x_k} (f(x_{k-1}+) + f'(\xi_k(x))(x-x_{k-1}))(p(x)-\hat{p}(x))dx\Big| \\
&\le ||f||_{\mathcal{X}} \cdot |F_k - \hat{F}_k - (F_{k-1} - \hat{F}_{k-1})| + ||f'||_{\mathcal{X}} \cdot \eta \cdot \int_{x_{k-1}}^{x_k} |p(x)-\hat{p}(x)|dx \\
&\le 2 \cdot ||f||_{\mathcal{X}} \cdot E_{\mathcal{X}} + ||f'||_{\mathcal{X}} \cdot \eta \cdot \int_{x_{k-1}}^{x_k} |p(x)-\hat{p}(x)|dx.
\end{aligned}$$

Here $f(x_{k-1}+)$ is the right limit of f at x_{k-1}, which is finite by the assumptions. In general, if $f(x)$ is not differentiable at n_k^f points in (x_{k-1}, x_k), where $\sum_{k=1}^{K} n_k^f = n^f$, it can be shown in the same way as above that

$$\Big|\int_{x_{k-1}}^{x_k} f(x)(p(x)-\hat{p}(x))dx\Big| \le 2(n_k^f+1) \cdot ||f||_{\mathcal{X}} \cdot E_{\mathcal{X}} + ||f'||_{\mathcal{X}} \cdot \eta \cdot \int_{x_{k-1}}^{x_k} |p(x)-\hat{p}(x)|dx.$$

Note that

$$\begin{aligned}
\int_{x_{k-1}}^{x_k} |p(x)-\hat{p}(x)|dx &= \int_{x_{k-1}}^{x_k} |p(x) - \frac{F_k - F_{k-1}}{\eta} + \frac{F_k - F_{k-1}}{\eta} - \frac{\hat{F}_k - \hat{F}_{k-1}}{\eta}|dx \\
&\le |F_k - \hat{F}_k - (F_{k-1} - \hat{F}_{k-1})| + \int_{x_{k-1}}^{x_k} |p(x) - \frac{F_k - F_{k-1}}{\eta}|dx \\
&\le 2E_{\mathcal{X}} + \int_{x_{k-1}}^{x_k} |p(x) - \frac{F_k - F_{k-1}}{\eta}|dx.
\end{aligned}$$

However, we have the following:

$$\int_{x_{k-1}}^{x_k} |p(x) - \frac{F_k - F_{k-1}}{\eta}|dx = \frac{1}{\eta}\int_{x_{k-1}}^{x_k} \Big|\int_{x_{k-1}}^{x_k} (p(x)-p(y))dy\Big|dx.$$

Since $p(x)$ is differentiable in (x_{k-1}, x_k), using the mean value theorem again, for any $x \neq y$ in (x_{k-1}, x_k), there exists $\xi_k(x,y) \in (x,y)$ such that

$$p(y) = p(x) + p'(\xi_k(x,y))(y-x).$$

Therefore,

$$\int_{x_{k-1}}^{x_k} |p(x)-\hat{p}(x)|dx \le 2E_{\mathcal{X}} + ||p'||_{\mathcal{X}} \cdot \eta^2.$$

Consequently,

$$\Big|\sum_{k=1}^{K}\int_{x_{k-1}}^{x_k} f(x)(p(x)-\hat{p}(x))dx\Big| \le 2E_{\mathcal{X}}\Big((K+n^f)||f||_{\mathcal{X}}+||f'||_{\mathcal{X}}(x_K-x_0)\Big)+||f'||_{\mathcal{X}}||p'||_{\mathcal{X}}(x_K-x_0)\eta^2.$$

As for the last two terms in (4.16), we have

$$|f(x_0)\hat{F}_0| \le |f(x_0)|(E_{\mathcal{X}}+F_0), \quad |f(x_K)(1-\hat{F}_K)| \le |f(x_K)|(E_{\mathcal{X}}+1-F_K).$$

Combining the above, we obtain (4.15). □

Remark 3.

1. In option pricing applications, there often exists some point where f is not continuous or differentiable. According to Theorem 2, one may (but is not required to) make such a point one of the nodes in $\{x_0, x_1, \cdots, x_K\}$ to reduce the bias. Recall that the n^f points do not include grid points.

2. When $f(x)$ vanishes on the left of a point a, one may simply take $x_0 = a$ without introducing bias. Similarly, when $f(x)$ vanishes on the right of a point b, one may take $x_K = b$. Such cases naturally arise in pricing European vanilla options, as exhibited in Section 6.

3. For a given target level $\varepsilon_b > 0$ for the total bias, we determine x_0, x_K, K and the scheme for computing the probabilities in (3.11) correspondingly. In examples presented in this paper, the probabilities can be approximated highly accurately. Moreover, the matrix (3.11) only needs to be computed once. We thus select an appropriate $\mathcal{X} = [x_0, x_K]$ so that the first line in (4.15) is below $\varepsilon_b/2$. Then we select a large enough K so that the second line in (4.15) is below $\varepsilon_b/2$. Finally, we adjust the scheme for approximating the probabilities so that the third line in (4.15) is below $0.01\varepsilon_b$, which is negligible.

4.2 Distributions with analytic characteristic functions

Recall that Theorem 2 does not depend on how the probabilities in (3.11) are approximated. One may choose a method that is preferred. In the following, we assume that the characteristic function of the distribution is known and falls in the analytic class $H(\mathcal{D}_{(d_-,d_+)})$, and the probabilities are computed using (2.2): $\hat{F}_k = F_{h,M}(x_k)$. Denote

$$E_{h,M,\mathcal{X}} = \max_{0\le i\le K} |F(x_i) - F_{h,M}(x_i)|.$$

We have the following bound for the estimation bias.

Theorem 3. *Consider a continuous random variable X with characteristic function ϕ. Let $\{x_0, x_1, \cdots, x_K\}$ be a uniform grid with step size $\eta = (x_K - x_0)/K$ for some positive integer K. Suppose $\phi \in H(\mathcal{D}_{(d_-,d_+)})$ for some $-\infty < d_- < 0 < d_+ < \infty$, and satisfies (2.3) for some $\nu > 0$ or (2.4) for some $\nu > 2$. Let $\hat{X}$ be a random variable with distribution function $\hat{F}$*

defined in (3.12), where $\hat{F}_k$ is given by $F_{h,M}(x_k)$ defined in (2.2). Suppose f is differentiable in $\bigcup_{i=1}^{K}(x_i, x_{i-1})$ except at n^f points, and $||f||_{\mathcal{X}} < \infty$, $||f'||_{\mathcal{X}} < \infty$. Then

$$
\begin{aligned}
|\mathbb{E}[f(X)] - \mathbb{E}[f(\hat{X})]| \leq\ & \frac{||\phi||^-}{2\pi}\Big(\int_{x_K}^{\infty}|f(x)|e^{xd_-}dx + \frac{1}{|d_-|}|f(x_K)|e^{x_K d_-}\Big) \\
& + \frac{||\phi||^+}{2\pi}\Big(\int_{-\infty}^{x_0}|f(x)|e^{xd_+}dx + \frac{1}{d_+}|f(x_0)|e^{x_0 d_+}\Big) \\
& + \frac{1}{2\pi K^2}||f'||_{\mathcal{X}}|\mathcal{X}|^3\int_{\mathbb{R}}|\xi\phi(\xi)|d\xi \\
& + \Big(|f(x_0)| + |f(x_K)| + 2(K + n^f)||f||_{\mathcal{X}} + 2||f'||_{\mathcal{X}}|\mathcal{X}|\Big)E_{h,M,\mathcal{X}}.
\end{aligned} \tag{4.17}
$$

Proof. Let F be the cdf of X. According to Theorem 2, the bias is bounded by (4.15). If $\phi \in H(\mathcal{D}_{(d_-,d_+)})$, from (2.7), for any $a \in (0, d_+)$, we have

$$
f(x_0)F(x_0) = f(x_0)\int_{\mathbb{R}}\frac{e^{-ix_0(\xi+ia)}\phi(\xi+ia)}{-2\pi i(\xi+ia)}d\xi = f(x_0)\int_{-\infty+ia}^{+\infty+ia}\frac{e^{-ix_0 z}\phi(z)}{-2\pi i z}dz.
$$

Since the integrand above is analytic in $\{z \in \mathbb{C} : \Im(z) \in (0, d_+)\}$, using the condition (2.1) and Cauchy's integral theorem, for any $\epsilon > 0$ such that $d_+ - \epsilon > a$, we have

$$
f(x_0)F(x_0) = f(x_0)\int_{-\infty+i(d_+-\epsilon)}^{+\infty+i(d_+-\epsilon)}\frac{e^{-ix_0 z}\phi(z)}{-2\pi i z}dz = f(x_0)e^{(d_+-\epsilon)x_0}\int_{\mathbb{R}}\frac{e^{-ix_0\xi}\phi(\xi+i(d_+-\epsilon))}{-2\pi i(\xi+i(d_+-\epsilon))}d\xi.
$$

Let $\epsilon \downarrow 0$, we obtain

$$
|f(x_0)F(x_0)| \leq \frac{||\phi||^+}{2\pi d_+}|f(x_0)|e^{x_0 d_+}.
$$

Note that the probability density $p(x)$ admits the following inverse Fourier transform representation:

$$
p(x) = \frac{1}{2\pi}\int_{\mathbb{R}}e^{-izx}\phi(z)dz.
$$

By Cauchy's integral theorem and the condition (2.1), for any $\epsilon > 0$ such that $d_+ - \epsilon > 0$, we have

$$
p(x) = \frac{1}{2\pi}\int_{-\infty+i(d_+-\epsilon)}^{+\infty+i(d_+-\epsilon)}e^{-izx}\phi(z)dz = e^{(d_+-\epsilon)x}\int_{\mathbb{R}}\frac{1}{2\pi}e^{-ix\xi}\phi(\xi+i(d_+-\epsilon))d\xi.
$$

Consequently,

$$
\int_{-\infty}^{x_0}|f(x)|p(x)dx \leq \frac{||\phi||^+}{2\pi}\int_{-\infty}^{x_0}|f(x)|e^{xd_+}dx.
$$

Similarly, using the representation (2.6), we have the following:

$$
|f(x_K)|(1 - F(x_K)) \leq \frac{||\phi||^-}{2\pi|d_-|}|f(x_K)|e^{x_K d_-},
$$

$$
\int_{x_K}^{\infty}|f(x)|p(x)dx \leq \frac{||\phi||^-}{2\pi}\int_{x_K}^{\infty}|f(x)|e^{xd_-}dx.
$$

Since $\xi\phi(\xi)$ is absolutely integrable on $\mathbb{R}$ by the assumptions on ϕ,

$$p'(x) = \frac{1}{2\pi}\frac{d}{dx}\int_{\mathbb{R}} e^{-i\xi x}\phi(\xi)d\xi = \frac{1}{2\pi i}\int_{\mathbb{R}} e^{-i\xi x}\xi\phi(\xi)d\xi,$$

where the interchange of the integration and differentiation is valid due to the dominated convergence theorem. Therefore,

$$||p'||_{\mathcal{X}} \leq \frac{1}{2\pi}\int_{\mathbb{R}} |\xi\phi(\xi)|d\xi.$$

Combining the above, we obtain the bound for the bias in (4.17). □

Remark 4.

1. From the first two terms in (4.17) and the known function $f(x)$, it is straightforward to compute x_0 and x_K so that the corresponding portion of the bias is below a desired level.
2. From Theorem 1, it is easy to determine h and M so that the last term in (4.17) is below a desired level.

4.3 The multidimensional case

The multidimensional case is more complex. It is difficult to obtain a realistic bound for the simulation bias that is easily computable. We provide a procedure for determining the grid and numerical parameters for characteristic function inversion. The following theorem guarantees the convergence of the procedure. Numerical results in Section 6 show that the method works well practically. Following the approach of [24], we can show the theorem below:

Theorem 4. *Let $X_i, 1 \leq i \leq d$, be d independent random variables with distributions $F^{(i)}$ and densities p_i. For $1 \leq i \leq d$, let $\{x_0^{(i)}, \cdots, x_{K_i}^{(i)}\}$ be a uniform grid with step size $\eta_i = (x_{K_i}^{(i)} - x_0^{(i)})/K_i$ for some integer $K_i > 0$. Let $\hat{X}_i, 1 \leq i \leq d$, be d independent random variables with distributions $\hat{F}^{(i)}$ defined in (3.12) on $\mathcal{X}_i = [x_0^{(i)}, x_{K_i}^{(i)}]$. For any $1 \leq i \leq d$, $f(x_1, \cdots, x_d)$ as a function of x_i is differentiable in $\mathcal{X}_i$ except at up to n^f points, and both $|f(x_1, \cdots, x_d)|$ and $|df(x_1, \cdots, x_d)/dx_i|$ are bounded by $g_1(x_1)g_2(x_2)\cdots g_d(x_d)$ for some $g_i \geq 0$. For $1 \leq i \leq d$, g_i is differentiable in $\mathcal{X}_i$ except at up to n^f points and*

$$||g_i||^*_{\mathcal{X}_i} = \max\Big(\sup_{x\in\mathcal{X}_i}|g_i(x)|, \operatorname{ess\,sup}_{x\in\mathcal{X}_i}|g_i'(x)|\Big) < \infty.$$

Denote $||p_i g_i||_1 = \int_{\mathbb{R}} p_i(x)g_i(x)dx, 1 \leq i \leq d$. Define

$$\begin{aligned} B_i \;=\; & g_i(x_0^{(i)})F^{(i)}(x_0^{(i)}) + g_i(x_{K_i}^{(i)})(1 - F^{(i)}(x_{K_i}^{(i)})) + \Big(\int_{-\infty}^{x_0^{(i)}} + \int_{x_{K_i}^{(i)}}^{\infty}\Big) g_i(x)p_i(x)dx \\ & + \frac{1}{K_i^2}||g_i||^*_{\mathcal{X}_i}||p_i'||_{\mathcal{X}_i}|\mathcal{X}_i|^3 \\ & + \Big(g_i(x_0^{(i)}) + g_i(x_{K_i}^{(i)}) + 2(K_i + n^f)||g_i||^*_{\mathcal{X}_i} + 2||g_i||^*_{\mathcal{X}_i}|\mathcal{X}_i|\Big) E_{\mathcal{X}_i}, \end{aligned} \tag{4.18}$$

where $||p_i'||_{\mathcal{X}_i}, |\mathcal{X}_i|, E_{\mathcal{X}_i}$ *are defined similarly as in Theorem 2. Let* $B = \max(B_1, \cdots, B_d)$*. Then there exist constants* $a_i > 0, 0 \le i \le d-1$*, independent of* B *such that*

$$|\mathbb{E}[f(X_1, \cdots X_d)] - \mathbb{E}[f(\hat{X}_1, \cdots, \hat{X}_d)]| \le B(a_0 + a_1 B + \cdots + a_{d-1} B^{d-1}). \tag{4.19}$$

Proof. Denote the density of $\hat{F}^{(i)}$ by $\hat{p}_i$. From Theorem 2, we have the following for any $1 \le i \le d$:

$$\int_{\mathbb{R}} g_i(x)\hat{p}_i(x)dx \le ||p_i g_i||_1 + B_i,$$

$$\Big| \int_{\mathbb{R}} f(x_1, \cdots, x_d)(p_i(x_i) - \hat{p}_i(x_i))dx_i \Big| \le B_i \prod_{j=1, j\neq i}^{d} g_j(x_j).$$

Following [24], $|\mathbb{E}[f(X_1, \cdots X_d)] - \mathbb{E}[f(\hat{X}_1, \cdots, \hat{X}_d)]|$ is bounded by the following:

$$\begin{aligned}
&\Big|\mathbb{E}[f(X_1, X_2, \cdots, X_{d-1}, X_d)] - \mathbb{E}[f(X_1, X_2, \cdots, X_{d-1}, \hat{X}_d)]\Big| \\
+ &\Big|\mathbb{E}[f(X_1, X_2, \cdots, X_{d-1}, \hat{X}_d)] - \mathbb{E}[f(X_1, X_2, \cdots, \hat{X}_{d-1}, \hat{X}_d)]\Big| + \\
&\vdots \\
+ &\Big|\mathbb{E}[f(X_1, \hat{X}_2, \cdots, \hat{X}_{d-1}, \hat{X}_d)] - \mathbb{E}[f(\hat{X}_1, \hat{X}_2, \cdots, \hat{X}_{d-1}, \hat{X}_d)]\Big| \\
\le \quad & B_d \prod_{j=1}^{d-1} ||p_j g_j||_1 + B_{d-1} \prod_{j=1}^{d-2} ||p_j g_j||_1 ||\hat{p}_d g_d||_1 + \cdots + B_1 \prod_{j=2}^{d} ||\hat{p}_j g_j||_1 \\
\le \quad & B\Big(\prod_{j=1}^{d-1} ||p_j g_j||_1 + \prod_{j=1}^{d-2} ||p_j g_j||_1 (||p_d g_d||_1 + B) + \cdots + \prod_{j=2}^{d} (||p_j g_j||_1 + B)\Big),
\end{aligned}$$

where $||\hat{p}_i g_i||_1 = \int_{\mathbb{R}} \hat{p}_i(x) g_i(x) dx, 1 \le i \le d$. The conclusion then follows immediately. □

When d is large, the constant a_0 could be difficult to compute and it is inconvenient to use (4.19) directly to control the estimation bias. However, the theorem suggests that the bias is approximately proportional to B. We therefore determine the grids $\{x_0^{(i)}, \cdots, x_{K_i}^{(i)}\}$ and numerical parameters for computing $\hat{F}^{(i)}, 1 \le i \le d$, by bounding B. Since we may not know a_0, we select a sequence of decreasing tolerance levels for B in practice. For each tolerance level, we can precisely determine the grids and the numerical parameters for computing the probabilities. Theorem 4 guarantees that, as B decreases, $\mathbb{E}[f(\hat{X}_1, \cdots, \hat{X}_d)]$ converges to $\mathbb{E}[f(X_1, \cdots X_d)]$.

Obviously, when the characteristic function $\phi^{(i)}$ of the distribution $F^{(i)}$ satisfies the conditions of Theorem 3 for some $-\infty < d_-^{(i)} < 0 < d_+^{(i)} < +\infty$, we may replace B_i in Theorem 4 with

the following:

$$
\begin{aligned}
B_i \;=\; & \frac{||\phi^{(i)}||^-}{2\pi}\Big(\int_{x_{K_i}^{(i)}}^{\infty} g_i(x)e^{xd_-^{(i)}}dx + \frac{1}{|d_-^{(i)}|}g_i(x_{K_i}^{(i)})e^{x_{K_i}^{(i)}d_-^{(i)}}\Big) \\
& + \frac{||\phi^{(i)}||^+}{2\pi}\Big(\int_{-\infty}^{x_0^{(i)}} g_i(x)e^{xd_+^{(i)}}dx + \frac{1}{d_+^{(i)}}g_i(x_0^{(i)})e^{x_0^{(i)}d_+^{(i)}}\Big) \\
& + \frac{1}{2\pi K_i^2}||g_i||_{\mathcal{X}_i}^*|\mathcal{X}_i|^3\int_{\mathbb{R}}|\xi\phi^{(i)}(\xi)|d\xi \\
& + \Big(g_i(x_0^{(i)}) + g_i(x_{K_i}^{(i)}) + 2(K_i+n^f)||g_i||_{\mathcal{X}_i}^* + 2||g_i||_{\mathcal{X}_i}^*|\mathcal{X}_i|\Big)E_{h_i,M_i,\mathcal{X}_i}.
\end{aligned}
\tag{4.20}
$$

In general, the distributions $F^{(i)}$, $1 \le i \le d$, could be different. So are $g_i, 1 \le i \le d$. However, when pricing discretely and evenly monitored option contracts in Lévy models, not only (4.20) is applicable, but also $F^{(1)} = \cdots = F^{(d)}$. Moreover, it is often that $g_1 = \cdots = g_d$. In such cases, the computational effort is greatly reduced since all the grids and probability approximation schemes are the same, and (4.20) only needs to be used once.

5 Options pricing in Lévy models

We consider options pricing in Lévy models. We assume that the price of an asset follows a geometric Lévy process under a given equivalent martingale measure:

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

where S_0 is the initial asset price and X_t is a Lévy process starting at the origin at time zero. Lévy models are attractive since they naturally incorporate jumps in asset prices that are difficult to capture in the celebrated Black-Scholes-Merton model [8, 32]. Depending on the intensity of arrivals of jumps, a Lévy process could be a finite activity jump diffusion process [26, 33] or an infinite activity pure jump process [5, 11, 12, 16, 31]. In this paper, we test three representative cases:

1. The normal inverse Gaussian (NIG) process [5] which can be simulated as a subordinated Brownian motion [38] and through bridge sampling [35];
2. Kou's jump diffusion process [26] which can be simulated straightforwardly by generating a Brownian motion, a Poisson process, and a sequence of double exponential jump sizes;
3. The CGMY process [11] which does not admit simple structures and simulating from its characteristic function seems to be a natural choice.

5.1 The normal inverse Gaussian (NIG) process

A NIG process can be characterized by

$$X_t = \mu t + \beta z_t + B_{z_t},$$

where B_t is a standard Brownian motion and z_t is an independent inverse Gaussian process. z_t has the same distribution as the first hitting time to δt of a Brownian motion with drift $\gamma > 0$.

Let $\alpha = \sqrt{\beta^2 + \gamma^2}$. In a geometric Lévy model, by the requirement that the discounted gain process is a martingale, $\mu = r - q + \delta(\sqrt{\alpha^2 - (\beta+1)^2} - \sqrt{\alpha^2 - \beta^2})$. Here r is the risk free interest rate, and q is the rate at which the underlying asset is paying out (e.g., dividend yield for a stock or foreign risk free interest rate for a foreign currency). The characteristic function of X_t is given by

$$\phi_t(\xi) = \mathbb{E}[e^{i\xi X_t}] = \exp\left(i\mu t\xi - \delta t(\sqrt{\alpha^2 - (\beta + i\xi)^2} - \sqrt{\alpha^2 - \beta^2})\right).$$

It is in $H(\mathcal{D}_{(d_-,d_+)})$ with $d_- = \beta - \alpha$ and $d_+ = \beta + \alpha$. In particular, it satisfies the exponential tail condition (2.3) with $\kappa = \exp(\delta t\sqrt{\alpha^2 - \beta^2}), c = \delta t$ and $\nu = 1$.

The NIG process can be simulated by generating z_t and a standard normal random variable representing the Brownian motion component. For one dimensional applications, we compare the inverse transform method with this algorithm, which is summarized in the following (see [38] as well as [2] Chapter XII).

Algorithm 1 (Simulating a NIG process through a Brownian subordination)**.**

For $t > 0$, simulate $X_t = \mu t + \beta z_t + B_{z_t}$

1. Generate a standard normal random variable G_1 using the Beasley-Springer-Moro algorithm described on p.67 of [22]. With $\gamma = \sqrt{\alpha^2 - \beta^2}$, $Z = G_1^2/\gamma$, compute
$$\zeta = \frac{1}{\gamma}(\delta t + \frac{1}{2}Z - \sqrt{\delta t Z + Z^2/4}).$$
2. Generate a uniform random variable U on $(0,1)$. If $U < \delta t/(\delta t + \gamma\zeta)$, $z_t = \zeta$. Otherwise, $z_t = \delta^2 t^2/(\gamma^2\zeta)$.
3. Generate a standard normal random variable G_2. Let $X_t = \mu t + \beta z_t + \sqrt{z_t}G_2$.

When one is simulating a discrete sample path of a NIG process, inverse Gaussian bridge can be used together with quasi Monte Carlo methods to potentially reduce the effective dimension [35]. For multidimensional applications, we compare the inverse transform method with this algorithm, which is described below.

Algorithm 2 (Simulating a NIG process using inverse Gaussian bridge)**.**

For $0 = t_0 < t_1 < \cdots < t_d$, simulate $X_{t_j} = \mu t_j + \beta z_{t_j} + B_{z_{t_j}}, 1 \le j \le d$

1. Generate z_{t_d} by following steps 1-2 in Algorithm 1. Generate a standard normal random variable G_d and obtain $W_{z_{t_d}} = \beta z_{t_d} + \sqrt{z_{t_d}}G_d$.
2. Construct the bridge as below: for $0 \le t_i < t_j < t_k \le t_d$,
   (a) Inverse Gaussian bridge: conditional on z_{t_i} and z_{t_k}, generate a standard normal random variable G_{1j}, and compute
   $$\lambda = \frac{\delta^2(t_k - t_j)^2}{z_{t_k} - z_{t_i}},\ \theta = \frac{t_k - t_j}{t_j - t_i},\ Q = G_{1j}^2,\ s_1 = \theta + \frac{\theta^2 Q}{2\lambda} - \frac{\theta}{2\lambda}\sqrt{4\theta\lambda Q + \theta^2 Q^2}.$$
   Generate a uniform random variable U_j on $(0,1)$. If $U_j < \frac{\theta(1+s_1)}{(1+\theta)(\theta+s_1)}$, $s = s_1$; otherwise, $s = \theta^2/s_1$. Then
   $$z_{t_j} = z_{t_i} + \frac{z_{t_k} - z_{t_i}}{1+s}.$$

(b) Brownian bridge (see [22] Chapter 3): conditional on $W_{z_{t_i}}$ and $W_{z_{t_k}}$, generate a standard normal random variable G_{2j}, and compute

$$m = \frac{(z_{t_k} - z_{t_j})W_{z_{t_i}} + (z_{t_j} - z_{t_i})W_{z_{t_k}}}{z_{t_k} - z_{t_i}}, \sigma^2 = \frac{(z_{t_j} - z_{t_i})(z_{t_k} - z_{t_j})}{z_{t_k} - z_{t_i}},\ W_{z_{t_j}} = m + \sigma G_{2j}.$$

3. Let $X_{t_j} = \mu t_j + W_{z_{t_j}}, 1 \le j \le d$.

Remark 5.

- In Algorithm 1, to generate X_t, three uniform random variables are needed: two for z_t, and one for the Brownian motion. In contrast, the inverse transform method only requires one uniform random variable. Similarly, in Algorithm 2, to generate a sample path with d Lévy increments, $3d$ uniform random variables are needed. The inverse transform method requires d uniform random variables only. We thus expect that the inverse transform method will be faster in general. This is verified in our numerical examples.

5.2 Kou's double exponential jump diffusion process

Kou's double exponential jump diffusion process is given by

$$X_t = \mu t + \sigma B_t + \sum_{i=1}^{N_t} Z_i,$$

where B_t is a standard Brownian motion, N_t is a Poisson process with intensity $\lambda > 0$, and $\{Z_i, i \ge 1\}$ are i.i.d. jump sizes. $B_t, N_t, \{Z_i, i \ge 1\}$ are independent. The jump sizes $\{Z_i, i \ge 1\}$ have the following density:

$$p\eta_1 e^{-\eta_1 x}\mathbf{1}_{\{x>0\}} + (1-p)\eta_2 e^{\eta_2 x}\mathbf{1}_{\{x<0\}},$$

where $0 \le p \le 1$ is the probability of a positive jump size. Positive jump sizes are exponential with intensity η_1. Negative jump sizes follow the negative of an exponential distribution with intensity η_2. By the martingale requirement, $\mu = r - q - \frac{1}{2}\sigma^2 - \lambda(p/(\eta_1 - 1) - (1-p)/(\eta_2 + 1))$.

The distribution of Kou's double exponential jump diffusion process is highly complex. However, its characteristic function admits a very simple expression:

$$\phi_t(\xi) = \exp\Big(-\frac{1}{2}\sigma^2 t\xi^2 + i\mu t\xi + i\lambda t\xi\Big(\frac{p}{\eta_1 - i\xi} - \frac{1-p}{\eta_2 + i\xi}\Big)\Big).$$

Moreover, it is in $H(\mathcal{D}_{(d_-,d_+)})$ for any $-\eta_1 < d_- < 0 < d_+ < \eta_2$. It satisfies the exponential tail condition (2.3) with $\kappa = 1, c = \frac{1}{2}\sigma^2 t, \nu = 2$.

Kou's double exponential process can be simulated directly by generating $B_t, N_t, Z_1, \cdots, Z_{N_t}$ separately. We would like to compare the inverse transform method with this direct algorithm, which is summarized below:

Algorithm 3 (Simulating Kou's double exponential jump diffusion)**.**

For $t > 0$, simulate $X_t = \mu t + \sigma B_t + \sum_{i=1}^{N_t} Z_i$

1. Generate a standard normal random variable G.

2. Generate the Poisson process N_t using the inverse transform method described on p.128 of [22].

3. Generate $Z_i, 1 \leq i \leq N_t$, in the following way: generate U_i that is uniform in $(0,1)$; if $U_i < p$, generate Z_i from an $\exp(\eta_1)$ distribution; otherwise, generate Z_i from the negative of an $\exp(\eta_2)$ distribution.

4. Let $X_t = \mu t + \sigma\sqrt{t}G + Z_1 + \cdots + Z_{N_t}$.

Obviously, to generate X_t using the above algorithm, $2(\lambda t + 1)$ uniform random variables are required on average. This is in contrast to one uniform random variable in the inverse transform method. When λt is large, this algorithm could be much more demanding. Recall that a pure jump Lévy process could also be simulated by first approximating it by a jump diffusion. Obviously, compared to the inverse transform method, such an approach could be expensive too since the intensity of the approximating jump diffusion could be large.

5.3 The CGMY process

A CGMY process X_t is a pure jump Lévy process with drift μ and the following Lévy density

$$\frac{Ce^{Gx}}{|x|^{1+Y}}\mathbf{1}_{\{x<0\}} + \frac{Ce^{-\mathbb{M}x}}{|x|^{1+Y}}\mathbf{1}_{\{x>0\}}$$

for some $C > 0, G > 0, \mathbb{M} > 0, 0 < Y < 2$. The martingale condition requires $\mu = r - q - C\Gamma(-Y)((\mathbb{M}-1)^Y - \mathbb{M}^Y + (G+1)^Y - G^Y)$. Explicit expressions for the density and the cdf of X_t are not available. However, the characteristic function of X_t is known explicitly:

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

where $\Gamma(\cdot)$ is the gamma function. ϕ is in $H(\mathcal{D}_{(d_-,d_+)})$ with $d_- = -\mathbb{M}, d_+ = G$. When $0 < Y < 1$, ϕ satisfies (2.3) with $\kappa = \exp(-tC\Gamma(-Y)(\mathbb{M}^Y + G^Y))$, $c = 2tC|\Gamma(-Y)\cos(\pi Y/2)|$, and $\nu = Y$. This allows us to construct (3.11) for a given bias tolerance level when the inverse transform method is used.

5.4 European vanilla options

European vanilla options pricing is used to test Theorem 3. For a European vanilla put option with maturity T and strike price K, the payoff is given by $\max(0, K - S_T)$. The price of such an option at time 0 is given by

$$V = e^{-rT}\mathbb{E}[\max(0, K - S_0e^{X_T}] = S_0e^{-rT}\mathbb{E}[f(X_T)],$$

where $X_t = \ln(S_t/S_0)$ is a Lévy process and $f(x) = \max(0, K/S_0 - e^x)$. When the inverse transform method is used, according to Theorem 2, one may simply take $x_K = \ln(K/S_0)$. The payoff of a call option is given by $\max(0, S_T - K)$. Call options can be priced similarly, and one can take $x_0 = \ln(K/S_0)$ for call options.

5.5 Floating strike lookback put options

We use discretely monitored floating strike lookback put options to illustrate Theorem 4 and to compare the inverse transform method with Algorithm 2 for a NIG process. The payoff of such an option with maturity T and monitoring interval $\Delta = T/d$ is given by $\max(S_0, S_\Delta, \cdots, S_{d\Delta}) - S_{d\Delta}$. The option price at time 0 is given by

$$V = e^{-rT}\mathbb{E}[\max(S_0, S_\Delta, \cdots, S_{d\Delta}) - S_{d\Delta}] = S_0 e^{-rT}\mathbb{E}[f(Y_1, \cdots, Y_d)],$$

where $Y_i = X_{i\Delta} - X_{(i-1)\Delta}, 1 \le i \le d$, and

$$f(y_1, \cdots, y_d) = \max(1, e^{y_1}, e^{y_1+y_2}, \cdots, e^{y_1+\cdots+y_d}) - e^{y_1+\cdots+y_d}.$$

Obviously,

$$|f(y_1, \cdots, y_d)| \le e^{|y_1|}e^{|y_2|}\cdots e^{|y_d|}.$$

For any $1 \le i \le d$, $f(y_1, \cdots, y_d)$ as a function y_i is

$$f(y_1, \cdots, y_d) = \max(a, be^{y_i}) - ce^{y_i},$$

where $a = \max(1, e^{y_1}, \cdots, e^{y_1+\cdots+y_{i-1}})$, $b = e^{y_1+\cdots+y_{i-1}}\max(1, e^{y_{i+1}}, \cdots, e^{y_{i+1}+\cdots+y_d})$, and $c = e^{y_1+\cdots+y_{i-1}+y_{i+1}+\cdots+y_d}$. Therefore, f as a function of y_i is differentiable except at one point. Moreover,

$$\Big|\frac{df}{dy_i}\Big| = \begin{cases} ce^{y_i}, & y_i < \ln(a/b) \\ (b-c)e^{y_i}, & y_i > \ln(a/b) \end{cases} \le e^{|y_1|}e^{|y_2|}\cdots e^{|y_d|}.$$

Theorem 4 is thus applicable with $g_i(x) = e^{|x|}$, $1 \le i \le d$, and $n^f = 1$.

5.6 Fixed strike Asian call options

We use discrete fixed strike Asian call options to illustrate the simulation of a CGMY process and the accuracy and speed of the inverse transform method when it is combined with quasi-Monte Carlo methods and variance reduction techniques. The payoff of such an option with maturity $T = d\Delta$ is given by $\max(0, A_T - K)$, where $A_T = \frac{1}{d}\sum_{k=1}^{d} S_{k\Delta}$. The price of the option is given by

$$V = e^{-rT}\mathbb{E}[\max(0, A_T - K)] = S_0 e^{-rT}\mathbb{E}[f(Y_1, \cdots, Y_d)],$$

where $Y_i = X_{i\Delta} - X_{(i-1)\Delta}, 1 \le i \le d$, and

$$f(y_1, \cdots, y_d) = \max\Big(0, \frac{1}{d}(e^{y_1} + e^{y_1+y_2}\cdots + e^{y_1+\cdots+y_d}) - \frac{K}{S_0}\Big).$$

Obviously,

$$|f(y_1, \cdots, y_d)| \le e^{|y_1|}e^{|y_2|}\cdots e^{|y_d|}.$$

For any $1 \le i \le d$, $f(y_1, \cdots, y_d)$ as a function y_i is

$$f(y_1, \cdots, y_d) = \max(0, a + be^{y_i}),$$

where $a = \frac{1}{d}(e^{y_1} + e^{y_1+y_2} + \cdots + e^{y_1+\cdots+y_{i-1}}) - K/S_0$, $b = \frac{1}{d}e^{y_1+\cdots+y_{i-1}}(1 + e^{y_{i+1}} + \cdots + e^{y_{i+1}+\cdots+y_d})$. If $a \ge 0$, f as a function of y_i is differentiable everywhere and satisfies

$$\Big|\frac{df}{dy_i}\Big| = be^{y_i} \le e^{|y_1|}e^{|y_2|}\cdots e^{|y_d|}.$$

If $a < 0$, f as a function of y_i is differentiable except at one point:

$$\left|\frac{df}{dy_i}\right| = \begin{cases} 0, & y_i < \ln(-a/b) \\ be^{y_i}, & y_i > \ln(-a/b) \end{cases} \leq e^{|y_1|}e^{|y_2|}\cdots e^{|y_d|}.$$

Theorem 4 is thus applicable with $g_i(x) = e^{|x|}$, $1 \leq i \leq d$, and $n^f = 1$.

Most traded Asian options assume arithmetic average described above. Closed form solutions usually do not exist for such contracts. However, if the average is computed in the geometric sense, option prices can be computed very fast and accurately using the Hilbert transform method. Geometric Asian options serve as excellent control variates for the pricing of arithmetic Asian options. We describe geometric Asian options pricing below and briefly review the control variates technique.

5.6.1 Geometric Asian options

The price of a discrete geometric Asian call option with maturity $T = d\Delta$ and strike K is given by the following:

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

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

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

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

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

Therefore,

$$\begin{aligned} V^g &= e^{-rT}\mathbb{E}[(S_0e^Y - K)\mathbf{1}_{\{Y>\ln(K/S_0)\}}] \\ &= e^{-rT}S_0\mathbb{E}[e^Y\mathbf{1}_{\{Y>\ln(K/S_0)\}}] - Ke^{-rT}\mathbb{P}(Y > \ln(K/S_0)). \end{aligned} \tag{5.21}$$

The second term above can be computed using Theorem 1 directly. Using a standard change of measure technique (see [17] for more details), the first term can also be turned into a probability and hence computed using Theorem 1:

$$\mathbb{E}[e^Y\mathbf{1}_{\{Y>\ln(K/S_0)\}}] = \phi_Y(-i)\mathbb{P}^*(Y > \ln(K/S_0)),$$

where under the new measure $\mathbb{P}^*$, Y has a characteristic function

$$\phi_Y^*(\xi) = \mathbb{E}^*[e^{i\xi Y}] = \frac{\phi_Y(\xi - i)}{\phi(-i)}.$$

Geometric Asian put options can be priced similarly. In summary, for Lévy models with analytic characteristic functions that decay exponentially, the price of a geometric Asian option can be computed highly accurately using the Hilbert transform method.

5.6.2 Control variates

To compute $V = e^{-rT}\mathbb{E}[\max(0, A_T - K)]$ using control variates, we simulate d Lévy increments and hence a Lévy path $\{X^n_\Delta, \cdots, X^n_{d\Delta}\}$ and compute the payoff of the arithmetic Asian option V_n and that of the geometric Asian option V^g_n. The discrepancy between V^g_n and $\mathbb{E}[\max(0, A^g_T - K)]$ can be used to adjust V_n in the following way so that the resulting quantity is closer to $\mathbb{E}[\max(0, A_T - K)]$:

$$V^b_n := V_n + b(\mathbb{E}[\max(0, A^g_T - K)] - V^g_n).$$

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

$$\bar{V}^b_N := \frac{1}{N}\sum_{n=1}^{N} V^b_n.$$

By the law of large numbers, $\bar{V}^b_N$ converges to $\mathbb{E}[\max(0, A_T - K)]$ as $N \to +\infty$. Moreover, let σ_a denote the standard deviation of $\max(0, A_T - K)$, σ_g the standard derivation of $\max(0, A^g_T - K)$, 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 $\bar{V}^b_N$ is reduced to

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

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^2 is close to 1, 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, an estimate can be obtained by simulating a predetermined number (e.g., 1000) of pairs of (V_n, V^g_n) and replacing the correlation coefficient and standard deviations with sample correlation coefficient and sample standard deviations.

6 Numerical results

In this section, we verify the theoretical results in Section 4, and illustrate how the probability grid (3.11) and numerical parameters for characteristic function inversion can be determined. We show that the inverse transform method could be several times faster than alternative schemes for the NIG process and Kou's jump diffusion. We then illustrate the efficiency of the combination of the inverse transform method, quasi-Monte Carlo schemes and control variates techniques by pricing an Asian option in the CGMY model. Computations are done on a Lenovo laptop T61p with Intel Core 2 Duo 2.5GHz CPU and 3GB RAM.

6.1 The normal inverse Gaussian process

We would like to compare the performance of the inverse transform method and Algorithms 1, 2 for the NIG process. We first consider a European vanilla put option. We use the same parameters as in [18]:

$$\alpha = 15, \beta = -5, \delta = 0.5, r = 0.05, q = 0.02, S_0 = K = 100, T = 0.5.$$

European put in the NIG model

ϵ_b	x_0	x_K	K	$E_{h,M,\mathcal{X}}$	h	M
10^{-2}	-0.477	0	22	5.5×10^{-6}	4.926	11
10^{-3}	-0.736	0	133	7.1×10^{-8}	3.630	20
10^{-4}	-0.983	0	645	1.2×10^{-9}	2.903	30

Lookback put in the NIG model

ϵ_b	x_0	x_K	K	$E_{h,M,\mathcal{X}}$	h	M
10^{-2}	-0.721	0.340	273	8.8×10^{-8}	3.127	76
10^{-3}	-0.977	0.461	1546	1.2×10^{-9}	2.456	123
10^{-4}	-1.233	0.582	7875	1.8×10^{-11}	2.028	180

European put in Kou's model

ϵ_b	x_0	x_K	K	$E_{h,M,\mathcal{X}}$	h	M
10^{-2}	-0.715	0	31	3.0×10^{-6}	3.651	13
10^{-3}	-1.029	0	167	4.6×10^{-8}	2.803	19
10^{-4}	-1.332	0	774	8.8×10^{-10}	2.295	27

Asian call in the CGMY model ($d = 6$)

ϵ_b	x_0	x_K	K	$E_{h,M,\mathcal{X}}$	h	M
10^{-2}	-0.213	0.205	44	8.7×10^{-7}	10.899	21
10^{-3}	-0.260	0.244	188	2.0×10^{-8}	9.107	28
10^{-4}	-0.307	0.283	771	4.8×10^{-10}	7.822	35

Asian call in the CGMY model ($d = 26$)

ϵ_b	x_0	x_K	K	$E_{h,M,\mathcal{X}}$	h	M
10^{-2}	-0.129	0.111	45	9.3×10^{-7}	15.748	31
10^{-3}	-0.176	0.150	226	1.8×10^{-8}	12.187	49
10^{-4}	-0.223	0.189	1039	3.8×10^{-10}	9.954	73

Table 1: Grids and parameters for characteristic functions inversion.

We take $x_K = \ln(K/S_0) = 0$ and determine $x_0 < 0$ according to the second term in (4.17):

$$\frac{||\phi_T||^+}{2\pi}\Big(\int_{-\infty}^{x_0}|f(x)|e^{xd_+}dx + \frac{1}{d_+}|f(x_0)|e^{x_0d_+}\Big) = \frac{||\phi_T||^+}{2\pi d_+}e^{x_0d_+}\Big(2 - \frac{2d_+ + 1}{d_+ + 1}e^{x_0}\Big).$$

For three tolerance levels of total bias $\epsilon_b = 10^{-2}, 10^{-3}, 10^{-4}$, we find the largest x_0 so that the above is bounded by $\epsilon_b/2$. One may further relax the right hand side in the above to $||\phi_T||^+e^{x_0d_+}/(\pi d_+)$, which allows one to compute x_0 directly. Alternatively, one may use a standard root finding procedure to find the desired x_0, which obviously will be tighter. For the above NIG process, $d_- = \beta - \alpha = -20, d_+ = \beta + \alpha = 10$. Correspondingly, $||\phi_T||^- \approx 751.318$, $||\phi_T||^+ \approx 45.223$. With the root finding approach, we obtain x_0 as reported in Table 1. We then select the step size $(x_K - x_0)/K$ of the grid according to the third term in (4.17):

$$\frac{1}{2\pi K^2}||f'||_{\mathcal{X}}|\mathcal{X}|^3\int_{\mathbb{R}}|\xi\phi_T(\xi)|d\xi.$$

The smallest positive integer K so that the above is bounded by $\epsilon_b/2$ is given by:

$$K = \Big(\frac{1}{\pi\epsilon_b}||f'||_{\mathcal{X}}|\mathcal{X}|^3\int_{\mathbb{R}}|\xi\phi_T(\xi)|d\xi\Big)^{1/2}.$$

It is easy to find that $||f'||_{\mathcal{X}} = 1$, $\int_{\mathbb{R}}|\xi\phi(\xi)|d\xi \approx 137.505$. The corresponding values for K are reported in Table 1. We then control the last term in (4.17):

$$(|f(x_0)|+|f(x_K)|+2K||f||_{\mathcal{X}}+2||f'||_{\mathcal{X}}(x_K-x_0))E_{h,M,\mathcal{X}} = ((2K+1)(1-e^{x_0})+2(x_K-x_0))E_{h,M,\mathcal{X}}.$$

We make the above equal to $0.01\epsilon_b$, which is negligible when compared to ϵ_b. The corresponding values for $E_{h,M,\mathcal{X}}$ can be found in Table 1. According to (2.8), since $x_K = 0$, we have

$$\begin{aligned} E_{h,M,\mathcal{X}} \quad \leq \quad & \frac{e^{-2\pi|d_-|/h+x_0d_-}}{2\pi|d_-|(1-e^{-2\pi|d_-|/h})}||\phi_T||^- + \frac{e^{-2\pi d_+/h}}{2\pi d_+(1-e^{-2\pi d_+/h})}||\phi_T||^+ \\ & + \frac{\kappa}{2\pi}\Big(\frac{1}{M} + \frac{4}{\nu c(Mh)^\nu}\Big)e^{-c(Mh)^\nu}. \end{aligned} \tag{6.22}$$

Recall that for the NIG process, $\kappa = \exp(\delta t\sqrt{\alpha^2 - \beta^2}), c = \delta t, \nu = 1$. We then solve for h and M so that the summation of the first two terms above and the third term above are equal to one half of the value in the fifth column of Table 1 respectively. The resulting h and M are reported in Table 1.

With the settings in Table 1, we price the European vanilla put option. The benchmark price is computed using the Hilbert transform method of [17] to be 4.58980916. The results are shown in Table 2. "N" refers to the sample size in thousands. "SE" represents the standard error. "Error" represents the actual pricing error. "CPU" represents the computational time in seconds. The columns labeled "Gain" record the efficiency gains of the inverse transform method when compared with Algorithm 1. The efficiency gain is computed as follows:

$$\frac{\sigma_A^2 \cdot t_A}{\sigma_B^2 \cdot t_B},$$

where σ_B and t_B are the standard error and computational time when the inverse transform method is used, and σ_A and t_A are the corresponding values when the method that is compared to (Algorithm 1 in the current case) is used. We observe the following:

	Algorithm 1				$\epsilon_b = 10^{-2}$			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
64	3.0E-2	3.0E-2	0.036		2.9E-2	7.1E-3	0.007	5.5
256	1.5E-2	3.6E-2	0.139		1.5E-2	7.2E-3	0.026	5.3
1024	7.4E-3	1.4E-2	0.545		7.3E-3	7.8E-3	0.096	5.8
4096	3.7E-3	1.1E-2	2.191		3.7E-3	7.6E-3	0.387	5.7
16384	1.9E-3	1.8E-3	8.640		1.8E-3	7.1E-3	1.576	6.1
65536	9.3E-4	5.9E-4	34.382		9.2E-4	7.4E-3	6.162	5.7
	$\epsilon_b = 10^{-3}$				$\epsilon_b = 10^{-4}$			
64	3.0E-2	4.8E-2	0.009	4.0	3.0E-2	3.2E-2	0.016	2.3
256	1.5E-2	3.2E-4	0.029	4.8	1.5E-2	6.3E-3	0.040	3.5
1024	7.4E-3	6.8E-3	0.111	4.9	7.4E-3	1.3E-3	0.132	4.1
4096	3.7E-3	5.9E-3	0.444	4.9	3.7E-3	5.1E-4	0.494	4.4
16384	1.9E-3	1.7E-3	1.777	4.9	1.9E-3	1.1E-4	1.965	4.4
65536	9.3E-4	1.5E-3	7.108	4.8	9.3E-4	2.8E-4	7.745	4.4

Table 2: European vanilla put in the NIG model: inverse transform method vs Algorithm 1.

- When the sample size N is large enough, the bias will dominate. For example, in the case with $\epsilon_b = 10^{-2}$, we find by looking at the bottom of the "Error" column that the actual pricing bias of the inverse transform method is likely between 7×10^{-3} and 8×10^{-3}. This is smaller than our target level, which is $S_0 e^{-rT} \epsilon_b$ (recall that ϵ_b is the bias tolerance level for $\mathbb{E}[f(X_T)]$ and the option price is $V = S_0 e^{-rT} \mathbb{E}[f(X_T)]$).
- For practical levels of accuracy (e.g., 1 cent or 0.1 cent), the inverse transform method is several times faster than Algorithm 1. The performance of the inverse transform method will be even better in the above example if the approach described in Remark 2 is used.

Now we consider a one-year ($T = 1$) floating strike lookback put option in the NIG model with the same parameters as above. We assume $d = 8$ monitoring intervals. The length of each monitoring interval is $\Delta = T/d = 1/8$. According to Sections 4.3 and 5, we select the grid by bounding (4.20) with $n^f = 1$, where $g_i(x) = e^{|x|}, 1 \le i \le d$. When determining x_0 and x_K, we bound each of the first two terms in (4.20) by $\epsilon_b/4$ and proceed in similar ways as in the European vanilla put case. Note that we are simulating Lévy increments repeatedly on time intervals of length Δ. The corresponding characteristic function that is used in (4.20) is thus ϕ_Δ. The grids and the numerical parameters for characteristic function inversion corresponding to $\epsilon_b = 10^{-2}, 10^{-3}, 10^{-4}$ are reported in Table 1. With these settings, we compare the inverse transform method with Algorithm 2. Both are implemented using the randomized quasi-Monte Carlo (RQMC) method discussed in Section 3.2. [28] suggests a batch number L that is between 5 and 25. In our implementations, we use $L = 10$. The sample size N reported in Table 3 refers to LN_0 - the total number of Lévy paths generated (see Section 3). The benchmark price for the lookback put is computed to be 10.18611401 using the method of [20]. Since K is relatively large in this example, we use the Toeplitz matrix vector multiplication algorithm discussed previously to compute the probabilities. For example, for the case with $\epsilon_b = 10^{-4}$, this algorithm is able to reduce the computational time from 0.538 seconds to 0.012 seconds to obtain all the $K+1 = 7876$ probabilities to an accuracy of 1.8×10^{-11}. We observe the following from Table 3:

	Algorithm 2 with RQMC					$\epsilon_b = 10^{-2}$				
N ($\times 10^3$)	SE	Error	CPU	Price	Gain	SE	Error	CPU	Price	Gain
16	1.5E-2	1.2E-2	0.049	10.1985		1.8E-2	2.4E-3	0.024	10.1837	1.4
64	1.2E-2	9.0E-3	0.186	10.1951		6.4E-3	7.6E-3	0.085	10.1937	7.7
256	3.5E-3	3.3E-3	0.737	10.1828		2.6E-3	1.9E-3	0.332	10.1880	4.0
1024	1.9E-3	1.4E-3	2.847	10.1847		6.3E-4	1.2E-3	1.320	10.1873	19.6
4096	4.9E-4	7.0E-4	11.446	10.1854		1.8E-4	2.8E-3	5.290	10.1889	16.0
16384	2.0E-4	8.2E-5	45.787	10.1862		7.4E-5	2.6E-3	21.221	10.1887	15.8
65536	9.5E-5	7.0E-5	183.005	10.1862		4.3E-5	2.5E-3	84.185	10.1886	10.6
	$\epsilon_b = 10^{-3}$					$\epsilon_b = 10^{-4}$				
16	1.3E-2	3.6E-2	0.028	10.1500	2.3	1.6E-2	1.4E-2	0.040	10.1717	1.1
64	5.8E-3	2.0E-3	0.100	10.1842	8.0	4.9E-3	1.8E-3	0.113	10.1879	9.9
256	2.0E-3	1.1E-3	0.373	10.1873	6.1	1.3E-3	1.7E-3	0.423	10.1844	12.6
1024	9.6E-4	4.1E-4	1.464	10.1857	7.6	8.0E-4	7.5E-4	1.650	10.1869	9.7
4096	3.2E-4	3.9E-4	5.818	10.1865	4.6	1.6E-4	2.4E-4	6.574	10.1864	16.3
16384	7.5E-5	3.9E-5	23.307	10.1862	14.0	5.0E-5	4.6E-5	25.973	10.1862	28.2
65536	4.1E-5	1.2E-4	94.097	10.1862	10.4	3.6E-5	1.0E-5	103.713	10.1861	12.3

Table 3: Discrete floating strike lookback put in the NIG model: inverse transform method with RQMC vs Algorithm 2 with RQMC.

- The actual bias of this multidimensional problem decreases as we reduce the tolerance level ϵ_b for B, as suggested by Theorem 4. This can be seen at least when we reduce ϵ_b from 10^{-2} to 10^{-3}.
- Compared to Algorithm 2 with the inverse Gaussian bridge, the inverse transform method without a bridge could still be many times better.

6.2 Kou's double exponential jump diffusion process

We now compare the performance of the inverse transform method and Algorithm 3 for Kou's double exponential jump diffusion process. We consider a European vanilla put option in Kou's model with the same parameters as in [18]:

$$\sigma = 0.1, \lambda = 3, p = 0.3, \eta_1 = 40, \eta_2 = 12, r = 0.05, q = 0.02, S_0 = K = 100, T = 1.$$

We take $x_K = \ln(K/S_0) = 0$ and determine $x_0 < 0$ in the same way as in the previous section. Recall that in Kou's model, any number in $(0, \eta_2)$, but not η_2 itself, could be used as d_+. We test a few d_+ that are smaller than 12 and select the tightest x_0 that we obtain. K, $E_{h,M,\mathcal{X}}$, h and M are determined similarly and reported in Table 1. The benchmark price is computed to be 5.98007999. Numerical results are reported in Table 4. The observations are similar to the case of the NIG process: the inverse transform method is several times better in this example for practically relevant levels of accuracy, and the actual pricing bias is below the target level $S_0 e^{-rT} \epsilon_b$.

As discussed previously, a Lévy process may be approximated by a jump diffusion process first and then simulated. However, the resulting jump diffusion process may have a large jump arrival rate, and the jump size may not be as easy to simulate as double exponential jump sizes. The inverse transform method could thus be more attractive than such an approach.

	Algorithm 3				$\epsilon_b = 10^{-2}$			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
64	4.0E-2	3.5E-2	0.021		4.0E-2	7.1E-3	0.008	2.6
256	2.0E-2	3.8E-2	0.082		2.0E-2	2.8E-2	0.022	3.7
1024	1.0E-2	5.1E-3	0.335		1.0E-2	1.6E-2	0.075	4.5
4096	5.0E-3	4.7E-3	1.332		5.0E-3	1.1E-2	0.297	4.5
16384	2.5E-3	2.2E-3	5.380		2.5E-3	1.4E-2	1.201	4.5
65536	1.3E-3	2.0E-3	21.270		1.2E-3	1.3E-2	4.846	5.2
	$\epsilon_b = 10^{-3}$				$\epsilon_b = 10^{-4}$			
64	4.0E-2	5.0E-2	0.008	2.6	4.0E-2	2.6E-2	0.015	1.4
256	2.0E-2	2.8E-2	0.024	3.4	2.0E-2	3.2E-2	0.033	2.5
1024	1.0E-2	3.3E-2	0.089	3.8	1.0E-2	1.5E-3	0.105	3.2
4096	5.0E-3	9.9E-3	0.341	3.9	5.0E-3	8.1E-3	0.394	3.4
16384	2.5E-3	1.4E-3	1.349	4.0	2.5E-3	2.8E-3	1.512	3.6
65536	1.3E-3	3.4E-3	5.377	4.0	1.3E-3	4.0E-4	6.029	3.5

Table 4: European vanilla put in Kou's model: inverse transform method vs Algorithm 3.

6.3 The CGMY process

In this section, we consider simulating a CGMY process. Such a process does not admit simple structures as those for Kou's process and the NIG process. However, the inverse transform method only uses the characteristic function and hence is applicable. We illustrate the effectiveness of the method by pricing a discrete arithmetic fixed strike Asian call option. The parameters are the same as those in [18]:

$$C = 4, G = 50, \mathbb{M} = 60, Y = 0.7, r = 0.05, q = 0.02, S_0 = K = 100, T = 0.5.$$

The average asset price is computed based on either monthly monitoring ($d = 6$) or weekly monitoring ($d = 26$). Benchmark prices are computed using the Fourier transform method [6, 13] to be $V = 4.00703627$ for monthly monitoring and $V = 3.65349339$ for weekly monitoring. For $d = 6, 26$ and $\epsilon_b = 10^{-2}, 10^{-3}, 10^{-4}$ (the tolerance level for B in Theorem 4), we determine the grids and the numerical parameters for characteristic function inversion as before. They are reported in Table 1.

We compare standard Mote Carlo and randomized quasi-Monte Carlo methods with or without control variates. For the control variates, we use the corresponding discrete geometric Asian options. When $d = 6$, the geometric Asian option price is computed to be $V^g = 3.91754467$ using the method described in Section 5.6.1. When $d = 26$, the geometric Asian option price is $V^g = 3.56206157$. We first investigate the bias of the inverse transform method. For this, we use the randomized quasi-Monte Carlo method with control variates (RQMC-CV). Implementation details will be given in the next paragraph. Table 5 shows that as we decrease ϵ_b from 10^{-2} to 10^{-4}, we obtain very accurate approximations to the Asian option prices for both monitoring frequencies. The decreasing of the bias can be clearly seen when we decrease ϵ_b from 10^{-2} to 10^{-3}.

We then fix ϵ_b at 10^{-3} in Table 6 and compare four different schemes: the standard Monte Carlo method (MC), the standard Monte Carlo method with control variates (MC-CV), the randomized quasi-Monte Carlo method (RQMC), and the randomized quasi-Monte Carlo method

$d = 6$

	$\epsilon_b = 10^{-2}$			$\epsilon_b = 10^{-3}$			$\epsilon_b = 10^{-4}$		
N ($\times 10^3$)	SE	Error	Price	SE	Error	Price	SE	Error	Price
1	1.5E-3	2.4E-4	4.0068	1.8E-3	2.0E-3	4.0090	2.4E-3	3.7E-4	4.0067
4	4.9E-4	5.9E-4	4.0076	4.7E-4	1.7E-3	4.0088	7.8E-4	1.6E-4	4.0072
16	2.5E-4	7.3E-5	4.0071	3.4E-4	3.6E-4	4.0067	3.8E-4	3.0E-4	4.0073
64	1.2E-4	1.4E-4	4.0072	7.8E-5	1.6E-4	4.0072	1.4E-4	1.9E-4	4.0072
256	3.5E-5	2.6E-4	4.0068	5.9E-5	5.9E-6	4.0070	4.0E-5	7.9E-5	4.0070
1024	3.0E-5	3.0E-4	4.0073	2.5E-5	1.5E-5	4.0070	2.8E-5	3.9E-5	4.0071
4096	1.6E-5	9.7E-4	4.0080	8.6E-6	2.4E-5	4.0071	8.2E-6	1.7E-5	4.0070
16384	4.6E-6	8.7E-4	4.0079	3.9E-6	2.2E-5	4.0071	2.9E-6	3.6E-6	4.0070
65536	1.6E-6	9.4E-4	4.0080	1.9E-6	4.9E-6	4.0070	2.3E-6	1.0E-6	4.0070

$d = 26$

	$\epsilon_b = 10^{-2}$			$\epsilon_b = 10^{-3}$			$\epsilon_b = 10^{-4}$		
N ($\times 10^3$)	SE	Error	Price	SE	Error	Price	SE	Error	Price
1	2.1E-3	1.8E-3	3.6553	2.5E-3	2.6E-4	3.6537	2.6E-3	5.2E-3	3.6587
4	8.4E-4	1.2E-3	3.6547	8.8E-4	8.3E-4	3.6527	1.1E-3	2.1E-4	3.6537
16	6.3E-4	2.6E-4	3.6537	5.2E-4	1.6E-4	3.6537	4.8E-4	4.1E-4	3.6539
64	1.9E-4	5.5E-5	3.6534	1.9E-4	1.4E-4	3.6534	1.6E-4	8.1E-5	3.6534
256	5.2E-5	9.0E-5	3.6536	8.5E-5	3.4E-4	3.6532	7.0E-5	3.6E-5	3.6535
1024	3.4E-5	6.3E-5	3.6536	2.2E-5	4.5E-5	3.6535	2.2E-5	5.5E-6	3.6535
4096	1.6E-5	1.2E-4	3.6536	1.9E-5	2.4E-5	3.6535	1.7E-5	9.8E-6	3.6535
16384	4.1E-6	1.8E-4	3.6537	9.7E-6	1.2E-5	3.6535	8.7E-6	7.7E-6	3.6535
65536	3.9E-6	5.8E-5	3.6536	4.3E-6	5.5E-6	3.6535	2.6E-6	4.5E-6	3.6535

Table 5: Discrete Asian call in the CGMY model: randomized quasi-Monte Carlo ($L = 10$) with control variates.

Monthly monitoring ($d = 6$)

	MC				MC-CV			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
1	1.8E-1	5.8E-2	0.004		3.2E-3	2.4E-3	0.004	3164
4	9.2E-2	8.1E-2	0.006		1.7E-3	2.3E-3	0.006	2929
16	4.6E-2	3.1E-2	0.015		8.7E-4	8.9E-4	0.016	2621
64	2.3E-2	3.8E-2	0.046		4.5E-4	8.4E-5	0.054	2225
256	1.2E-2	2.6E-2	0.173		2.3E-4	1.7E-4	0.204	2308
1024	5.8E-3	1.0E-2	0.687		1.1E-4	9.2E-5	0.805	2373
4096	2.9E-3	4.1E-3	2.757		5.6E-5	4.9E-5	3.215	2300
16384	1.4E-3	4.7E-4	10.906		2.8E-5	1.9E-5	12.764	2136
65536	7.2E-4	8.1E-4	43.234		1.4E-5	2.6E-5	50.907	2246
	RQMC				RQMC-CV			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
1	5.8E-2	7.5E-2	0.004	10	1.8E-3	2.0E-3	0.005	8000
4	2.1E-2	2.9E-2	0.007	16	4.7E-4	1.7E-3	0.009	25544
16	6.7E-3	4.4E-5	0.017	42	3.4E-4	3.6E-4	0.019	14451
64	2.1E-3	2.8E-3	0.054	102	7.8E-5	1.6E-4	0.064	62495
256	5.1E-4	7.3E-5	0.208	460	5.9E-5	5.9E-6	0.241	29695
1024	1.4E-4	4.7E-4	0.864	1365	2.5E-5	1.5E-5	0.959	38558
4096	7.1E-5	3.3E-4	3.288	1399	8.6E-6	2.4E-5	3.835	81747
16384	3.1E-5	4.1E-4	13.017	1709	3.9E-6	2.2E-5	15.302	91843
65536	1.2E-5	3.9E-4	51.935	2997	1.9E-6	4.9E-6	61.222	101409

Weekly monitoring ($d = 26$)

	MC				MC-CV			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
1	1.7E-1	6.1E-2	0.009		3.0E-3	6.3E-4	0.016	1806
4	8.6E-2	8.3E-2	0.021		1.6E-3	1.6E-3	0.022	2758
16	4.2E-2	1.4E-2	0.053		7.8E-4	1.5E-4	0.061	2519
64	2.1E-2	1.3E-3	0.187		4.0E-4	1.6E-4	0.209	2466
256	1.1E-2	3.4E-3	0.736		2.0E-4	2.2E-4	0.825	2699
1024	5.2E-3	2.0E-3	2.993		1.0E-4	4.2E-5	3.279	2468
4096	2.6E-3	4.0E-4	11.776		5.0E-5	1.4E-5	13.032	2443
16384	1.3E-3	1.2E-3	46.944		2.5E-5	8.5E-6	51.760	2452
65536	6.6E-4	1.2E-3	186.393		1.2E-5	2.0E-5	207.167	2722
	RQMC				RQMC-CV			
N ($\times 10^3$)	SE	Error	CPU	Gain	SE	Error	CPU	Gain
1	9.0E-2	5.4E-2	0.011	3	2.5E-3	2.6E-4	0.010	4162
4	3.7E-2	4.5E-3	0.020	6	8.8E-4	8.3E-4	0.021	9551
16	1.2E-2	6.4E-3	0.062	10	5.2E-4	1.6E-4	0.066	5239
64	2.8E-3	2.6E-3	0.219	48	1.9E-4	1.4E-4	0.268	8524
256	1.0E-3	3.9E-4	0.862	103	8.5E-5	3.4E-4	0.974	12655
1024	6.5E-4	5.5E-4	3.464	55	2.2E-5	4.5E-5	3.866	43252
4096	2.5E-4	5.6E-4	13.882	92	1.9E-5	2.4E-5	15.446	14276
16384	1.4E-4	4.5E-5	54.602	74	9.7E-6	1.2E-5	61.938	13613
65536	4.0E-5	2.1E-4	218.003	233	4.3E-6	5.5E-6	244.542	17957

Table 6: Discrete Asian call in the CGMY model: standard and quasi-Monte Carlo methods with or without control variates.

with control variates (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 batch number L is small, 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 method, we increase the sample size N and record the standard error ("SE"), the absolute pricing error ("Error"), the computational time ("CPU"), and the efficiency gain ("Gain") when compared with the standard Monte Carlo method. For MC and MC-CV, $N = 1000, 4000, \cdots, 65536000$ refers to the sample size (the total number of Lévy paths generated) and the SE is computed using these N realizations. For RQMC and RQMC-CV, we use $L = 10$ batches and for each batch we use a sequence of length $N_0 = 100, 400, \cdots, 6553600$. Therefore, $N = LN_0$ again refers to the total number of Lévy paths that are generated. The standard error "SE" is computed from the L estimates we obtain.

We observe that the standard Monte Carlo method is less accurate in general. The randomized quasi-Monte Carlo method greatly improves the performance, in particular for $d = 6$. The improvement is less significant for $d = 26$. It is well known that quasi-Monte Carlo methods worsen when the dimension of the problem increases. We also observe that when geometric Asian options are used as control variates, the performance of the simple Monte Carlo method greatly improves. The MC-CV method is even better than the RQMC method. Finally, the RQMC method with control variates outperforms all other methods and is tens of thousands of times better than the standard Monte Carlo method.

7 Conclusions

In this paper, we show that the inverse transform method is effective for simulating from distributions with analytic characteristic functions. It is very efficient when combined with the Hilbert transform method for characteristic function inversion, quasi-Monte Carlo methods and appropriate variance reduction techniques. The explicit bound we obtain for the estimation bias allows us to determine the grid where probabilities are computed and numerical parameters for characteristic function inversion. Examples show that the method is attractive for options pricing in Lévy models. The results we obtain can be applied to simulation from tabulated distribution functions in general and characteristic functions in our analytic class in particular.

References

[1] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10(1-2):5–88, 1992.

[2] S. Asmussen and P. W. Glynn. *Stochastic Simulation*. Springer-Verlag, Berlin, Germany, 2007.

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

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

[5] O. E. Barndorff-Nielsen. Process of normal inverse Gaussian type. *Finance and Stochastics*, 2(1):41–68, 1998.

[6] E. Benhamou. Fast Fourier transform for discrete Asian options. *Journal of Computational Finance*, 6(1), 2002.

[7] F. Benth, M. Groth, and P. Kettler. 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, 2006.

[8] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

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

[10] S. I. Boyarchenko and S. Z. Levendorskiĭ. *Non-Gaussian Merton-Black-Scholes Theory*. World Scientific, Singapore, 2002.

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

[12] P. Carr and L. Wu. Finite moment log stable process and option pricing. *Journal of Finance*, 58(2):753–777, 2003.

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

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

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

[16] E. Eberlein, U. Keller, and K. Prause. New insights into smile, mispricing, and value at risk: the hyperbolic model. *Journal of Business*, 71(3):371–405, 1998.

[17] L. Feng and X. Lin. Inverting analytic characteristic functions and financial applications. *Under review*, 2011.

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

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

[20] L. Feng and V. Linetsky. Computing exponential moments of the discrete maximum of a Lévy process and lookback options. *Finance and Stochastics*, 13(4):501–529, 2009.

[21] M. C. Fu. Variance-gamma and Monte Carlo. In M. C. Fu, R. A. Jarrow, J. Yen, and R. J. Elliott, editors, *Advances in Mathematical Finance*, pages 21–35. Birkh Boston, 2007.

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

[23] P. Glasserman and Z. Liu. Sensitivity estimates from characteristic functions. In S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, editors, *Proceedings of the 207 Winter Simulation Conference*, pages 932–940. 2007.

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

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

[26] S. G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

[27] A. Kyprianou. *Introductory Lectures on Fluctuations of Lévy Processes with Applications*. Springer-Verlag, Berlin, Germany, 2006.

[28] P. L'Ecuyer. Quasi-Monte Carlo methods in finance. In R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, editors, *Proceedings of the 2004 Winter Simulation Conference*. 2004.

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

[30] E. Lukacs. *Characteristic Functions*. Charles Griffin & Company Limited, London, second edition, 1970.

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

[32] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.

[33] R. C. Merton. Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144, 1976.

[34] N. Prabhu. *Stochastic Storage Processes: Queues, Insurance Risk, Dams, and Data Communication*. Springer-Verlag, Berlin, Germany, second edition, 1998.

[35] C. Ribeiro and N. Webber. A Monte Carlo method for the normal inverse Gaussian option valuation model using an inverse Gaussian bridge. *Working paper*, 2003.

[36] C. Ribeiro and N. Webber. Valuing path-dependent options in the variance-gamma model by Monte Carlo with a gamma bridge. *Journal of Computational Finance*, 7(2):81–100, 2003.

[37] J. Rosiński. Series representations of Lévy processes from the perspective of point processes. In O. Barndorff-Nielsen, T. Mikosch, and S. Resnick, editors, *Lévy Processes – Theory and Applications*, pages 401–415. Birkhäuser, Boston, 2001.

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

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

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

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