A QUASI-MONTE CARLO ALGORITHM FOR THE NORMAL INVERSE GAUSSIAN DISTRIBUTION AND VALUATION OF FINANCIAL DERIVATIVES

FRED ESPEN BENTH, MARTIN GROTH, AND PAUL C. KETTLER

Abstract. We propose a quasi-Monte Carlo (qMC) algorithm to simulate variates from the normal inverse Gaussian (NIG) distribution. The algorithm is based on a Monte Carlo technique found in Rydberg (Rydberg 1997), and is based on sampling three independent uniform variables. We apply the algorithm to three problems appearing in finance. First, we consider the valuation of plain vanilla call options and Asian options. The next application considers the problem of deriving implied parameters for the underlying asset dynamics based on observed option prices. We employ our proposed algorithm together with the Newton Method, and show how we can find the scale parameter of the NIG-distribution of the logreturns in case of a call or an Asian option. We also provide an extensive error analysis for this method. Finally we study the calculation of Value-at-Risk for a portfolio of nonlinear products where the returns are modeled by NIG random variables.

1. Introduction

The fair price of a financial derivative can be expressed in terms of a risk-neutral expectation of a random pay-off. In some cases the expectation is explicitly computable, the Black & Scholes formula for call options on assets modeled by a geometric Brownian motion being the prime example. However, considering for instance an Asian option, there exists no longer closed form expressions for the price, and numerical methods are called for. This may even be the case when considering plain vanilla call options written on assets with non-normal returns. In the present paper we propose a quasi-Monte Carlo algorithm for the valuation of expectations of functionals of normal inverse Gaussian distributed random variables.

Barndorff-Nielsen (Barndorff-Nielsen 1998) proposed to model the log-returns of asset prices by using the normal inverse Gaussian (NIG) distribution. This family of distributions has proven to fit the semi-heavy tails observed in financial time series of various kinds extremely well (see, *e.g.*, (Rydberg 1997) or (Eberlein and Keller 1995) who apply the hyperbolic distribution, being a close relative to the NIG.) The time dynamics of the asset prices are modeled by an exponential Lévy process. To price derivatives, even simple call and put options, we need to consider the numerical evaluation of the expectation. Raible (Raible 2000) has considered a Fourier method to evaluate call and put options. An alternative to this could be Monte Carlo method, however, these are rather slow in convergence.

The quasi-Monte Carlo (qMC) method has been applied with success in financial applications by many authors (see (Glasserman 2003), and references therein), and has very powerful convergence properties. Even though it samples deterministically, it is often considered as a

Date: 14 January 2007.

2000 *Mathematics Subject Classification.* 49M15, 65C05, 68Q25, 65D30.

Key words and phrases. Quasi-Monte Carlo, normal inverse Gaussian distribution, Newton-Raphson method, option pricing, implied volatility.

kind of Monte Carlo algorithm. Most of the work done on applying these simulation techniques to finance has concentrated on problems where one needs to simulate from the normal distribution. One exception is Kainhofer (Kainhofer 2003), who proposes a qMC algorithm for NIG variables based on a technique proposed by Hlawka and Mück (Hlawka and Mück 1972) to produce low-discrepancy samples for general distributions. His method requires knowledge of the cumulative NIG distribution function, which needs to be computed using numerical integration. We propose a qMC algorithm based on a simulation method for generalized inverse Gaussian distributions suggested by Michael, Schucany, and Haas (Michael, Schucany, and Haas 1976). The algorithm requires the simulation of three independent uniform random variables, and NIG samples are calculated via explicit transformations of these. For simplicity, the algorithm is given in one dimension, but extends readily to many dimensions. Our qMC-algorithm for NIG variates does not require the numerical inversion of the NIG cumulative distribution function.

We apply our algorithm on three financial problems, two one-dimensional option pricing problems and a multivariate portfolio problem. The first involves the pricing of a plain vanilla call option and an Asian call option, being a call option written on the average of the asset price over a specified time period. We can approximate the price of the latter as an expectation of a functional of a NIG distribution, which we evaluate based on our qMC algorithm. We compare our results with the algorithm proposed by Kainhofer (Kainhofer 2003). Our next application involves finding the "implied volatility" from a call and an Asian option based on a NIG model. More precisely, given the price of an Asian option, and supposing that the log-returns of the underlying asset is NIG distributed, how can we find one (or more) of the parameters of the NIG distribution? This is an inverse problem, where we try to find the parameter in the NIG distribution which is so that the quoted price is achieved. A natural approach is to use Newton's method, which involves calculating the option price along with its derivative. Thus, we need to calculate two expectations involving a multivariate NIG, and iterate these until convergence is reached. We provide a general analysis of the convergence properties of such an algorithm. Our final application is on Value-at-Risk (VaR). This is somewhat detached from option pricing, but still is an interesting application of our qMC-algorithm. We consider a portfolio of assets where the log-returns are modeled using NIG distributions (independently!), and compare with a crude Monte Carlo algorithm. Since the calculation of the VaR for a portfolio can be recast as finding a quantile, we may apply the Newton's method. However, it turns out that this is not a fruitful way compared to the usual approach with (quasi-) Monte Carlo and sorting.

The paper is organized as follows: In the next section we present the theory relating to pricing options with the NIG distribution. Following that we investigate a quasi-Monte Carlo algorithm for simulating NIG distributed random variables. Continuing, we go about finding implied parameters using Newton's Method and qMC. Next we turn attention to applications to finance. Finally, we summarize our conclusions.

2. Pricing options with the NIG distribution

Let $(\Omega, \mathcal{F}, P)$ be a probability space equipped with a filtration $\{\mathcal{F}_t\}_{t\in[0,T]}$ satisfying the *usual conditions*[1], with $T < \infty$ being the time horizon. Let $L(t)$ be a Lévy process being right-continuous with left-limit (RCLL, or càdlàg), and consider the following exponential

[1]see, *e.g.*, (Karatzas and Shreve 1991).

model for the asset price dynamics

$$S(t) = S(0)\exp\big(L(t)\big) \tag{2.1}$$

In this paper we will mostly be concerned with the exponential NIG-Lévy process dynamics, meaning that $L(t)$ has increments being distributed according to a NIG distribution.

The NIG family of distributions is specified by four parameters. A random variable is said to be NIG distributed with parameters μ, β, α and δ, denoted $X \sim \mathrm{NIG}(\alpha, \beta, \mu, \delta)$, where μ is the location, β the skewness, α the tail-heaviness and δ the scale. The density of a $\mathrm{NIG}(\alpha, \beta, \mu, \delta)$-variable is given by

$$p(x; \mu, \beta, \alpha, \delta) = \frac{\delta\alpha}{\pi}\exp\Big(\delta\sqrt{\alpha^2-\beta^2} + \beta(x-\mu)\Big)\,\frac{K_1\big(\alpha s(x-\mu)\big)}{s(x-\mu)} \tag{2.2}$$

where

$$x \in \mathbb{R}, \quad \mu \in \mathbb{R}, \quad \delta > 0, \quad 0 \le |\beta| \le \alpha$$

and

$$s(x) = \sqrt{\delta^2 + x^2},$$

and where K_1 is the modified Bessel function of the third kind with index 1. Specifically,

$$K_1(y) = \frac{y}{4}\int_0^\infty \exp\left[-\left(t + \frac{y^2}{4t}\right)\right] t^{-2}\,\mathrm{d}t, \quad y \in \mathbb{R}$$

The NIG family of distributions is infinitely divisible, which means that there exists a Lévy process such that for $\Delta t > 0$,

$$L(t+\Delta t) - L(t) \sim L(\Delta t) \sim \mathrm{NIG}(\alpha, \beta, \mu, \delta),$$

for every $t \ge 0$. It turns out that this Lévy process is a pure-jump process, and the associated Lévy measure is absolutely continuous with respect to the Lebesgue measure and its density can be calculated explicitly as

$$\ell(z) = \pi^{-1}\delta\alpha\,|z|^{-1}\,K_1\left(\alpha\,|z|\right)\mathrm{e}^{\beta z} \tag{2.3}$$

Note that $\int_{\mathbb{R}} \min(1, z^2)\ell(z)\,\mathrm{d}z < \infty$. We refer to (Barndorff-Nielsen 1998), (Geman 2002), and (Rydberg 1997), for a discussion of the NIG distribution and the corresponding Lévy process.

Considering an asset dynamics given by the exponential NIG-Lévy process, we can find the price of a European call option with strike price K at exercise time T as

$$C(0) = \mathrm{e}^{-rT}\,\mathrm{E}_Q\left[\max(S(T) - K, 0)\right], \tag{2.4}$$

where r is the risk-free interest rate and Q is an equivalent martingale measure. The exponential NIG-Lévy model gives rise to an incomplete market, thus leading to a continuum of equivalent martingale measures that can be used for risk-neutral pricing. However, we choose the approach of Raible (Raible 2000), and consider the Esscher transform method to derive a Q-measure for pricing. This approach is so-called structure preserving, in the sense that we search for equivalent probability measures Q such that the distribution of $L(T)$ remains in the class of NIG distributions and where the log-return of S is the risk-free return r. Thus, supposing $L(T) \sim \mathrm{NIG}(\alpha, \beta, \mu, \delta)$, we can recast the expression in Equation (2.4) as

$$C(0) = \mathrm{e}^{-rT}\,\mathrm{E}\left[\max\left(S(0)\mathrm{e}^X - K, 0\right)\right] \tag{2.5}$$

where X is a NIG$(\alpha, \hat{\beta}, \mu, \delta)$-variable with

$$\hat{\beta} = -\frac{1}{2} + \operatorname{sgn}(\beta)\sqrt{\frac{(\mu - r)^2}{\delta^2 + (\mu - r)^2}\alpha^2 - \frac{(\mu - r)^2}{4\delta^2}}.$$

In this paper we shall also be concerned with Asian options written on $S(t)$. Consider such an option with exercise at time T and strike price K on the average over the time span up to T. The risk-neutral price is

$$A(0) = \mathrm{e}^{-rT}\,\mathrm{E}_Q\left[\max\left(\frac{1}{T}\int_0^T S(t)\,\mathrm{d}t - K, 0\right)\right] \tag{2.6}$$

Again applying the Esscher transform, we have that $L(t)$ is still a NIG-Lévy process, and approximating the integral with a Riemann sum[2] yields the price

$$A(0) = \mathrm{e}^{-rT}\,\mathrm{E}\left[\max\left(\frac{S(0)}{N}\sum_{i=1}^{N}\exp\big(L(t_i)\big)\Delta t - K, 0\right)\right] \tag{2.7}$$

By using the independent increment property of the Lévy process we may rewrite the sum into a function of N increments of L, that is, into a function $g : \mathbb{R}^N \to \mathbb{R}$ such that

$$A(0) = \mathrm{e}^{-rT}\,\mathrm{E}\left[\max\left(g(X_1, \ldots, X_N) - K, 0\right)\right] \tag{2.8}$$

Here, $X_i = L(t_i) - L(t_{i-1})$, for $i = 1, \ldots, N$. For simplicity we focus on regular time partitions, $\Delta t = t_i - t_{i-1}$.

From the considerations above we see that both the call and the Asian pricing functional can be written as

$$C(0) = \mathrm{E}\left[f(X_1, \ldots, X_d)\right] \tag{2.9}$$

where $d \geq 1$ and X_i are *i.i.d.* NIG$(\alpha, \beta, \mu, \delta)$-variables. We note that numerous other type of options can be expressed in the same way, counting for instance spread options and barrier options. The number d gives the dimensionality of the problem, and the function f is connected to the payoff of the option and the exponential function giving the asset price dynamics. The rest of the paper is concerned with developing and analyzing a qMC method to valuate the expectation in Equation (2.9).

3. A quasi-Monte Carlo algorithm for simulating NIG distributed random variables

We develop a quasi-Monte Carlo method for simulating expectation of function of NIG distributed random variables. Include some discussion of convergence, and a numerical evaluation of the $\log N/N$ convergence.

Consider the simulation algorithm for sampling from a NIG$(\alpha, \beta, \mu, \delta)$-distributed variable X proposed by Rydberg (Rydberg 1997) building on work by Michael, Schucany and Haas (Michael, Schucany, and Haas 1976) (referred to from now on as the Rydberg-MC method):

- Sample Z from IG$(\delta^2, \alpha^2 - \beta^2)$

[2]Note that in practice there exists no Asian options with continuous averaging. The Asian options traded in the market has discrete averaging, also known as Bermudian options, and thus a simple Riemann approximation is the most natural.

- Sample Y from $\mathrm{N}(0,1)$
- Return $X = \mu + \beta Z + \sqrt{Z}Y$

The sampling of Z consists of first drawing a random variable V which is $^2(1)$-distributed, defining a random variable

$$W = \xi + \frac{\xi^2 V}{2\delta^2} - \frac{\xi}{2\delta^2}\sqrt{4\xi\delta^2 V + \xi^2 V^2}$$

and then letting

$$Z = W \cdot 1_{\left\{U_1 \leq \frac{\xi}{\xi+W}\right\}} + \frac{\xi^2}{W} \cdot 1_{\left\{U_1 > \frac{\xi}{\xi+W}\right\}},$$

U_1 being uniformly distributed, and $\xi = \delta/\sqrt{\alpha^2 - \beta^2}$. This provides us with a Monte Carlo algorithm for simulating an $\mathrm{NIG}(\alpha, \beta, \mu, \delta)$-distributed variable X.

From the algorithm, we see that to sample from X we basically need to sample a standard normal Y, a χ^2 distributed V, and a uniform U_1. The two first ones can be sampled from two independent uniform distributions U_2 and U_3 by a transformation using the normal distribution function; we are thus led to the conclusion that sampling from X entails sampling from three independent uniformly distributed random variables:

$$X = \mu + \beta q(U_2, U_3) + \sqrt{q(U_2, U_3)}\Phi^{-1}(U_1),$$

where Φ is the cumulative distribution function of the standard normal distribution, and

$$q(x,y) = w(x) \cdot 1_{\left\{y \leq \frac{\xi}{\xi+w(x)}\right\}} + \frac{\xi^2}{w(x)} \cdot 1_{\left\{y > \frac{\xi}{\xi+w(x)}\right\}},$$

with

$$w(x) = \xi + \frac{\xi^2\left(\Phi^{-1}(x)\right)^2}{2\delta^2} - \frac{\xi}{2\delta^2}\sqrt{4\xi\delta^2\left(\Phi^{-1}(x)\right)^2 + \xi^2\left(\Phi^{-1}(x)\right)^4}$$

These considerations give us a scheme to sample low-discrepancy sequences for the NIG distribution by combining three low-discrepancy sequences and appealing to the fast inversion algorithm for the normal distribution given by Moro (Moro 1995). We refer to this qMC algorithm for NIG as the Rydberg-qMC method.

We now discuss some issues on the convergence of this algorithm applied to calculating the prices of financial derivatives based on NIG models. First, in view of Equation (2.9) and the algorithm above, we can write

$$\begin{aligned} C(0) &= \mathrm{E}\left[f(X_1, \ldots, X_d)\right] \\ &= \mathrm{E}\left[h(U_1^1, U_2^1, U_3^1, U_1^2, U_2^2, U_3^3, \ldots, U_1^d, U_2^d, U_3^d)\right] \end{aligned}$$

for d independent triples of three independent uniform random variables (U_1^i, U_2^i, U_3^i), $i = 1, \ldots, d$. The function h is a combination of f and the transforms above. We can state this as an integration over the unit hypercube:

$$C(0) = \int_{[0,1)^{3d}} h(y_1, \ldots, y_{3d})\,\mathrm{d}y$$

Thus finding the price $C(0)$ using our proposed qMC algorithm entails in a $3 \times d$-dimensional problem. If we have a low-discrepancy sequence $\{y^k\}_{k=1,\ldots}$ in $[0,1)^{3d}$, the Koksma-Hlawka bound says that

$$\left| \int_{[0,1)^{3d}} h(y_1, \ldots, y_{3d})\, \mathrm{d}y - \frac{1}{N} \sum_{k=1}^{N} h(y_k) \right| \leq V(h) c(d) \frac{\log^{3d} N}{N}$$

where $V(h)$ is the variation of h in the sense of Hardy and Krause (see, *e.g.*, (Glasserman 2003)) and $c(d)$ is a constant only dependent on the dimension d. Note that this bound is only valid for functions h with finite variation, $V(h) < \infty$, which in general is not the case in financial applications since h may be unbounded. Also, the result predicts a rather slow convergence in higher dimensions. In practical examples the rate of convergence is, however, much better (see (Papageorgiou 2003) for a discussion of convergence related to financial applications).

We provide some numerical results indicating the convergence rate for our algorithm. A mathematical analysis of the properties of the algorithm will be provided elsewhere. In Fig. 1 we display some simulations of the convergence rate. We use a Niederreiter sequence to generate uniformly distributed low-discrepancy numbers and the Rydberg-qMC algorithm to get normal inverse Gaussian distributed numbers. We simulate an indicator function $1_{\{a,b\}}(x)$ and compare to a simulated true value. Fig. 1 show the relative error of the quasi-Monte Carlo simulation together with the smooth curve $c * \log^3 N/N$ with the constant $c = 0.2$. It is clear that for these simulations the convergence rate of the Rydberg-qMC numbers are of order $\log^3 N/N$ or better, and other simulations also confirms this.

Our proposed Rydberg-qMC algorithm is an alternative to the Hlawka-Mück method for qMC simulations from general distributions. The latter is used by Kainhofer (Kainhofer 2003) to generate qMC-samples from a NIG distribution. To sample a point set from a distribution with cumulative distribution function F we start with a uniformly distributed set $\omega = (x_1, x_2, \ldots, x_n)$ on the half open unit interval with discrepancy $D_n(\omega)$. We then let

$$y_k = \frac{1}{n} \sum_{r=1}^{n} 1_{\{0,x_k\}} F(x_r)$$

and get the F-distributed point set $\tilde{\omega} = (y_1, y_2, \ldots, y_n)$. Hence, every point in $\tilde{\omega}$ is of the form $i/n,\ i = 0, \ldots, n$ and we observe that we need to have, at least, a numerical approximation of the cumulative distribution function. If $M = \sup_{x \in [0,1)} f(x)$, where $f(x)$ is the corresponding density function, then the discrepancy of $\tilde{\omega}$ is bounded by

$$D_{n,F}(\tilde{\omega}) \leq (1 + M) D_n(\omega),$$

see (Kainhofer 2003). We shall refer to this algorithm as the HM-method and it extends readily to higher dimensions.

Since the Hlawka-Mück method only applies for distributions supported on the unit hypercube, Kainhofer (Kainhofer 2003) considers a transformation between the real line and unit interval given by the double-exponential distribution with parameter $\lambda > 0$, having cumulative distribution function

$$H_\lambda(x) = \begin{cases} \frac{1}{2}\exp(\lambda x) & \text{if } x < 0 \\ 1 - \frac{1}{2}\exp(-\lambda x) & \text{if } x \geq 0 \end{cases} \tag{3.1}$$

and inverse given as

$$H_\lambda^{-1}(y) = \begin{cases} \frac{1}{\lambda}\log(2y) & \text{if } y < \frac{1}{2} \\ -\frac{1}{\lambda}\log(2-2y) & \text{if } y \geq \frac{1}{2} \end{cases}$$

To prevent having an argument equal to zero in the logarithm, Kainhofer (Kainhofer 2003) suggests to shift zero by $1/n$, where n is the number of points in the sequence. This is shown to have minor influence on the properties of the sequence.

4. Finding implied parameters using Newton's Method and qMC

In finance one is often interested in the implied volatility, that is, the volatility of the asset price dynamics yielding a certain option price. If the option in question is of Asian type, one can not resort to the Black & Scholes formula to derive the implied volatility, but need to employ a numerical procedure involving calculation of the option price and search for the volatility for a given price. If the underlying asset is modeled using a exponential NIG-Lévy process, there are essentially three parameters to search for in a risk neutral pricing paradigm. We shall later concentrate on deriving the implied δ, and use the Newton Method in conjunction with our proposed qMC algorithm to find the implied δ from a given Asian option price.

We can state the problem in general as follows: Let $x \in \mathbb{R}$ be a parameter of the distribution for a random variable (being multi-dimensional in general) X. Define p to be

$$p = \mathrm{E}\left[f\big(X(x)\big)\right], \tag{4.1}$$

where we use the notation $X(x)$ to indicate that the distribution of X depends on x. Here, f is some function (in our context, the payoff from some option), and we assume that $f\big(X(x)\big)$ has finite variance. The problem is to find x for a given p, when the family of distributions for $X(x)$ is known but parametrized by x. For notational simplicity, define the function $g : \mathbb{R} \to \mathbb{R}$ to be

$$g(x) = \mathrm{E}\left[f\big(X(x)\big)\right] \tag{4.2}$$

It is natural to use the Newton Method to find x. However, this requires an evaluation of g along with its derivative $g'(x)$, and in our situation we do not have a functional expression even for $g(x)$ when X is NIG distributed. To evaluate $g(x)$ for a given x we will apply our Rydberg-qMC algorithm, but this introduces an error in the estimation. Even more, when estimating the derivative $g'(x)$ by numerical differentiation (and thereby a re-estimation of the function g at a slightly perturbed location) this error may become even bigger. We provide an error analysis of the methods in question, and show that by a careful increase in the length of the sampling sequence at each Newton step preserve the quadratic convergence property of the Newton algorithm.

Suppose $g \in C^2$, with $g'(x) \neq 0$ in U, and $|g''(x)| \leq K$ uniformly in U for some subset $U \subset \mathbb{R}$. Suppose further that there exists a low-discrepancy sequence for the distribution of $X(x)$ with convergence independent of $x \in U$, and given by the rate $\log^d N/N$ where N is

the length of the sequence and d the dimension. Recall that for the NIG distribution the dimension is $3 \times d$, with d being the dimension of X. Newton's Method takes the form

$$x_{i+1}^N = x_i^N - \frac{g^N(x_i^N) - p}{(g^N)'(x_i^N)} \tag{4.3}$$

after selecting an initial point x_0. In the process it makes a functional evaluation $g^N(x)$ by qMC, wherein the superscript N denotes the number of samples at step i. It will later be natural to index N by i, that is $N(i)$, to indicate that the number of samples in the qMC-sequence may depend on the step i in the Newton iteration. If we skip the index N, and write $g(x)$, we mean that the function g is evaluated accurately.

At each step the algorithm estimates $\big(g^N\big)'(x_i^N)$ by the secant method, using for the second point $g^N(x_i^N + \Delta_i)$, with the increment Δ_i chosen carefully to preserve accuracy in the next step.

We now move on the analyze the convergence properties of the method when the functional evaluations is made by qMC. The analysis addresses in particular the functional form of the requisite number of samples in the sequence, depending on the step index i.

4.1. **Convergence to a fixed-point.** With exact valuation of $g(x)$ and $g'(x)$, the i^{th} step takes x_i to x_{i+1} as follows.

$$x_{i+1} = x_i - \frac{g(x_i) - p}{g'(x_i)} \tag{4.4}$$

The second term is the exact error of the algorithm at step i, say ε_i. So,

$$\varepsilon_i := x_{i+1} - x_i = -\frac{g(x_i) - p}{g'(x_i)}$$

With qMC valuations, the approximate error of the algorithm would be, say ε_i^N. So,

$$\varepsilon_i^N := x_{i+1}^N - x_i = -\frac{g^N(x_i) - p}{\left[\dfrac{g^N(x_i + \Delta_i) - g^N(x_i)}{\Delta_i}\right]}$$

It is desired to keep the difference of these error terms small. To this end, see the difference, say ι_i, as

$$\iota_i := \varepsilon_i^N - \varepsilon_i = x_{i+1}^N - x_{i+1} = -\frac{g^N(x_i) - p}{\left[\dfrac{g^N(x_i + \Delta_i) - g^N(x_i)}{\Delta_i}\right]} - \varepsilon_i \tag{4.5}$$

We know, from the specification of qMC, that for some constant $c_i > 0$,

$$\big|g^N(x_i) - g(x_i)\big| \vee \big|g^N(x_i + \Delta_i) - g(x_i + \Delta_i)\big| \leq c_i \frac{\log^d N}{N} \xrightarrow[N\to\infty]{} 0 \tag{4.6}$$

where d is the dimension of the valuation domain. This fact, along with the continuity of $g'(x)$, guarantees from Equation (4.5) that

$$\lim_{\substack{\Delta_i \to 0 \\ N \to \infty}} \iota_i = 0, \tag{4.7}$$

and thus for sufficiently small Δ_i and large N, the introduction of qMC valuations compromises neither the existence of the successive approximations $\{x_i\}$ of Newton's Method, nor their accuracy. A consequence is that the algorithm produces a virtual fixed point at a solution.

4.2. **Rate of convergence.** We approach convergence of the Newton-qMC algorithm in three parts, determining

(1) the choice of Δ_i to ensure that for sufficiently large N, ι_i is small
(2) the choice of N, with corresponding estimate for ι_i
(3) an implicit function $N(i)$ expressing the number of samples through the steps

4.2.1. *Choice of* Δ_i. This Subsubsection presents a basic error analysis for using the secant method to approximate a derivative, in the context of a Newton's method step, and using the introduced notation. Similar analyses appear in many places under the heading "numeric differentiation." A good source is (Griewank 2003), which contains an extensive bibliography encompassing the relevant issues.

Looking to Equation (4.7) we wish to select an appropriate value of Δ_i so that step i of the algorithm can provide a sufficiently accurate value x_{i+1}^N. Herein we take "sufficiently accurate" to mean that any inaccuracy in estimating $g'(x_i)$ by substituting the exact secant slope adds no more error to x_{i+1}^N than the estimated error of the algorithm at the following step, $\tilde{\varepsilon}_{i+1}$, a value developed below as Equation (4.13). This error is estimable from the quadratic convergence of Newton's Method, wherein ε_{i+1} is $\mathrm{O}(\varepsilon_i^2)$. Specifically,

$$\varepsilon_{i+1} \approx -\frac{g''(x_{i-1})}{2g'(x_{i-1})}\varepsilon_i^2 \tag{4.8}$$

We make these concepts more precise, and end the narrative with the principal result, Equation (4.14) below. Refer first to Equation (4.4). Continuing with exact analysis, that is, without yet the invocation of qMC valuations, let us estimate the effect of using a secant approximation to $g'(x_i)$. Allow this estimate to be

$$\bar{g}'(x_i) := \frac{g(x_i + \Delta_i) - g(x_i)}{\Delta_i}, \tag{4.9}$$

and then let

$$\hat{x}_{i+1} := x_i - \frac{g(x_i) - p}{\bar{g}'(x_i)},$$

and further let

$$\hat{\varepsilon}_i := \hat{x}_{i+1} - x_i$$

By Taylor's expansion

$$\bar{g}'(x_i) = g'(x_i) + \frac{1}{2}g''(x_i)\Delta_i,$$

ignoring third and higher order terms. So,

$$\hat{\varepsilon}_i = -\frac{g(x_i) - p}{g'(x_i) + \frac{1}{2}g''(x_i)\Delta_i}$$

The effect of the secant approximation $\bar{g}'(x_i)$, therefore, is to induce a second order error to x_{i+1}^N of magnitude

$$\kappa_i := \hat{\varepsilon}_i - \varepsilon_i = -\frac{g(x_i) - p}{g'(x_i) + \frac{1}{2}g''(x_i)\Delta_i} - \varepsilon_i,$$

and so

$$\kappa_i = \frac{\frac{1}{2}g''(x_i)\Delta_i\varepsilon_i}{g'(x_i) + \frac{1}{2}g''(x_i)\Delta_i}$$

But $|g''(x_i)| \leq K$, and therefore one may first choose

$$|\Delta_i| \leq \frac{g'(x_i)}{K} \tag{4.10}$$

to ensure that

$$|\kappa_i| \leq K \left| \frac{\Delta_i\varepsilon_i}{g'(x_i)} \right| \tag{4.11}$$

One may further choose Δ_i to meet any desired maximal value for $|\kappa_i|$.

To this end, return to the estimated error of the algorithm at the following step, $\tilde{\varepsilon}_{i+1}$. In the iteration of Newton's Method at the i^{th} step we have in hand the error terms $\hat{\varepsilon}_{i-2}$ and $\hat{\varepsilon}_{i-1}$. These are related, at least approximately insofar as qMC valuations are incorporated, by Equation (4.8), adjusted back two iterations. Thus we may infer

$$\hat{\varepsilon}_{i-1} \approx -\frac{g''(x_{i-3})}{2g'(x_{i-3})}\hat{\varepsilon}_{i-2}^2$$

The coefficient herein, we assume is bounded on the domain of convergence through the iterations, and thus

$$\nu := \sup_{i\geq 2} \frac{|\hat{\varepsilon}_{i-1}|}{\hat{\varepsilon}_{i-2}^2} \text{ exists.}$$

It follows readily that

$$\tilde{\varepsilon}_i \leq \nu\hat{\varepsilon}_{i-1}^2, \tag{4.12}$$

and that

$$\tilde{\varepsilon}_{i+1} \leq \nu^3\hat{\varepsilon}_{i-1}^4 \tag{4.13}$$

This last estimate is the one we merge with Equation (4.11) to provide a choice of Δ_i. Remembering the first constraint on Δ_i, as expressed in Equation (4.10), we have

$$|\Delta_i| \leq \frac{|g'(x_i)|}{K} \wedge \frac{|g'(x_i)|}{K}\nu^3\frac{\hat{\varepsilon}_{i-1}^4}{|\varepsilon_i|} = \frac{|g'(x_i)|}{K}\left(1 \wedge \nu^3\frac{\hat{\varepsilon}_{i-1}^4}{|\varepsilon_i|}\right) \tag{4.14}$$

In practice neither ε_i nor $g'(x)$ is known in advance, so we substitute in the former instance the value $\tilde{\varepsilon}_i$ from Equation (4.12), and in the latter instance the value $\bar{g}'(x_i)$ from Equation (4.9).

4.2.2. *Number of samples for a step.* Again looking to Equation (4.7), we wish to select an appropriate value of N so that step i of the algorithm can provide a sufficiently accurate value x_{i+1}^N, ending the narrative with the principal result, Equation (4.18) below. We take "sufficiently accurate" to mean that any inaccuracy in estimating $g'(x_i)$ by approximating $g(x_i)$ and $g(x_i + \Delta_i)$ by $g^N(x_i)$ and $g^N(x_i + \Delta_i)$, respectively, further adds no more error to x_{i+1}^N than the estimated error of the algorithm at the following step, $\tilde{\varepsilon}_{i+1}$.

With a choice of Δ_i made, we look to the outer error bound for qMC, as expressed in Equation (4.6), as a guide in selecting sample size. To proceed it is first necessary to estimate empirically the coefficient c_i, for there are some variables which are intractable analytically, such as the effect of a particular choice of sampling scheme. It may well be also that c_i is not sensitive to the step of the iteration, and so may be chosen uniformly.

Refer to Equation (4.5) and Equation (4.6). It is desired to select the number of samples N such that

$$|\iota_i| \leq |\tilde{\varepsilon}_{i+1}|$$

To this end, assume that

$$\begin{aligned} g^N(x_i) &= g(x_i) + \zeta_i, \\ g^N(x_i + \Delta_i) &= g(x_i + \Delta_i) + \eta_i, \end{aligned}$$

and

$$|\zeta_i| + |\eta_i| \leq c_i \frac{\log^d N}{N} \tag{4.15}$$

Then, after some elementary manipulation,

$$\iota_i = -\frac{\big(g(x_i) - p\big) + \zeta_i}{\left[\dfrac{g(x_i + \Delta_i) - g(x_i)}{\Delta_i} + \dfrac{\eta_i - \zeta_i}{\Delta_i}\right]} - \varepsilon_i$$

This calculation considers the combined effects of estimating $g'(x_i)$, and of using qMC to value $g(x_i)$ and $g(x_i + \Delta_i)$. Insofar as error in $g'(x_i)$ has already been accounted, replace the term

$$\frac{g(x_i + \Delta_i) - g(x_i)}{\Delta_i} \text{ above, with } g'(x_i),$$

to focus on the error induced by qMC valuations. Thus, we wish to set

$$\left| -\frac{\big(g(x_i) - p\big) + \zeta_i}{\left[g'(x_i) + \dfrac{\eta_i - \zeta_i}{\Delta_i}\right]} - \varepsilon_i \right| = \left| \frac{\zeta_i + \dfrac{\eta_i - \zeta_i}{\Delta_i}\varepsilon_i}{g'(x_i) + \dfrac{\eta_i - \zeta_i}{\Delta_i}} \right| \leq |\tilde{\varepsilon}_{i+1}| \tag{4.16}$$

Assume by Equation (4.15) that we have chosen N sufficiently large that

$$|\zeta_i| + |\eta_i| \leq \frac{|\Delta_i g'(x_i)|}{2},$$

and thus that

$$\left|\frac{\eta_i - \zeta_i}{\Delta_i}\right| \leq \frac{|\eta_i| + |\zeta_i|}{|\Delta_i|} \leq \frac{|g'(x_i)|}{2}$$

Replace the first factor of the denominator in Equation (4.16) by $\frac{1}{2}|g'(x_i)|$, which is smaller, giving

$$\frac{\left|\zeta_i + \frac{\eta_i - \zeta_i}{\Delta_i}\varepsilon_i\right|}{\frac{1}{2}|g'(x_i)|} \leq |\tilde{\varepsilon}_{i+1}|$$

Enlarging the numerator gives

$$\frac{|\zeta_i| + \frac{|\eta_i| + |\zeta_i|}{|\Delta_i|}|\varepsilon_i|}{\frac{1}{2}|g'(x_i)|} = \frac{\left(1 + \left|\frac{\varepsilon_i}{\Delta_i}\right|\right)|\zeta_i| + \left|\frac{\varepsilon_i}{\Delta_i}\right||\eta_i|}{\frac{1}{2}|g'(x_i)|} \leq |\tilde{\varepsilon}_{i+1}| \tag{4.17}$$

further as sufficient. This last expression can be driven to zero with large N.

The formulation to calculate a sufficient N is evident. If Equation (4.15) holds, then also

$$|\zeta_i| \leq c_i \frac{\log^d N}{N}$$

and

$$|\eta_i| \leq c_i \frac{\log^d N}{N}$$

independently. These relations combined with Equation (4.17) evolve to

$$\frac{\left(1 + 2\left|\frac{\varepsilon_i}{\Delta_i}\right|\right)}{\frac{1}{2}|g'(x_i)|} c_i \frac{\log^d N}{N} \leq |\tilde{\varepsilon}_{i+1}| \tag{4.18}$$

as a sufficient condition on N. One may solve this relation numerically to guarantee the qMC induced error small, that is, within the bound of $\tilde{\varepsilon}_{i+1}$, as expressed. In practice neither ε_i nor $g'(x)$ is known in advance, so we substitute in the former instance the value $\tilde{\varepsilon}_i$ from Equation (4.12), and in the latter instance the value $\bar{g}'(x_i)$ from Equation (4.9).

Under some circumstances convergence of qMC may be faster than that indicated herein. For a discussion see (Papageorgiou 2003).

4.2.3. *Step dependent qMC sampling.* N, recall, is the number of samples taken for qMC valuation of $g(x)$ and $g'(x)$ at Newton step i. We shall indicate this dependence as $N(i)$. The principal result herein is Proposition 4.1.

From Equation (4.12) we have implied, given the assumed stability of ν and the faithful prediction of $\hat{\varepsilon}_i$ by $\tilde{\varepsilon}_i$, that

$$|\tilde{\varepsilon}_{i+1}| = \nu\tilde{\varepsilon}_i^2$$

Combined therefore with Equation (4.6) we have

$$|\tilde{\varepsilon}_{i+1}| = \nu\left[c_i \frac{\log^d N(i)}{N(i)}\right]^2, \tag{4.19}$$

but by the same reasoning,

$$|\tilde{\varepsilon}_{i+1}| = c_{i+1} \frac{\log^d N(i+1)}{N(i+1)} \tag{4.20}$$

One may assume that the series $\{c_i\}$ is stable through the Newton steps, especially as the steps get smaller as a solution is approached. Assume, therefore $c := c_i, \quad i \geq 0$, as

this approximate value. Equations (4.19) and (4.20) therefore imply a relationship between $N(i+1)$ and $N(i)$. This is given implicitly by

$$\frac{\log^d N(i+1)}{N(i+1)} = \nu c \left[\frac{\log^d N(i)}{N(i)}\right]^2 \tag{4.21}$$

Table 1 below shows an example of the evolving number of samples necessary to maintain accuracy, computed recursively from Equation (4.21) above, for the captioned parameters.

Iteration i	Samples $N(i)$	$\log N(i)$
0	1,000	6.908
1	2,035	7.618
2	7,534	8.927
3	80,926	11.301
4	5,969,401	15.602
5	16,024,385,755	23.497

TABLE 1. qMC Samples by Iteration: $N(0) = 1000, \nu = 1, c = 2, d = 3$

Next, we state formally this observed growth of $N(i)$.

Proposition 4.1 (Log Samples Limit)**.** *If*

$$\gamma := \nu c \frac{\log^d N(0)}{N(0)} < 1,$$

then

$$\liminf_{i\to\infty} \frac{\log N(i)}{2^i \log \gamma^{-1}} \geq 1$$

Proof. Take Equation (4.21) above, and compute the Newton error by recursion to step i beginning at step 0. Resulting is this relationship.

$$\frac{\log^d N(i)}{N(i)} = (\nu c)^{2^i-1} \left[\frac{\log^d N(0)}{N(0)}\right]^{2^i} = (\nu c)^{2^i-1} \left[\frac{\gamma}{\nu c}\right]^{2^i} = \frac{\gamma^{2^i}}{\nu c}$$

Assuming $N(i) > 1$,

$$\frac{\log N(i)}{2^i \log \gamma^{-1}} = \frac{d \log\log N(i)}{2^i \log \gamma^{-1}} + \frac{\log(\nu c)}{2^i \log \gamma^{-1}} + 1$$

As $i \to \infty$ the denominators of these terms increase without bound, because $\log \gamma^{-1} > 0$ by the hypothesis. Therefore, the second term on the right converges to zero. If the first term on the right also converges to zero, then the assertion follows to a limit of one. Otherwise the limit inferior is greater. □

5. Applications to finance

In this Section we consider three applications of our qMC method for simulating the normal inverse Gaussian distributed variables. The first example contains the valuation of a plain vanilla call option and different Asian options when the underlying asset price dynamics is driven by a geometric NIG-Lévy process. Next we consider the problem of recovering parameters of the underlying asset price dynamics when observing option prices. This is a problem similar to finding the implied volatility in the Black & Scholes context, however, in our situation we need to resort to simulation since there is no analytical option pricing formula. We find the implied δ in the NIG distribution from the observed plain vanilla call and Asian option prices, and our method combines qMC-valuation of these option prices with Newton's method to iterate toward the implied value. In our final application we analyze the qMC method to deriving the Value-at-Risk measure for a portfolio of assets. We compare with standard Monte Carlo, but also demonstrate how we can use Newton's method to simulate VaR, even though not much is gained with this approach. In our applications we focus on both accuracy and efficiency in terms of speed.

5.1. **Calculating option prices with qMC for NIG.** We consider the problem of pricing options written on an asset dynamics given by an exponential NIG-Lévy process. We suppose that parameters of the NIG distributed log-returns under the equivalent martingale measure given by the Esscher transform of the asset is given by

$$\mu = 0.00395, \beta = -15.1977, \alpha = 136.29, \delta = 0.0295$$

which are the same set of parameters as in (Kainhofer 2003, Ch. 8). We note in passing that these parameters are relevant for daily observed stock price log-returns (see, *e.g.*, (Rydberg 1997) for empirical analysis of Danish stock returns). We suppose further that the stock price today is $S(0) = 100$ and that the risk-free interest rate is $r = 3.75\%$ yearly.

Consider first European at-the-money call options with a common strike $K = 100$ and exercise horizons of four, eight, or twelve weeks, calculated by weekly sampling with NIG parameters as above. We now compare our proposed Rydberg-qMC method with the HM-qMC method. To show the superiority qMC-methods over crude Monte Carlo, we also include a comparison with the Rydberg-MC method and an acceptance-rejection Monte Carlo method (AR-MC). For the HM-qMC method, we apply $\lambda = 95.2271$ as in (Kainhofer 2003) in the double-exponential transformation of Equation (3.1). For both the Monte Carlo algorithms we use the built-in functions in Matlab for simulating uniform and normal variables. A Halton sequence is used for the HM-qMC method, while for Rydberg-qMC we base our sampling with a three dimensional Niederreiter sequence. We compare the four approaches in terms of their relative error, where the "correct" price is obtained from a long Monte Carlo simulation. In Fig. 2 we have plotted the relative error as a function of the number of samples in the sequence, with log-scale on both axes. The error for the Hlawka-Mück method is generally lower than for the other methods but the quasi-Monte Carlo method is superior to the two Monte Carlo methods. The results for the two Monte Carlo methods are the mean over ten consecutive runs and we observe that the two methods perform equivalently for all sets of points. Our quasi-Monte Carlo method is slightly worse off than the Hlawka-Mück method in accuracy, which is expected. In one dimension the Hlawka-Mück points are filling the space in a more even way than the qMC points. Hence, we expect that the Hlawka-Mück method is more accurate for a given point set, but are confident that the quasi-Monte Carlo method performs better than ordinary Monte Carlo.

Points	H-M	qMC	Rydberg-MC	A-R
32	0.1400	0.0500	0.0000	0.0084
64	0.1600	0.0100	0.0000	0.0164
128	0.3200	0.0300	0.0000	0.0344
256	0.6900	0.0400	0.0100	0.0672
512	1.6200	0.1000	0.0100	0.1372
1024	4.3500	0.1800	0.0300	0.2704
2048	16.6100	0.3800	0.0600	0.5404
4096	91.1800	0.7800	0.1200	1.0812
8192	912.4300	1.6900	0.3300	2.1676
16386	8861.8200	4.1100	0.9000	4.3488

TABLE 2. Table of the execution times in seconds for the vanilla call option price with different sizes of the sequence

In accuracy the Hlawka-Mück method seems to outperform the other methods. However, the execution times differ significantly, see Table 2. The generation of the Hlawka-Mück numbers involves calculating the cumulative distribution function using numerical integration and iterates over all points repeatedly, which makes the method very slow compared to the other ones. Even though the integration is in only one dimension, and thus avoids the curse of dimensionality in multi-dimensional integration, the execution time for the Hlawka-Mück method compared to the other ones is a clear indication that there is more work done than necessary. The quasi-Monte Carlo method we propose has slightly lower accuracy than the Hlawka-Mück method, but when considering the time it takes to reach a certain level of accuracy our method is clearly competitive. The Rydberg-MC method is the fastest method for a given point set but it suffers, along with the Acceptance-Rejection method, from lower accuracy.

We also consider the same Asian option pricing problem that Kainhofer (Kainhofer 2003) examines. The option is sampled in weekly intervals and the parameters for the distribution are taken from Kainhofer. We let the options, as noted, have maturities of four, eight, or twelve weeks, and use a Sobol sequence for all quasi-Monte Carlo methods. We see from Figure 3 that our method is not as accurate as the Hlawka-Mück method in general, but still better than the crude Monte Carlo. In 12 dimensions we observe that we do not get nearly as good results for the Hlawka-Mück method as in (Kainhofer 2003). This could perhaps be attributed to a better numerical integration in Kainhofer, who uses Mathematica to do the integration before applying the Hlawka-Mück method. We do the integration within the method, using native Matlab routines. Also, even if the Hlawka-Mück method gives a better result over a given number of points, we may reach the same accuracy in shorter time with our method, using more points.

5.2. **Finding implied parameters from option prices.** We next turn our attention to the problem of finding parameters implied from given prices. We could imagine that we have option prices quoted on some market and a model for the dynamics of the underlying assets. For example, we could imagine that the underlying asset follows some stochastic model making the log-returns normal inverse Gaussian distributed. Since the methods can only work with one single parameter we must for a four-parameter distribution such as the normal inverse

δ	$\hat{\delta}$	Time	δ	$\hat{\delta}$	Time
0.001	0.00098	0.30912	0.011	0.01090	0.17700
0.002	0.00192	0.69724	0.012	0.01177	0.19173
0.003	0.00287	0.22056	0.013	0.01269	0.21593
0.004	0.00390	0.34375	0.014	0.01357	0.18810
0.005	0.00494	0.20619	0.015	0.01460	0.17739
0.006	0.00599	0.17977	0.016	0.01563	0.17744
0.007	0.00698	0.18817	0.017	0.01664	0.17814
0.008	0.00799	0.18907	0.018	0.01768	0.19812
0.009	0.00898	0.18820	0.019	0.01869	0.18875
0.010	0.00993	0.18844	0.020	0.01968	0.17994

TABLE 3. Simulated δ with a combination of Newton's method and quasi-Monte Carlo simulations. Columns three and six give the time until the methods terminate. The relatively long time for $\delta = 0.002$ is due to the maximal number of iterations being exceeded before any other terminal condition was met.

Gaussian have some other way to assess the other three parameters beforehand. When we have the other parameters in place it is an easy task for the algorithm to find the remaining parameter sought from the given price.

In any such computational process, a good stopping rule is essential. Among such rules are these.

(1) Perform a predetermined number of iterations, based on an analysis of errors,
(2) Iterate until successive absolute differences fall below some threshold, and
(3) Iterate until successive absolute differences fail to get smaller, choosing the next to last value as best.

The third of these is a good choice for Newton's Method if one desires full machine accuracy across computing platforms in a production environment.

We start testing the method with a European call option. The method is implemented in C++ and we use a very long Monte Carlo simulation to get a "true value" for the option, which will be the designated target. We use parameter values from (Rydberg 1997) for the NIG distribution and choose the set of parameters for Deutsche Bank as our test example. The estimated value of the scale parameter is in this case $\delta = 0.012$, but to test the model we try a range of values such that $\delta \in [0.001, 0.002, \ldots, 0.020]$. We implement a few different termination criteria for Newton's method, settling on a combination of the first and second rules as listed above. We iterate to a selected small difference of successive values, but only until a chosen maximum number of them. We found that using 4096 points for the quasi-Monte Carlo method gave a good balance between speed and accuracy in the simulations, and proved sufficient for our research needs. We see from Table 3 that the method finds the given value of δ within a few percent relative error in about one fifth of a second when the method terminates before reaching the maximal number of iterations allowed. It should be noted that the method compares option prices with a precision of order 10^{-5}, which is much more precise than what is quoted as market prices. Also, the Monte Carlo simulation of the "true price"

δ	$\hat{\delta}$	Time	δ	$\hat{\delta}$	Time
0.001	0.00098	1.08032	0.011	0.01089	0.74758
0.002	0.00197	1.08491	0.012	0.01194	0.74309
0.003	0.00300	1.08687	0.013	0.01292	0.74683
0.004	0.00397	0.97535	0.014	0.01379	0.75018
0.005	0.00492	0.97139	0.015	0.01475	0.74524
0.006	0.00592	0.86727	0.016	0.01564	0.85670
0.007	0.00691	0.86690	0.017	0.01668	0.85645
0.008	0.00787	0.85736	0.018	0.01770	0.85376
0.009	0.00890	0.86405	0.019	0.01889	0.86557
0.010	0.00980	0.63186	0.020	0.01984	0.87636

TABLE 4. Simulated δ with a combination of Newton's method and quasi-Monte Carlo simulations for an Asian option over ten days. $\hat{\delta}$ is the estimate when we assume we have quoted price with two decimals.

adds some additional errors which we do not have if we consider the quoted price as the true observed price.

This method easily extends to path dependent options such as Asian options. To illustrate this we test the method using a 10 days Asian option with daily normal inverse Gaussian distributed log-returns and parameters as above. We apply a Sobol sequence for the low-discrepancy numbers and the effective dimension is 30. We now consider a case where we have quoted option prices with only two decimals precision. In reality this is the situation we would find if we used real data as the basis for our root finding algorithm. We lower the required accuracy in the Newton's method to account for this. The method now requires longer time, since we need much more work to evaluate the option in each qMC step. As we can see in Table 4 the time for the simulation is now around a second. The precision in the estimates are overall not significantly worse than previous results. Clearly, to have prices quoted with many decimals is not crucial for the result. The error in the Monte Carlo or quasi-Monte Carlo evaluations is probably more influential than the error in the terminal condition of the Newton's method.

Following the convergence analysis in Section 4.2.3 we tried an approach where we increased the number of qMC points in the function evaluation for every step of the Newton's method. This would ensure that the function evaluation is of the same order as the expected error from the Newton's iteration. However, we found that this did not improve the convergence, rather the opposite. We believe that this is a practical problem, because the number of points and iterations is comparably small. The change in the function evaluation we experience by changing the number of points distracts the Newton's method, requiring more iterations to get the same accuracy. However, if we let the number of points and iteration approach infinity the convergence analysis show that we converge to the correct answer, while with a fixed number of points the method will converge to an estimate with an error bounded by the qMC error.

5.3. **Calculating the Value-at-Risk for a portfolio.** Let X be a random variable describing the portfolio position at time T. We are interested in finding the Value-at-Risk $VaR_T(p)$

for a given risk level $p \in (0,1)$ at time T, defined as:

$$\Pr\left[X \leq VaR_T(p)\right] = p \tag{5.1}$$

We can rewrite this as

$$\mathrm{E}\left[1_{\{X \leq VaR_T(p)\}}\right] = p$$

To this end, define the function $g : \mathbb{R}_+ \to [0,1]$ as

$$g(x) = \mathrm{E}\left[1_{\{X \leq x\}}\right] \tag{5.2}$$

and note that $VaR_T(p)$ is a solution of the equation $g(x) = p$. We can find this solution by using a fixed-point iteration in conjunction with some simulation method enabling us to calculate $g(x)$ for a given x. We suggest using quasi-Monte Carlo techniques for the latter. Letting $x_0 \in \mathbb{R}_+$ be our initial guess of $VaR_T(p)$, we can use Newton's Method to iterate as follows:

$$x_{n+1} = x_n - \frac{g(x_n) - p}{g'(x_n)} \tag{5.3}$$

We now elaborate a bit on the form of g. We let X be the value of a portfolio of n risky assets or a mixture of assets and options on these, represented as

$$X = \sum_{i=1}^{n} f_i\big(S_1(T), \ldots, S_m(T)\big) \tag{5.4}$$

Here $S_j(t), j = 1, \ldots, m$, are m independent geometric NIG Lévy processes and f_j are the pay-off functions. If asset number j is a stock, then $f_j(x_1, \ldots, x_m) = x_j$, while if it is a call option we can write it as $f_j(x_1, \ldots, x_m) = [x_j - K]^+$. However, the specification of the f_j's can be chosen rather freely. We conclude with

$$g(x) = \mathrm{E}\left[1_{\left\{0 \leq \sum_{i=1}^{n} f_i\left(S_1(T), \ldots, S_m(T)\right) \leq x\right\}}\right]$$

We then turn the attention to the simulation of Value-at-Risk with the combined qMC and Newton's method approach. We use an ordinary Newton-Raphson method and a Sobol sequence (Sobol 1967; Sobol 1969) to generate the uniform quasi-random numbers. For the numerical derivative we keep track of the closest point larger than the current estimate. We then use the difference between the the function values at the two points divided by the distance between them.

Our test case is a portfolio consisting of 10 options. We use normal inverse Gaussian log-returns employing the proposed quasi-Monte Carlo (Rydberg-qMC) sampling algorithm, and let the options have different parameters to reflect the different heavinesses of the tails. Observe that we estimate the quantile rather then the possible loss. For a true value we use a Monte Carlo simulation over $100,000$ points.

One concern we must address is the problem with the number of points. Using a Sobol sequence to generate quasi-random numbers we would preferably use 2^k points, where k is a positive integer. However, as we are interested, for example, in Value-at-Risk at 5%, using $2^{10} = 1024$ points gives a subsample of $0.05 \cdot 1024 = 51.2$ points, which is not an integer. We could use a number of points such that the subsample is an integer, but the risk is that this practice would demolish the advantage of the quasi-random numbers.

Taking n points and letting the desired level of Value-at-Risk be VaR, the way our method works we can not hope for a better value for VaR than that between the VaR-point and the point above in the sorted point set; see Table 5 and Figure 4. We see that we do better than

VaR	Lognormal	True	MC	qMC Sorting	qMC Newton
0.010	9.7652	4.2166	4.9494	3.8483	$= 3.8483$
0.020	10.4651	5.1591	6.0128	3.9128	< 4.0000
0.014	11.1104	6.0225	7.9668	4.5565	$= 4.5565$
0.016	11.7137	6.9381	8.2702	4.8286	$= 4.8286$
0.018	12.2833	7.8956	10.0260	5.4323	< 5.6703
0.020	12.8253	8.7591	12.2965	6.3773	$= 6.3773$
0.022	13.3442	9.6287	12.9095	7.0618	< 7.8710
0.024	13.8433	10.4094	14.3063	8.4053	< 8.7925
0.026	14.3254	11.2482	14.6840	9.1187	< 9.4216
0.028	14.7927	11.9398	15.6789	10.6132	< 10.8920
0.030	15.2469	12.8199	18.0643	11.6665	$= 11.6665$
0.032	15.6895	13.5697	18.3838	12.5485	< 13.3410
0.034	16.1217	14.3761	19.0249	14.1657	$= 14.1657$
0.036	16.5445	15.1415	19.3859	14.4826	< 14.6097
0.038	16.9588	15.9424	20.3904	14.9184	< 15.4281
0.040	17.3655	16.6828	22.1250	15.5063	$= 15.5063$
0.042	17.7651	17.5082	22.8483	16.0907	$= 16.0907$
0.044	18.1583	18.3226	23.4318	16.3529	$= 16.3529$
0.046	18.5455	19.0592	23.6134	16.5205	< 17.2845
0.048	18.9273	19.7754	24.1981	17.8377	< 17.8994

TABLE 5. Results for Value-at-Risk, in order: Log Normal, True over 100,000 points, Monte Carlo, quasi-Monte Carlo and sorting, quasi-Monte Carlo and Newton's Method

the Monte Carlo method, but we are not very close to the true solution. Our hope is that our method is faster than sorting the points to find this point. We run 10 consecutive runs and take the mean value over these times to try to smooth computer dependent variations. As we see in Table 6, our method is comparable with the approach to sort the points. But, if we look more closely into what takes time in the algorithms, we can see that drawing the quasi-random numbers and calculating the portfolio spends more than 0.20 seconds. This can be compared with the time of sorting 1000 points, which takes about 0.004 second. Hence, the time to gain with our approach is insignificant compared with the time it takes to draw the random numbers.

It is clear that our method gives no advantage over the sorting approach. It appears that the methods proposed do not show any improvement when calculating the Value-at-Risk.

6. Conclusions

We have proposed a qMC-algorithm to draw NIG-variates. The algorithm is applied to three problems appearing in finance, namely valuation of options, finding implied parameters from quoted option prices and deriving the Value-at-Risk for a nonlinear portfolio. Our algorithm is compared with several other ways to compute prices numerically, and it is demonstrated that it works efficiently and accurately. When finding implied parameters, we combine the

VaR	MC	qMC Sorting	qMC Newton
0.010	0.06258900	0.2563974	0.2412919
0.012	0.07825003	0.2359742	0.2495200
0.014	0.08729312	0.2661687	0.2155142
0.016	0.09785151	0.2145515	0.2720427
0.018	0.07999894	0.2354247	0.2664269
0.020	0.06265327	0.2047915	0.2147248
0.022	0.07821451	0.2614528	0.2394932
0.024	0.09920062	0.2495174	0.2204215
0.026	0.07245710	0.2384592	0.2364952
0.028	0.06869601	0.2336857	0.2383182
0.030	0.06599299	0.2326620	0.2714227
0.032	0.07176859	0.2544114	0.2112069
0.034	0.07416334	0.2097142	0.2486333
0.036	0.07306887	0.2751248	0.2388638
0.038	0.06022343	0.2146338	0.2116795
0.040	0.08779631	0.2060886	0.2044529
0.042	0.06652290	0.2058392	0.2299584
0.044	0.07902179	0.2013111	0.2250775
0.046	0.07487483	0.2545944	0.2380706
0.048	0.08748519	0.1969024	0.2340689

TABLE 6. Times to calculate Value-at-Risk for in order, Monte Carlo, quasi-Monte Carlo and sorting, quasi-Monte Carlo and Newton's Method

qMC algorithm with a Newton Method, for which we also provide an analysis of convergence properties.

Our qMC-algorithm is based on a Monte Carlo simulation algorithm suggested by Rydberg (Rydberg 1997). It is an alternative to the general Hlawka-Mück method for sampling non-uniform distributions, and we argue for its superiority in the sense of computational speed and simplicity. Our proposed sampling technique involves simulating three uniform variables based on low-discrepancy sequences, instead of doing a numerical integration to achieve the cumulative distribution function which is the case for the Hlawka-Mück method.

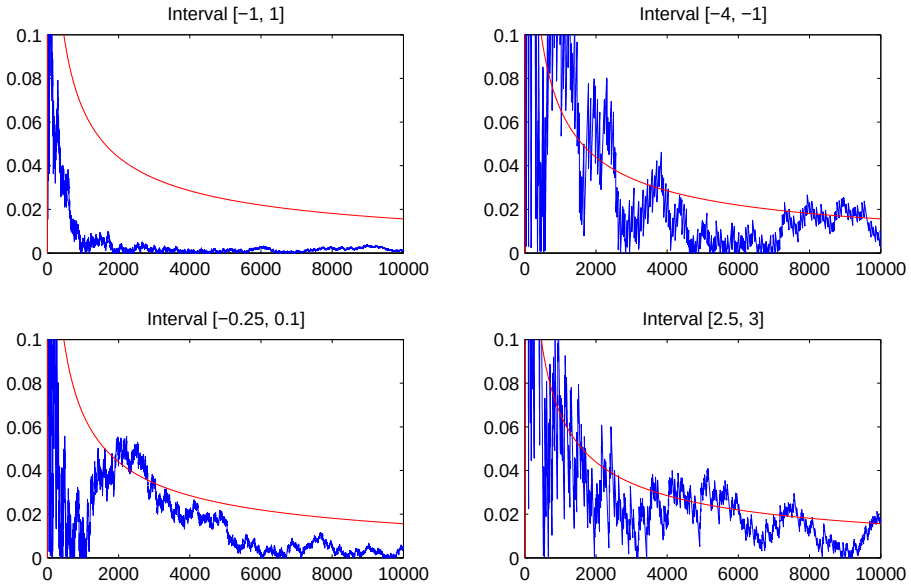

FIGURE 1. Convergence of the Rydberg-qMC algorithm for the estimate of an indicator function over an interval with distribution NIG(1, 0.75, 1, 0). The smooth curves show the function $c * \log^3 N/N$ with the constant $c = 0.2$.

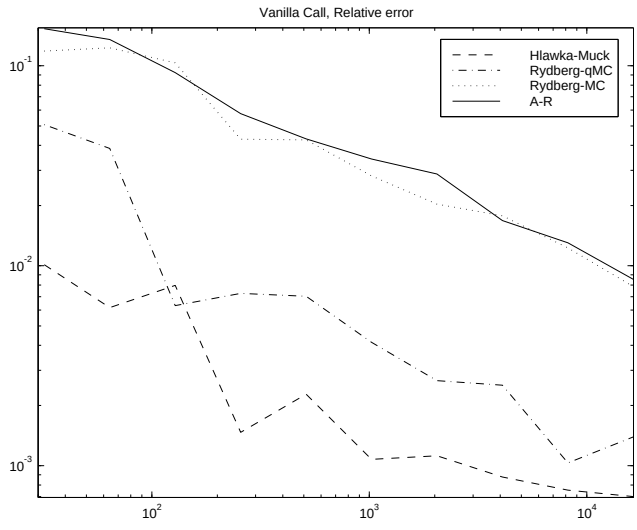

FIGURE 2. Comparison of the different methods when calculating the price of a vanilla call option

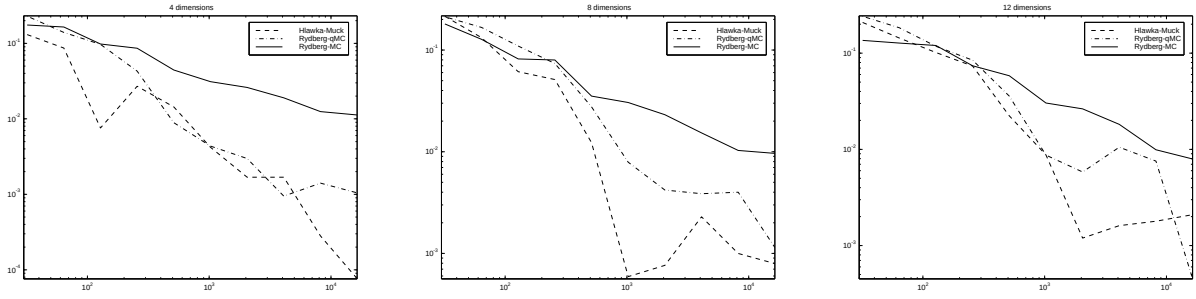

FIGURE 3. Log-Log plot of the relative errors for the Asian call option price with different sizes of the sequence. Quasi-Monte Carlo results are from a single run, Monte Carlo results are the average over 25 runs.

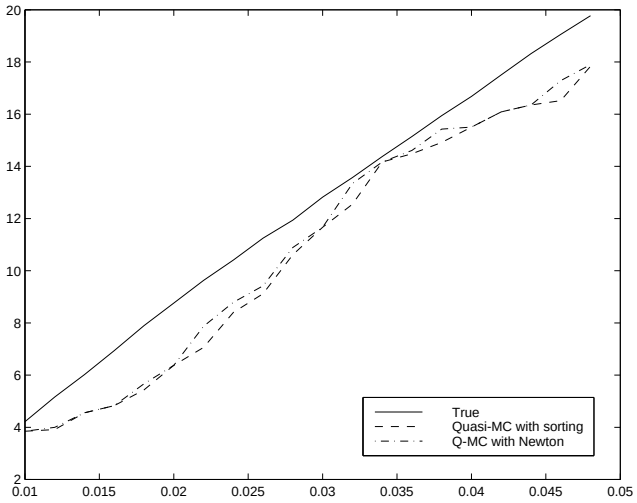

FIGURE 4. Plots of the quantile value for our test portfolio

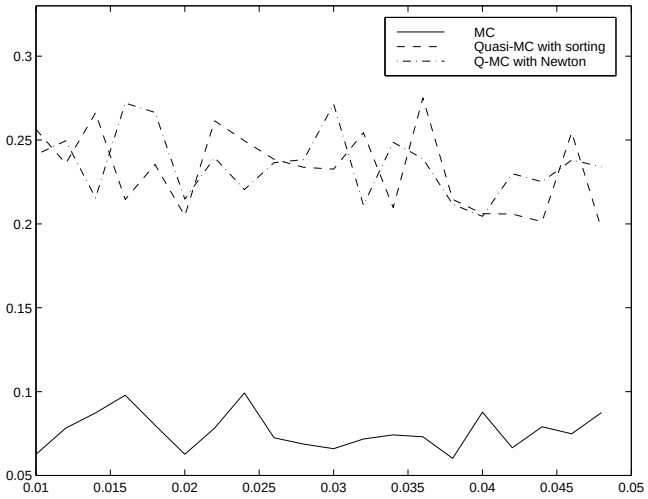

FIGURE 5. Plots of the times to calculate VaR for 1000 points

References

Barndorff-Nielsen, O. E. (1998). Processes of normal inverse Gaussian type. *Finance and Stochastics 2*, 41–68.

Eberlein, E. and U. Keller (1995). Hyperbolic distributions in finance. *Bernoulli 1*, 281–299.

Geman, H. (2002). Pure jump Lévy processes for asset price modelling. *J. Banking Finance 26*, 1257–1317. Special issue: Beyond VaR.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. New York: Springer-Verlag.

Griewank, A. (2003). A mathematical view of automatic differentiation. *Acta Numer. 12*, 1–78.

Hlawka, E. and R. Mück (1972). A transformation of equidistributed sequences. In S. K. Zaremba (Ed.), *Applications of Number Theory to Numerical Analysis*, pp. 371–388. New York: Academic Press.

Kainhofer, R. F. (2003, Apr.). *Quasi-Monte Carlo algorithms with applications in numerical analysis and finance*. Ph. D. thesis, Graz University of Technology, Austria. Advisor, Professor Robert F. Tichy.

Karatzas, I. and S. E. Shreve (1991). *Brownian Motion and Stochastic Calculus* (2nd ed.). New York: Springer-Verlag.

Michael, J. R., W. R. Schucany, and R. W. Haas (1976). Generating random variates using transformations with multiple roots. *Amer. Statist. 30*, 88–90.

Moro, B. (1995, Feb.). The full Monte. *Risk 8*(2), 57–58.

Papageorgiou, A. (2003). Sufficient conditions for fast quasi-Monte Carlo convergence. *J. Complexity 19*, 332–351.

Raible, S. (2000). *Lévy Processes in Finance: Theory, Numerics, and Empirical Facts*. Ph. D. thesis, Albert-Ludwigs-Universität, Frieburg.

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

Sobol, I. M. (1967). The distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys. 7*(4), 86–112. (Цобол′, И. М.).

Sobol, I. M. (1969). *Multidimensional Quadrature Formulas and Haar Functions*. Moscow: Izdat. "Nauka". (Цобол′, И. М. Многомерные квадратурные формулы и функции Хаара).

(Fred Espen Benth)
Centre of Mathematics for Applications
Department of Mathematics
University of Oslo
P.O. Box 1053, Blindern
N–0316 Oslo, Norway
and
Agder University College
School of Management
Serviceboks 422
N-4604 Kristiansand, Norway
E-mail address: fredb@math.uio.no
URL: http://www.math.uio.no/∼fredb/

(Martin Groth)
Centre of Mathematics for Applications
Department of Mathematics
University of Oslo
P.O. Box 1053, Blindern
N–0316 Oslo, Norway
E-mail address: martijg@math.uio.no
URL: http://www.math.uio.no/

(Paul C. Kettler)
Centre of Mathematics for Applications
Department of Mathematics
University of Oslo
P.O. Box 1053, Blindern
N–0316 Oslo, Norway
E-mail address: paulck@math.uio.no
URL: http://www.math.uio.no/∼paulck/