AN EXACT METHOD FOR THE SENSITIVITY ANALYSIS OF SYSTEMS SIMULATED BY REJECTION TECHNIQUES

MARK S. JOSHI AND DAN ZHU

ABSTRACT. We compute first and second-order sensitivities of functions simulated by rejection techniques. The methodology is to perform a measure change on every acceptance test, so that the pathwise discontinuities resulting from the rejection decisions are removed. The change of measure is chosen to be optimal in terms of minimizing variances of the likelihood ratio terms. Applications are presented for computing Greeks of equity options with Lévy-driven underlyings and sensitivities of performance measures in queueing systems. Numerical results are presented which demonstrate the efficacy and speed of the method.

1. INTRODUCTION

In recent years, there has been a surge of research into methods for estimating derivatives of expected performance measures calculated from sample paths of stochastic systems. In the case of financial products, derivatives of their prices with respect to the parameters of interest, such as the current underlying stock price and the parameters of the stock process, are called "Greeks". Banks and insurance companies with derivative exposures have to compute Greeks frequently in order to implement a sound risk management strategy. Similarly, in the case of queueing systems, there are many critical measures of performance such as the average time spent, the average queue length and the average delay in the system. Derivative estimation with respect to parameters of the service and interarrival time distributions provides a way of optimizing performance of operations and conducting sensitivity analysis of experimental outcomes.

The calculation of expected performance measures, $\Upsilon(\theta)$, involves computing the expectation of the performance measure g with respect to a certain probability distribution,

$$\Upsilon(\theta) = \mathbb{E}_\theta[g(X, \theta)] = \int g(x, \theta) f(x, \theta) dx, \tag{1.1}$$

where f is the probability density function of the random variable X. Monte Carlo simulation is a useful method for solving these problems when analytical solutions do not exist or are too complicated to compute, it simulates pseudo random numbers from the probability distribution of interest, f, and computes the performance measure as a function, $g(X, \theta)$, of the simulated random variate X(Devroye,1986). The most natural technique for generating pseudo-random numbers is the inverse-transform method, as long as the cumulative distribution function, F, is known, and the inverse of the cumulative distribution function, F^{-1}, is tractable.

The inverse-transform method is of limited application when an explicit formula for $F^{-1}(U)$ is not easily available, for a given standard uniform random variable U. Various other methods have been introduced to solve this problem:

- The acceptance-rejection method (Neumann, 1951) involves simulating a variate from a related distribution F^*, and accepting it as a variate from the target distribution F provided that it satisfies certain acceptance criteria.
- The ratio-of-uniforms method (Kinderman and Monahan, 1977) uses the ratio of two random uniforms to obtain the target distribution, provided that the point presented by the two uniforms falls within a region in the plane characterized by the probability density function.
- The transformations-with-multiple-roots method (Michael et al, 1976) assumes a known relationship between two random variables, i.e. $\chi = L(X)$. In this relationship, X is the random variable from the target distribution, and χ is another random variable from a different distribution which can be generated easily. One can obtain X by solving the known relationship for a given χ. For cases where more than one root are obtained, each root is accepted with a certain probability.

The three rejection simulation approaches have a plethora of applications, due to their simplicity. When the closed bounds of the density function can be computed easily, rejection methods are powerful at simulating pseudo random variates. One important application is the valuation of derivative products. To model stock processes, S_t for $t \geq 0$, the Variance-Gamma (VG) process and the Normal-Inverse-Gaussian (NIG) process are often used for $\log(S_t)$. They are Lévy processes with an infinite arrival rate of jumps, and Monte-Carlo simulations are often used to evaluate option prices with such underlyings. One can model these processes as time-changed Brownian motions, such that the time is a gamma random variable for the VG process and an inverse-gaussian random variable for the NIG process. Neither the gamma distribution nor the inverse-gaussian distribution has an explicit inverse-cumulative, so efficiently simulating random variates from these distributions is an intricate problem, which has been explored widely by the research community. Rejection methods by Ahrens and Dieter (1982), Cheng and Feast (1980) and Tanizaki (2008) have been introduced to simulate gamma random variates; to simulate a random variate from the inverse-gaussian distribution, the standard approach is to implement a transformation-of-multiple-roots technique suggested by Michael et al (1976).

Another rejection method called "thinning" (Lewis and Shedler, 1979) was introduced to simulate the interarrival times of a non-homogeneous Poisson process with an instantaneous intensity, $\lambda(t)$, such that $\lambda(t) \leq \lambda$ for all $t \geq 0$. Given U, a simulated random uniform, obtaining an interarrival time random variate from such distributions by the inverse-transform amounts to finding x satisfying

$$\Lambda(x) = \int_t^{t+x} \lambda(y) dt = -\log(1-U). \tag{1.2}$$

In general, the above problem can only be solved by an extensive amount of computational effort through techniques such as numerical quadrature (Ortega and Rheinboldt, 1970). The intuitive idea behind thinning is to first find a rate function $\lambda^*(t)$, which dominates the desired rate function $\lambda(t)$ and has a tractable solution to equation (1.2); next, generate interarrival time random variates with the rate $\lambda^*(t)$, and reject an appropriate fraction of them so that the desired rate $\lambda(t)$ is achieved. Ross (2006) provided a straightforward modification to the thinning method with the objective of mitigating excessive rejections.

All rejection techniques involves an acceptance test, i.e. we accept the simulated Y as a pseudo-random variate from the target distribution F if

$$a_0(\theta, Y) \leq V^D < a_1(\theta, Y), \tag{1.3}$$

where V^D is a simulated standard uniform, which we call the decision variate. Here, we assume that a_i's are smooth functions of both θ and Y, and call them critical value functions. A small change in the parameter θ may change the values of the critical value functions, and thus may change the acceptance-rejection decision. Consequently, sensitivity analysis of systems simulated by rejection techniques is difficult under traditional methods of derivative estimation. To compute the derivative of a function, $\Upsilon(\theta)$, in equation (1.1), there are three fundamental methods (Glasserman 2004), the finite differencing (FD), the likelihood ratio(LR) and the pathwise methods. The three methods are of limited application when the underlying random variables require rejection techniques to simulate:

- The FD method with a small bump size produces a biased estimator. Since a small bump in the parameter of interest may alter the acceptance-rejection decisions, the resulting performance measure from the bumped path might be materially different from the unbumped path, i.e. performance measures simulated by rejection techniques are discontinuous functions. Therefore, the resulting FD estimators of derivatives have extremely large variances if the bump size is small.
- The pathwise method can be viewed as the limit case of the FD method, and it produces unbiased estimates when it is applicable. The computation of first-order sensitivities under this approach requires the performance measure, $g(x(\theta), \theta)$, to be Lipschitz continuous everywhere and differentiable almost surely as a function of θ. This is not the case for rejection-simulated paths.
- The LR method also produces unbiased estimates, but it is only applicable when the underlying distributions have known and tractable probability density functions, $f(x, \theta)$. This is often not the case for applications which require rejection techniques. In addition, for cases where the LR method is applicable, it has a tendency to produce derivative estimates with high variances.

The three methods of derivative estimations above have been used commonly in computing Greeks of financial products, nonetheless the pathwise method when applicable produces estimators with the smallest variances among the three. In spite of its superiority for minimizing variances of the derivative estimates, for financial products with discontinuous payoffs applying the pathwise method to compute Greeks often requires additional endeavour. One approach is the Optimal Partial Proxy(OPP) algorithm by Chan and Joshi(2012); a measure change is performed at each pathwise discontinuity defined by the payoff function g, so the simulated payoff function $\hat{g}^{OPP}$ is Lipschitz continuous everywhere and differentiable almost surely, the pathwise method is then applied to $\hat{g}^{OPP}$ to compute the pathwise estimators of first-order sensitivities. Joshi and Zhu(2014) extended the OPP method to computing Hessians of financial products with angular or discontinuities payoffs(HOPP); their measure change removes the pathwise discontinuities of both the payoff function g and its first order derivatives Δ_g. The resulting simulated payoff function $\hat{g}^{HOPP}$ has first order derivatives which are Lipschitz continuous everywhere and differentiable almost surely (we shall say a function with such properties is $\hat{C}^2$), the pathwise method is then applied to $\hat{g}^{HOPP}$ to compute the pathwise estimator of the Hessian.

We note there has been much work on the adaptation of pathwise methods to discontinuous integrands. A few of these are the vibrato technique of Giles (2008), Malliavin techniques Benhamou (2003), measured-valued differentiation Heidergott et al (2010), and Wang, Fu and Marcus (2012) who use a mixed approach which reduces to likelihood ratio in one dimension. Hong and Liu (2011) approximate delta distributions in the pathwise derivatives with Gaussians. Lyuu and Teng (2008) present an approach using importance sampling but it requires one simulation per discontinuity. Glasserman (1992) presents an approach where the integrand is multiplied by a function vanishing at the discontinuities that does not change the integral's value. Whilst all of these approaches are interesting, none of them are obviously adaptable to the case of rejection sampling.

The increasing importance of sensitivity analysis in discrete-event systems makes it desirable to construct efficient and reliable simulation algorithms for derivative estimation. The application of the pathwise method to computing sensitivities of performance measures in queueing systems is widely accepted for the steady state cases (Suri and Zazanis, 1988). The pathwise method which is also called Infinitesimal perturbation analysis (IPA), estimates derivatives of performance measures with respect to model inputs from a single sample path of a discrete-event system (Ho and Cao, 1982). The consistency of IPA estimates generally depends on the validity of assuming that small parameter changes will cause small changes in state holding times along a sample path but no change in the sequence of states. This is analogous to the Lipschitz continuity condition, which we have discussed above for financial modelling. Glasserman (1991) developed a general framework for computing IPA derivative estimates of performance measures calculated from finite-time-horizon discrete-event simulations, and further provided sufficient conditions for them to be unbiased. In this paper, we consider cases that fail to satisfy Glasserman's conditions, and apply the HOPP approach to remove the pathwise discontinuities of the performance measure function.

The HOPP algorithm addresses the pathwise discontinuities of g and Δ_g, but not those of the simulation algorithm. When the state variables are simulated by rejection techniques, the simulation algorithm, $X(\theta, V)$, for turning the parameter of interest θ and a sequence of random uniform variates into the state variable is a discontinuous function of θ for a given V, and thus the resulting $\hat{g}^{HOPP}$ is no longer $\hat{C}^2$. Here, we introduce a change of measure function $U(\theta, V^D)$ at each acceptance test to replace the decision variate V^D in equation (1.3), so the small bump in the parameter of interest does not alter the acceptance-rejection decision and the pathwise discontinuities are removed up to the third order. The resulting pathwise performance measure is

$$\hat{g}(\theta) = g(\hat{X}(\theta, V), \theta)\mathbb{W}(\theta, V) \tag{1.4}$$

where $\hat{X}$ is the state variables simulated under the new scheme and $\mathbb{W}$ is the likelihood ratio weight as a result of the sequence of measure changes. The pathwise method can now be applied to $\hat{g}$ to compute the pathwise Hessian, as it is now $\hat{C}^2$. To constrain variances of the derivative estimates, the change of measure function U is chosen to minimize the variance of the likelihood ratio term $\mathbb{W}$.

Our objective is to compute both first and second-order sensitivities. For multiple parameters of interest, another obstacle to calculating the Hessian is the computational complexity required to obtain the entire matrix of second-order sensitivities. Even for cases where the pathwise method is applicable, computing each entry individually requires a prohibitive amount of effort. For example, the average time spent in a queue depends on the parameters of both the service time and the interarrival time distributions, it will be preferable to compute sensitivities of all parameters simultaneously on each sample path. Efficient simulation algorithms for computing Hessians has long been the interest to the community, we refer the readers to Griewank and Walther (2008) for an overview. Here, we adopt the Algorithmic Hessian approach introduced by Joshi and Yang (2010) for convenience. Their methodology is to decompose the evolution into elementary operations, then initialize the Hessian and gradient at the terminal time point and update them in a backward fashion with one elementary operation each step. We name the simulation algorithm resulting from combining our discontinuity removed rejection algorithm with the Joshi–Yang method, the Optimal Sensitivities for Rejection Sampling (OSRS) method.

We emphasize the widespread applicability of OSRS: it does not require the simulation algorithm to be continuous; in conjunction with OPP, the pay-off function need not be continuous either; the marginal density is not explicitly needed; and the measure changes are benign and result in much smaller variances than LR when both are applicable.

The remaining sections of the report are organized as follows. The basic idea of OSRS is presented in Section 2. In Section 3, we apply our OSRS to computing sensitivities of call options and barrier options with Lévy-driven underlyings. In Section 4, we apply the model to compute parameter sensitivities of the average time spent of a finite-time horizon $M_t|M|1$ queue, where the interarrival time is simulated by the thinning technique.

2. The Basic idea of OSRS

2.1. **A simple example: removing discontinuities of the rejection algorithm.** The objective of our OSRS algorithm is to remove pathwise discontinuities of the performance measures simulated by rejection techniques up to the third order, so the pathwise method can be applied to compute first and second-order derivatives. To illustrate the idea, we consider the case where the performance measure is the random variable itself, that is

$$\Upsilon(\theta) = \mathbb{E}[X(\theta, V)],$$

with only one parameter of interest $\theta \in \mathbb{R}$ within a small neighbourhood Θ about the base point θ_0. Here, $X(\theta, V)$ is the original rejection algorithm for turning the parameter of interest θ and a sequence of random uniforms into the random variable. It is a discontinuous function of θ, thus it is not valid to apply the pathwise method to compute derivative estimates. We construct a new simulation algorithm, $\hat{X}(\theta, V)$, such that

- $\hat{X}(\theta, V)$ is $\hat{C}^2$,
- $\hat{X}(\theta, V) - X(\theta, V) = \mathcal{O}(|\theta - \theta_0|^3)$,

to which we can apply the pathwise method to compute unbiased first and second-order derivative estimates.

We assume that

- there exists a critical value function, $a\big(Y(\theta, V_i), \theta\big)$, which is a C^2 function of both y and θ;
- the ith outcome is accepted if the decision variate $V_i^D < a\big(Y(\theta, V_i), \theta\big)$ and rejected otherwise;
- $Y(\theta, V_i)$ is the algorithm for turning a standard uniform random variate V_i and the parameter of interest θ into the simulated outcome. It is a C^2 function of θ and a differentiable function of V_i.

Under such construction, we can write

$$X(\theta, V) = Y(\theta, V_N^D)\mathbb{I}_{V_N^D < a\left(Y(\theta, V_N), \theta\right)} \prod_{i=1}^{N-1} \mathbb{I}_{V_i^D > a\left(Y(\theta, V_i), \theta\right)}, \text{ for } N \in \mathbb{Z}, \tag{2.1}$$

that is, we accept the Nth simulated outcome, $Y(\theta, V_N)$, as a random variable from the target distribution and reject the first $N-1$ simulated outcomes. The value $N(\theta) \in \mathbb{Z}$ is itself a discrete random variable, which depends on the parameter of interest, i.e. a small bump in θ may change the number of rejections required before an outcome is accepted as a random variate from the target distribution.

To remove the pathwise discontinuities of X, it is necessary to ensure a small bump in the parameter of interest does not alter the acceptance-rejection decisions, i.e. the values of the indicator functions. To achieve this, we replace the decision variate V_i^D by a function, $U_i(\theta, V_i^D)$, of it for $i = 1, 2, ..., N$ satisfying

(1) it is twice differentiable as a function of θ and piecewise differentiable as a function of V_i^D;
(2) it is bijective on $[0, 1]$ for fixed θ;
(3) $U_i\Big(\theta, a\big(Y(\theta_0, V_i), \theta_0\big)\Big) = a\big(Y(\theta, V_i), \theta\big)$;
(4) $U_i(\theta_0, V_i^D) = V_i^D$;
(5) $\frac{\partial}{\partial\theta}U_i(\theta, V_i^D)_{|\theta=\theta_0, v=a(Y(\theta_0,V_i),\theta_0)} = \frac{\partial}{\partial\theta}a\big(Y(\theta, V_i), \theta\big)_{|\theta=\theta_0}$;
(6) $\frac{\partial^2}{\partial\theta^2}U_i(\theta, V_i^D)_{|\theta=\theta_0, v=a\left(Y(\theta_0,V_i),\theta_0\right)} = \frac{\partial^2}{\partial\theta^2}a\big(Y(\theta, V_i), \theta\big)_{|\theta=\theta_0}$.

As a result of the sequence of measure changes, the limit does not move up to the third order, i.e.

$$
\begin{aligned}
&V_i^D > a\big(Y(\theta_0, V_i), \theta_0\big) \Longleftrightarrow U_i(\theta, V_i^D) > a\big(Y(\theta, V_i), \theta\big) + \mathcal{O}\big(|\theta - \theta_0|^3\big) \\
&V_i^D < a\big(Y(\theta_0, V_i), \theta_0\big) \Longleftrightarrow U_i(\theta, V_i^D) < a\big(Y(\theta, V_i), \theta\big) + \mathcal{O}\big(|\theta - \theta_0|^3\big)
\end{aligned} \tag{2.2}
$$

for $i = 1, 2, ...N(\theta_0)$. We therefore always use the unbumped paths events for the bumped paths, i.e. the acceptance-rejection decisions of the bumped paths are determined by the unbumped paths' decisions. Whilst this results in a bias, it is of order $\mathcal{O}\Big(|\theta - \theta|^3\Big)$ as $\theta \to \theta_0$, and so does not affect the computation of gradients and Hessians at θ_0.

For N acceptance tests in a sample path, we need to perform the change of measure N times. The resulting performance measure is

$$\hat{X}(\theta, V) = \mathbb{X}(\theta, V)\mathbb{W}(\theta, V) \tag{2.3}$$

where

$$\mathbb{X}(\theta, V) = Y(\theta, V_N)\mathbb{I}_{U_N(\theta,V_N^D)<a\left(Y(\theta,V_N),\theta\right)} \prod_{i=1}^{N-1} \mathbb{I}_{U_i(\theta,V_i^D)>a\left(Y(\theta,V_i),\theta\right)},$$

and

$$\mathbb{W}(\theta, V) = \prod_{i=1}^{N} \frac{\partial}{\partial v}U_i(\theta, V_i^D).$$

By (4), we have $U(\theta_0, V_i^D) = V_i^D$ so $\mathbb{W}(\theta_0) = 1$.

Now, we have made the performance measure simulated under the new scheme a $\hat{C}^2$ function, which is consistent with the conditions of the pathwise method. The resulting first-order derivative estimate is

$$\Upsilon'(\theta_0) = \mathbb{E}\Big[\frac{\partial\mathbb{X}(\theta_0, V)}{\partial\theta}\Big] + \mathbb{E}\Big[\mathbb{X}(\theta_0, V)\frac{\partial\mathbb{W}(\theta_0, V)}{\partial\theta}\Big]. \tag{2.4}$$

Similarly, we derive the second-order derivative as

$$\Upsilon''(\theta_0) = \mathbb{E}\Big[\frac{\partial^2\mathbb{X}(\theta_0, V)}{\partial\theta^2}\Big] + 2\mathbb{E}\Big[\frac{\partial\mathbb{X}(\theta_0, V)}{\partial\theta}\frac{\partial\mathbb{W}(\theta_0, V)}{\partial\theta}\Big] + \mathbb{E}\Big[\mathbb{X}(\theta_0, V)\frac{\partial^2\mathbb{W}(\theta_0, V)}{\partial\theta^2}\Big]. \tag{2.5}$$

2.2. **Optimization.** Our approach presented in the previous section can be viewed as a combination of the pathwise method and the LR method. The change of measure performed is to shift some of the dependence of θ from the performance measure to the probability density function, so we can apply the pathwise method to the performance measure function away from the discontinuities and the LR method near the discontinuities. Our next objective is to choose the optimal change of measure functions $U_i(\theta, v)$'s such that the variance of the Monte-Carlo implementation is minimized. As discussed by Glasserman (2004), the pathwise method when applicable produces estimators with lower variances than the LR estimators, therefore it is preferable to use the regularity of the performance measure away from the discontinuity as much as possible. This is however dependent on the structure of the performance measure function.

Chan and Joshi (2012) showed that the optimal choice for U_i is

$$U_i(\theta, V_i^D) = \frac{a\big(Y(\theta, V_i), \theta\big)}{a\big(Y(\theta_0, V_i), \theta_0\big)} V_i^D, \text{ if } V_i^D < a\big(Y(\theta_0, V_i), \theta_0\big) \tag{2.6}$$

$$U_i(\theta, V_i^D) = \frac{1 - a\big(Y(\theta, V_i), \theta\big)}{1 - a\big(Y(\theta_0, V_i), \theta_0\big)} \Big(V_i^D - a\big(Y(\theta_0, V_i), \theta_0\big)\Big) + a\big(Y(\theta, V_i), \theta\big),$$

otherwise, to minimize the expression

$$\int_0^1 \Big(\frac{\partial U_i(\theta, v)}{\partial v}\Big)^2 dv.$$

Consequently for cases where the continuous part of the performance measure function is constant, this change of variable function minimizes the variance of the likelihood ratio part of the first and second-order derivative estimates (Joshi and Zhu, 2014).

With such piecewise-linear change of measure functions, the first and second-order derivatives are

$$\Upsilon'(\theta_0) = \mathbb{E}\Big[\frac{\partial \mathbb{X}(\theta_0, V)}{\partial \theta}\Big] + \mathbb{E}\Bigg[\mathbb{X}(\theta_0, V)\Big(\sum_{i=1}^{N} L_{i,1}(\theta_0, V)\Big)\Bigg] \tag{2.7}$$

$$\Upsilon''(\theta_0) = \mathbb{E}\Big[\frac{\partial^2 \mathbb{X}(\theta_0, V)}{\partial \theta^2}\Big] + 2\mathbb{E}\Bigg[\frac{\partial \mathbb{X}(\theta_0, V)}{\partial \theta}\Big(\sum_{i=1}^{N} L_{i,1}(\theta_0, V)\Big)\Bigg] \tag{2.8}$$

$$+\mathbb{E}\Bigg[\mathbb{X}(\theta_0, V)\Big(\sum_{i=1}^{N} L_{i,2}(\theta_0, V) + \sum_{i=1}^{N} L_{i,1}(\theta_0, V)\sum_{j=1}^{N} L_{j,1}(\theta_0, V)\Big)\Bigg]$$

where

$$L_{i,1}(\theta_0, V) = \frac{\partial}{\partial \theta} \log\Big(1 - a\big(Y(\theta, V_i), \theta\big)\Big)_{|\theta=\theta_0} \text{ for } i < N,$$

$$L_{N,1}(\theta_0, V) = \frac{\partial}{\partial \theta} \log\Big(a\big(Y(\theta, V_N), \theta\big)\Big)_{|\theta=\theta_0} \text{ for } N \in \mathbb{Z},$$

and

$$L_{i,2}(\theta_0, V) = \frac{\partial^2}{\partial \theta^2} \log\Big(1 - a\big(Y(\theta, V_i), \theta\big)\Big)_{|\theta=\theta_0} \text{ for } i < N,$$

$$L_{N,2}(\theta_0, V) = \frac{\partial^2}{\partial\theta^2} \log\Big(a\big(Y(\theta, V_N), \theta\big)\Big)_{|\theta=\theta_0} \text{ for } N \in \mathbb{Z}.$$

2.3. **The generalization.** In our general framework, we allow

- multiple parameters of interest, $\theta \in \mathbb{R}^m$ within a small neighbourhood Θ about the base point θ_0;
- the state variables $S \in \mathbb{R}^n$ to be evolved over multiple steps, i.e. the evolution of state variables over the sample path is

$$K_Q = k_Q \circ k_{Q-1} \circ ... \circ k_1,$$

  where the number of steps Q may be a random variable $Q(\theta) \in \mathbb{Z}$ depending on the parameter of interest,
- the evolution k_i to comprise the following two sub-mapping functions:

$$k_i = O_i \circ R_i, \text{ for } i = 1, 2, ..., Q,$$

  such that R_i is a rejection algorithm for simulating one state variable and O_i is an ordinary vector function for evolving other state variables;
- more than one state variable to be evolved by rejection algorithms with different R_i's;
- non-smooth performance measure functions $g(S(\theta), \theta)$.

A. Model Assumptions

First, we establish the set of functions that our OSRS algorithm is applicable to.

a) *The rejection algorithms R_i's*

We assume

$$R_i : (0,1)^M \times \Theta \times \mathbb{R}^n_{S_{i-1}} \to \mathbb{R}^n_{S_i^*},$$

such that

(1) it only evolves one state variable by a rejection algorithm and is an identity mapping in all other state variables;
(2) R_i and R_j are conditionally independent evolutions for $j \neq i$, i.e. independent sequences of uniforms are used for each rejection sampling algorithm;
(3) R_i is the product of a continuous function of θ and a sequence of $N_i \in \mathbb{Z}$ indicator functions resulting from the acceptance-rejection decisions as shown in equation (2.1), where N_i is the number of acceptance test performed in R_i;
(4) there exist a sequence of twice-differentiable critical value functions such that the jth simulated outcome is accepted if

$$a_i^{(f-1)}(\theta) < V_{i,j}^D \leq a_i^{(f)}(\theta), \text{ for } f = 1, 2, ...M_i$$

and rejected otherwise, where M_i is the number of critical value functions for R_i and $V_{i,j}^D$ is the decision variate of the jth test.

These assumptions ensure that the only discontinuities in R_i arise from the acceptance-rejection decisions. Further, we require the critical value functions at each acceptance test to be C^2 in θ, so they can be used to define the corresponding C^2 change of variable functions.

b) *The ordinary evolutions O_i's*

We assume

$$O_i : (0,1)^{M^*} \times \mathbb{R}^n_{S_i^*} \times \Theta \to \mathbb{R}^n_{S_i},$$

such that it is a C^2 function of both S_i^* and θ for a fixed sequence of standard uniforms $V_i \in (0,1)^{M^*}$, where S_i^* is the state variables just before O_i.

c) *The performance measure function g*

We assume

$$g : \mathbb{R}^{n\times Q}_S \times \Theta \to \mathbb{R},$$

such that it is a C^2 function of θ for fixed S. As a function of S, we assume it satisfies the conditions set out in Joshi and Zhu (2014) for the application of HOPP(2) algorithm. For example, the payoff of a barrier option is continuous except when the stock value S_i equals to the barrier, and has angularities when the terminal stock value, S_T, equals to the strike.

With the above assumptions satisfied, we have obtained a class of functions that have regular behaviours away from the points of discontinuities. Discontinuities only occur either at the acceptance test of the rejection algorithms, R_i, or as the state variables pass across certain strata sets defined by the performance measure function, g. We now proceed to define a general OSRS algorithm.

B. Removing the discontinuities in acceptance-rejection decisions

First, we consider the discontinuities in the acceptance test, we perform a sequence of measure changes to remove the pathwise discontinuities of both R_i and R_i'. Each decision variate $V_{i,j}^D$ of R_i, for $j = 1, 2, ...N_i$, is replaced by a change of variable function, $U_i(\theta, V_{i,j}^D)$, of it such that

$$U_i(\theta, V_{i,j}^D) = \frac{a_i^{(f)}(\theta) - a_i^{(f-1)}(\theta)}{a_i^{(f)}(\theta_0) - a_i^{(f-1)}(\theta_0)}\left(V_{i,j}^D - a_i^{(f-1)}(\theta_0)\right) + a_i^{(f-1)}(\theta), \tag{2.9}$$

with the corresponding likelihood ratio weight

$$\frac{\partial U_i(\theta, V_{i,j}^D)}{\partial v} = \frac{a_i^{(f)}(\theta) - a_i^{(f-1)}(\theta)}{a_i^{(f)}(\theta_0) - a_i^{(f-1)}(\theta_0)},$$

if $a_i^{(f)}(\theta_0) \le V_{i,j}^D < a_i^{(f-1)}(\theta_0)$, for $f = 1, 2, ..., M_i$. Here, we shall call the OSRS version of the rejection algorithm, $\hat{R}_i$, it is $\hat{C}^2$ for both θ and S_{i-1}.

The simulation algorithm for state variables is $\hat{C}^2$ after replacing the R_i's by $\hat{R}_i$'s; this is because the composition of $\hat{C}^2$ functions is $\hat{C}^2$. The resulting performance measure can now be expressed as

$$\hat{g}^*(\theta) = g(\hat{S}^*(\theta), \theta) \prod_{i=1}^{Q} \prod_{j=1}^{N_i} \frac{\partial U_i(\theta, V_{i,j}^D)}{\partial v}, \tag{2.10}$$

where $\hat{S}^*(\theta)$ is the simulated state variables with $\hat{R}_i$'s replacing R_i's. The pathwise method can now be applied to g^* to compute the unbiased pathwise derivative estimators if the performance measure function is Lipschitz continuous.

C.Removing the discontinuities of the performance measure function

If g is a $\hat{C}^2$ function, we are now done. However, we have allowed g to be non-smooth in our general framework; in order to create an overall $\hat{C}^2$ performance measure, further work is required to remove the pathwise discontinuities of g with respect to the state variables. Here, we adopt the HOPP(2) algorithm of Joshi and Zhu (2014), that is we perform one additional change of measure at each time step to remove the pathwise discontinuities of both g and g'. This additional change of measure on each step is performed on a standard uniform other than the set of decision variates. The appropriate choice of such standard uniforms needs to be considered case by case, which depends on the evolution algorithm as well as the discontinuities of the performance measure function. The chosen standard uniform V_i^* of step i is replaced by another change of variable function, $U_i^*(\theta, V_i^*)$, of it, so that the limits of the performance measure function do not move with the parameters of interest θ up to the third order.

Finally, we have derived a new simulation algorithm for the performance measure function that is $\hat{C}^2$, and the resulting performance measure is

$$\hat{g}(\theta) = g(\hat{S}(\theta), \theta)\mathbb{W}(\theta, V), \tag{2.11}$$

where $\hat{S}(\theta)$ is the simulation algorithm for the state variables under the new scheme and $\mathbb{W}$ is the product of the likelihood ratio weights resulting from the measure changes performed for removing the discontinuities of the rejection algorithms and the performance measure function, i.e.

$$\mathbb{W}(\theta, V) = \Big(\prod_{i=1}^{Q}\prod_{j=1}^{N_i} \frac{\partial U_i(\theta, V_{i,j})}{\partial v}\Big)\prod_{i=1}^{N} \frac{\partial U_i^*(\theta, V_i^*)}{\partial v}.$$

The change of variables U_i and U_i^* for $i = 1, 2, ..., Q$, at each step have removed the limit dependence on θ of both g, g', K_N and K_N', so a small bump in the parameter of interest does not cause the bump path and the unbumped path finishing on different sides of discontinuities. The resulting $\hat{g}(\theta)$ now satisfies Glasserman's conditions for applying the pathwise method to computing first and second-order sensitivities

(1) the first-order derivatives of $\hat{g}(\theta)$ are Lipschitz continuous everywhere;
(2) the performance measure $\hat{g}(\theta)$ is twice-differentiable almost surely;
(3) the FD version of Gamma and Cross-Gamma estimators (Glasserman, 2004) under the new scheme are uniformly integrable.

We can now apply the pathwise method to construct unbiased estimates of Hessian under OSRS. The OSRS estimator of Cross-Gamma is given by

$$\mathbb{E}\left[\frac{\partial^2 \hat{g}(\theta)}{\partial\theta_s \partial\theta_k}\right]. \tag{2.12}$$

2.4. **Inexplicit critical value functions.** Our simulation algorithm is defined in terms of the discontinuities of the rejection technique, i.e. the critical value functions a_i^f's; up to this point, we have assumed that they can be found explicitly. For situations where this is not the case, we adopt Newton-Raphson's method with two steps to numerically approximate these critical value functions. Assume that there exists a differentiable function $\eta_i(V_{i,j}^D, \theta)$ for each rejection algorithm with strictly positive or negative derivatives with respect to the decision variate such that:

- if
$$\eta_i(V_{i,j}^D, \theta) > 0,$$
the simulated jth outcome is accepted and rejected otherwise,
- alternatively if
$$\eta_i(V_{i,j}^D, \theta) < 0,$$
the simulated jth outcome is accepted and rejected otherwise.

Here, we shall call η_i, the proxy constraint function. We approximate the critical values of $V_{i,j}^D$ such that $\eta_i = 0$ with the following procedures.

(1) *First, we rewrite the proxy constraint function as a function of a standard normal random variable, Z, such that*
$$\eta_i(Z, \theta) = \eta_i(\Phi(Z), \theta),$$
where Φ is the standard normal cumulative density function. We work with $Z = \Phi^{-1}(V_{i,j})$ to ensure that the resulting numerical approximation, $\hat{a}_i(\theta)$, lies between $(0, 1)$.

(2) *We approximate the proxy constraint function about the base point $Z_0 = \Phi^{-1}(\hat{V}_0)$, where $\hat{V}_0$ is some chosen initial attempt,*
$$\hat{\eta}_i^{(1)}(Z, \theta) = \eta_i(\Phi(Z_0), \theta) + \frac{\partial \eta_i(\Phi(Z_0), \theta)}{\partial V}\phi(Z)(Z - Z_0).$$

(3) *We equate this linear approximation to 0 and obtain the first estimate*
$$\hat{Z}_1(\theta) = -\eta_i(\Phi(Z_0), \theta)\Big(\frac{\partial \eta_i(\Phi(Z_0), \theta)}{\partial V}\phi(Z_0)\Big)^{-1} + Z_0.$$

(4) *We apply this method again to approximate the proxy constraint function about this new base point $\hat{Z}_1$ and then equate to 0. We obtain the second estimate,*
$$\hat{Z}_2(\theta) = -\eta_i(\Phi(\hat{Z}_1), \theta)\Big(\frac{\partial \eta_i(\Phi(\hat{Z}_1), \theta)}{\partial V}\phi(\hat{Z}_1)\Big)^{-1} + \hat{Z}_1.$$

(5) *The approximated critical value function is then given by*
$$\hat{a}_i(\theta) = \Phi\big[\hat{Z}_2(\theta)\big].$$

Since we are now working with approximated critical value functions, there is still a small possibility of having the bumped path and the unbumped path making different acceptance-rejection decisions. To resolve such problems, we adopt the approach in Joshi and Zhu (2014) which uses the unbumped path events for the bumped path, i.e. the acceptance-rejection decisions of the bumped paths are always determined by the unbumped path decisions, and this results in a bias of order $\mathcal{O}\Big(||\theta - \theta_0||^4\Big)$. It is clear that such bias vanishes for the first and second-order derivative estimates as $\theta \to \theta_0$.

2.5. **Automatic differentiation for multiple parameters of interest.** For financial product modeling, we are often interested in the sensitivities of the price with respect to more than one parameter, as well as the impact of their interactions on the price. This is because the derivatives exposures of banks and insurance companies are constantly affected by changes in interest rates, prices and the volatilities of the underlying stocks. In like manner, the performance measures of queueing systems are subject to the influence of the arrival process and the service time distributions, as well as their interactions. Computing sensitivities in such multi-dimensional settings is often demanding.

For stochastic systems computed by Monte Carlo simulations, automatic differentiation is applied to compute sensitivities of performance measures. It differentiates functions encoded as computer programs, rather than the actual formulae of the expected performance measure. It is based on the fact that all computer-program-encoded functions can be decomposed into a string of simple operations; the derivatives with respect to the input variables are then computed by a chain-rule-based technique. The adjoint version of automatic differentiation was introduced by Giles and Glasserman (2006) to derivative pricing to speed up the computation of Greeks for cases where there are a small number of outputs and a large number of inputs.

The computation of Hessians has long been explored by the research community, especially by algorithmic means. One important method was proposed by Christianson (1992), it involves n separate computations; one for the gradient of each first-order partial derivative. Another method is the Algorithmic Hessian approach introduced by Joshi and Yang(2010), their method contrasts with that of Christianson in that it computes all first and second-order derivatives simultaneously in adjoint fashion. Their key result is that, given the Hessian H_G of a function, $G : \mathbb{R}^M \to \mathbb{R}$, if l is a elementary operation which is the identity mapping in all coordinates except one and in that coordinate depends on only one or two coordinates then the Hessian of $G \circ l$, $H_{G \circ l}$ can be computed by overwriting H_G with $AM + B$ additional operations for some constants A and B depending only on the class of elementary operations. However, their method is only applicable when the integrand is $\hat{C}^2$.

For financial products with discontinuous or angular payoffs, they solved the problem by smoothing the points of discontinuity or angularity, so the resulting discounted payoff function are continuously twice-differentiable. However, the performance measure simulated by the OSRS method is $\hat{C}^2$ by construction, so we can apply the Algorithmic Hessian approach to it directly without smoothing to compute pathwise sensitivities of performance measure functions. Essentially, we decompose the algorithm into L elementary operations and apply the algorithmic Hessian method in a backward fashion to compute the Hessian estimator with $L(AM + B)$

additional operations, where M is the maximum number of variables necessary to express the state of the computation.

3. Application to computing Greeks of financial products with Lévy-driven underlying

In this section, we look at the application of OSRS to computing sensitivities of financial products with Lévy-driven underlyings. Black and Scholes (1973) demonstrated how to price European options based on the geometric Brownian motion model, but this option pricing model is inconsistent with empirical option data. As a model of stock log-returns, the NIG process was primarily introduced in the article by Barndorff-Nielsen (1997), and the VG process was introduced by Madan and Seneta (1990). They are Lévy processes with independent and stationary increments, which can be represented as the sum of three independent components: a deterministic drift, a continuous Wiener process, and a pure jump process (Sato, 1999).

The computation of option prices with Lévy models often requires Monte-Carlo simulations, and generating paths of the underlying Lévy process can be difficult when the arrival rate of the jumps is infinite, both the VG and NIG processes are such examples. One approach is to express them as time-changed (subordinated) Brownian motions, where the subordinator is a gamma process for VG (Madan et al, 1998) and a inverse-gaussian process for NIG (Barndorff-Nielsen, 1998). After conditioning on the gamma random variable for VG and the inverse-gaussian random variable for NIG, $\log(S_t)$ is distributed normally and can be generated easily. The problem is now how to efficiently simulate gamma random variates for VG and inverse-gaussian variates for NIG. Both distributions have intractable inverse cumulative functions, and various rejection techniques have been proposed to simulate these random variables.

We apply the OSRS method to compute Greeks of equity options with Lévy-driven underlyings simulated by rejection techniques. For Lévy-driven models, Glasserman and Liu(2011) proposed various exact and approximation methods for simulating first-order price sensitivities. Their two exact methods are

- the pathwise method provided that the payoff function is Lipschitz continuous everywhere and differentiable almost surely;
- the LR method provided that the underlying density function is tractable.

Their exact pathwise method requires computing parametric derivatives of a random variable, i.e. $\frac{\partial}{\partial\theta}X_\theta$, whose distribution depends on the parameter θ. Their results are based on the relationship

$$X_\theta = F_X^{-1}(U,\theta) \text{ for } U \sim U[0,1],$$

which gives $X = X_\theta$ a functional dependence on θ. The resulting derivative estimates are

$$\frac{\partial}{\partial\theta}X_\theta = -(f(X_\theta))^{-1}\frac{\partial}{\partial\theta}F(X_\theta,\theta),$$

and

$$\frac{\partial^2}{\partial\theta^2}X_\theta = -(f(X_\theta))^{-1}\Big(\frac{\partial^2}{\partial\theta^2}F(X_\theta,\theta) + 2\frac{\partial}{\partial\theta}f(X_\theta,\theta)\frac{\partial}{\partial\theta}X_\theta + \frac{\partial}{\partial X}f(X_\theta,\theta)\big(\frac{\partial}{\partial\theta}X_\theta\big)^2\Big).$$

For Lévy processes, the computation of $\frac{\partial}{\partial\theta}F(X_\theta,\theta)$ and $\frac{\partial^2}{\partial\theta^2}F(X_\theta,\theta)$ in the above two equations is often challenging. For example, computing pathwise sensitivities with VG underlyings requires efficient algorithms for approximating digamma and trigamma functions, and thus the performance of the resulting pathwise derivative estimator depends on the accuracy of the approximating algorithms used for these functions. On the other hand, their exact LR method requires closed form solutions to $\frac{\partial}{\partial\theta}\log(f(X,\theta))$, which is often not available when X is assumed to be the Lévy-distributed increment. The time-changed representation of Lévy processes is more useful in deriving these score functions, but the resulting derivative estimators have larger variances. In this paper for comparison purposes, we extend their exact methods to compute second-order sensitivities of call and barrier options.

The discounted payoff of a call option is

$$g(S_T,\theta) = \exp(-rT)(S_T - K)_+.$$

It is Lipschitz continuous everywhere and differentiable almost surely, the pathwise method is applied to compute the first-order Greeks. The payoff function however has an angularity when $S_T = K$, the combination of the pathwise and the LR method (PWLR) is applied to compute the second-order Greeks.

The discounted payoff function of a down-and-out call barrier option is

$$g(S(\theta),\theta) = \exp(-rT_Q)(S_{T_Q}(\theta,V) - K)_+ \prod_{i=1}^{Q-1}\mathbb{I}_{S_{T_i}>B},$$

where $S(\theta)$ is the evolution of the stock process over the time steps and $T_1, T_2, ..., T_Q$ are the deterministic observation dates. The payoff function is discontinuous when $S_i = B$, the LR method is applied to compute the first and second-order Greeks.

We apply our OSRS algorithm to compute first and second-order sensitivities of both call options and barrier options, where $\log(S_t)$ follows the VG and NIG processes simulated by rejection techniques. The results are compared with Glasserman and Liu (2011).

3.1. **The NIG process.** In this section, we consider the application of OSRS to compute sensitivities of a barrier option where $\log(S_t)$ follows the NIG process. We can express the evolution of NIG stock process over the observation dates $[T_{i-1}, T_i]$ under the risk neutral measure as

$$S_i = S_{i-1}\exp\Big((r+\omega+u)(T_i - T_{i-1}) + \beta I_i + \sqrt{I_i}\Phi^{-1}(U)\Big),\ \text{for } t \geq 0, \tag{3.1}$$

where I_i is an inverse-gaussian random variable with parameters $\delta(T_i - T_{i-1})$ and $\sqrt{\alpha^2 - \beta^2}$ and U is a standard uniform random variable, and

$$\omega = \delta\big(\sqrt{\alpha^2 - (\beta+1)^2} - \sqrt{\alpha^2 - \beta^2}\big).$$

Here, δ, α and β are parameters of the inverse-gaussian process (Schoutens, 2003). Michael et al (1976) introduced a transformations-with-multiple-roots method to simulate an inverse-gaussian random variable with parameters ρ and λ, the algorithm is shown below:

(1) *Generate a random variate from a standard normal distribution, i.e.* $Z = N(0,1)$*;*
(2) *set* $\chi = Z^2$ *and then calculate*

$$Y = \frac{\rho}{\lambda} + \frac{\chi}{2\lambda^2} - \frac{\sqrt{4\rho\lambda\chi + \chi^2}}{2\lambda^2};$$

(3) *generate* $V^D = U(0,1)$,
if $V^D \leq \frac{\rho}{\rho+\lambda Y}$, *then return* Y *as the* $IG(\rho,\lambda)$ *random number, else return* $\frac{\rho^2}{\lambda^2 Y}$.

Here, the critical value function is

$$a(\rho, \lambda, Y_j) = \frac{\rho}{\rho + \lambda Y_j},$$

which is a C^2 function.

We can compute sensitivities of a down-and-out call option with respect to the parameters of interest, S_0, r, u, δ, α and β. To remove the pathwise discontinuities of the payoff function, we perform an additional change of measure on U in equation (3.1) at each time step to ensure that the pathwise discontinuities are removed for both g and g'.

3.2. **The VG process.** In this section, we consider the application of OSRS to compute sensitivities of a call option where $\log(S_t)$ follows the VG process. We can express the VG stock under the risk neutral measure as

$$S_T = S_0 \exp\Big((r+\omega)T + \theta G_T + \sigma\sqrt{G_T}\Phi^{-1}(U)\Big), \tag{3.2}$$

where G_T is a gamma random variable with the shape parameter $\alpha = \frac{T}{v}$ and the scale parameter v, U is a standard uniform random variable, and

$$\omega = \frac{1}{v}\log(1 - \theta v - \frac{\sigma^2 v}{2}).$$

Here, we refer the readers to see Schoutens (2003) for detailed explanation of the VG process.

Rejection techniques for simulating random variates, X, distributed $\gamma(\alpha, 1)$ have been studied extensively. Because the behaviour of the gamma probability density function changes dramatically as the shape parameter moves across one, the appropriate rejection technique to implement depends on the value of α. In this section, we apply the OSRS method to three of them to cater for different shape parameters.

Ahrens and Dieter(1974)

This acceptance-rejection method is popular for simulating gamma random variable with shape parameter $\alpha \leq 1$, for OSRS, we use this approach for simulating gamma random variables with shape parameters $\alpha < 0.75$. The procedure is shown below

(1) *Setup:* $b = \frac{\alpha+e}{e}$.

(2) *Repeat: generate* $V_1^D = U(0,1)$ *and* $V_2^D = U(0,1)$ *independently, set*

$$\chi = bV_1^D;$$

(3) *if* $\chi > 1$,
Option One: set $Y = \chi^{\frac{1}{\alpha}}$,
if $V_2^D < \exp(-Y)$, *accept,*
otherwise,
Option Two: set $Y = -\log(\frac{b-\chi}{\alpha})$,
if $V_2^D \leq Y^{\alpha-1}$, *accept;*

(4) *until accept, return Z.*

In the above simulation algorithm, we perform change of measures on both V_1^D and V_2^D as they are both decision variates, i.e. V_1^D determines the choice of Option One and Option Two, and V_2 makes the acceptance-rejection decision. The critical value function for the first measure change on V_1^D is

$$a_1(b) = \frac{1}{b}.$$

The critical value function for the second measure change on V_2^D is

$$a_2(Y,\alpha) = \exp(-Y), \text{ Option One},$$

$$a_2(Y,\alpha) = Y^{\alpha-1}, \text{ Option Two}.$$

The $GKM1$ approach (Cheng and Feast, 1980)

This ratio-of-uniforms method is designed for gamma random variables with shape parameters $\alpha > 1$, for OSRS, we use this approach for simulating gamma random variables with shape parameters $\alpha \geq 3$. The procedure is shown below

(1) *Setup:* $\hat{\alpha} = \alpha - 1$, $b = \frac{\alpha - \frac{1}{6\alpha}}{\hat{\alpha}}$, $m = \frac{2}{\hat{\alpha}}$ *and* $d = m + 2$.

(2) *Repeat: generate* $V = U(0,1)$ *and* $V^D = U(0,1)$ *independently, set* $Y = b\frac{V^D}{V}$;

(3) *if* $m_1 V - d + Y + \frac{1}{Y} \leq 0$ *accept,*
else if $m \log(V) - \log(Y) + Y - 1 \leq 0$ *accept,*

(4) *until accept, return* $Z = \hat{\alpha}Y$.

For OSRS, the change of measure is only performed to ensure that the bumped path makes the same decision as the unbumped path in the second test, and the corresponding likelihood ratio weight is also determined by the critical value functions of the second test. This is because the first test is only a fast check that reduces the number of logarithmic evaluations, as the region defined by the first acceptance test is a subset of the region defined by the second acceptance test. Since the critical region of acceptance defined by the gamma probability density function is described by the second acceptance test function, we accept the outcome Y as long as it passes the second test. To clarify the point, we consider the following two cases

- the unbumped path passes the first test: the unbumped path accepts $Y(\theta_0)$, consequently it would also pass the second test, the OSRS change of measure ensures the bumped path passes the second test accordingly and accepts the corresponding $Y(\theta)$, and the likelihood ratio weight is determined by the critical values of the second test ;
- the unbumped path fails the first test: the OSRS change of variable ensures that the bumped path accepts $Y(\theta)$ if the unbumped path accepts $Y(\theta_0)$ in the second test and rejects it if the unbumped path rejects $Y(\theta_0)$, and the corresponding likelihood ratio weight is determined by the critical values of the second test.

Consequently, pathwise discontinuities of this algorithm are removed.

To ensure that the bumped paths make the same decision as the unbumped paths in the second test, a change of measure is performed on V^D whilst V is left unchanged. Observe that for the second proxy constraint function,

$$\eta(V^D) = m\log(V) - \log(Y(V^D)) + Y(V^D) - 1,$$

for a given $V \in (0,1)$, we do not have an explicit solution for the critical values such that $\eta(V^D) = 0$. Here, we need to apply the numerical technique in Section 2.4 to approximate them. Some simple analysis on $\eta(V^D)$ shows that for a given fixed V,

- as $V^D \to 0$, it approaches ∞;
- as $V^D \to 1$, it is always greater or equals to zero;
- the minimum of the function occurs when $V^D = \frac{V}{b} \in (0,1)$, at this point, the function is less than 0.

Consequently, there are two critical values $a^{(1)} \in (0, \frac{V}{b})$ and $a^{(2)} \in (\frac{V}{b}, 1)$ such that $\eta(a^{(1)}) = 0$ and $\eta(a^{(2)}) = 0$. Since η has strictly negative derivatives within the domain $(0, \frac{V}{b})$, we approximate $a^{(1)}$ using an initial attempt $Z_0 = \Phi^{-1}(\frac{V}{2b})$; it has strictly positive derivatives within the domain $(\frac{V}{b}, 1)$, so we approximate $a^{(2)}$ using a starting guess $Z_0 = \Phi^{-1}(\frac{V}{2b} + \frac{1}{2})$.

Tanizaki (2008) Approach

This ratio-of-uniforms approach caters for arbitrary shape parameters. The procedure is shown below

(1) *Setup:* $n = \frac{1}{\alpha} + \frac{\alpha - 0.4}{3.6\alpha}$, $b_1 = \alpha - \frac{1}{n}$, $b_2 = \alpha + \frac{1}{n}$, $C_1 = b_1\left(\frac{\log(b_1)-1}{2}\right)$ *and* $C_2 = b_2\left(\frac{\log(b_2)-1}{2}\right)$;

(2) *Repeat: generate* V *and* V^D *independently from* $U(0,1)$, *set* $w_1 = C_1 + \log(V)$, $w_2 = C_2 + \log(V^D)$, *and* $Y = n(b_1 w_2 - b_2 w_1)$;

(3) *if* $Y < 0$, *reject*
else set $X = n(w_2 - w_1)$, *and if* $\log(Y) \geq X$, *accept;*

(4) *until accept, return* $\exp(X)$.

In the above algorithm $\log(Y) \geq X \Rightarrow Y > 0$, therefore we focus on the second acceptance test that is $\log(Y) \geq X$. Given a fixed V, the proxy constraint function η is

$$\eta(V^D) = \log(b_2 w_1 - b_1 w_2(V^D)) - n w_2(V^D) + n w_1 + \log(n),$$

such that $\eta(V^D) \geq 0 \Rightarrow$ *accept*. Since the two critical value functions are inexplicit, we adopt the method in Section 2.4. Some simple analysis on the function η shows

- when $V^D = \exp(\frac{b_2}{b_1} w_1 - C_2)$, we shall call it $V_{-\infty}$, such that $\eta(V_{-\infty}) = -\infty$;
- when $V^D = 1$, η is less or equal to zero;
- when $V^D = \exp(\frac{1}{n} + \frac{b_2}{b_1} w_1 - C_2)$, we shall call it V_{max}, such that $\eta(V_{max})$ is at its maximum which is always greater than 0.

We numerically approximate $a^{(1)}$ with initial attempt $Z_0 = \Phi^{-1}(0.5(V_{-\infty} + V_{max}))$, and approximate $a^{(2)}$ with initial attempt $Z_0 = \Phi^{-1}(0.5V_{max} + 0.5)$.

For a call option with $\log(S_t)$ following the VG process, we perform a change of measure on U in equation (3.2) to remove the pathwise discontinuities in the first-order derivatives of the payoff function.

3.3. **Comparison of OSRS and Traditional Sensitivity Computation Approaches.** For the NIG process, we apply OSRS to compute the gradient and the Hessian of a down-and-out call option by setting the parameters $S_0 = 100$, $K = 100$, $r = 0.1$, $\delta = 0.31694$, $\alpha = 28.42$, $\beta = -15,0862$, $u = 0.05851$ and $T = 1$, the barrier level equals 80 and 12 observation dates are equally spaced; the numerical results are shown in Table 1 and 2. The payoff function of the barrier option is discontinuous, we do not have the luxury of the pathwise method for computing even the first-order sensitivities; here we compare the results computed by the OSRS algorithm with the gradient and Hessian computed by the exact LR method suggested by Glasserman and Liu(2011). The results show that, the OSRS algorithm produces unbiased estimators of first and second-order Greeks of a barrier option with $\log(S_t)$ following the NIG process. Observe that the gradient and Hessian computed by OSRS converge to the results by the LR method. Our OSRS produces smaller standard errors for both the first and second-order sensitivities.

To demonstrate the speed of the OSRS algorithm, we show that the time taken to compute the price by the transform-of-uniforms method is 0.215 seconds and the time taken for computing all the Greeks by OSRS is 1.418 seconds, both tests are performed with the sample size of 20,000 paths and the same set of parameters as stated above.

In the Figure 1 and 2, we plot the sum of standard errors of the gradient and the Hessian computed by OSRS method and the LR method, with varying number of observation dates from 4 to 12 over a one year horizon. For the same set of numbers of observation dates, the maximum sum of standard errors of the gradient calculated by OSRS is 0.78347 when the number of observation dates is 12 compared to 7.473587 by the LR method; the minimum is

0.70147 when the number of observation dates is 4, compared to 5.89543 by the LR method. Similarly, the maximum sum of standard errors of the Hessian calculated by OSRS is 29.20628 when the number of observation dates is 12 compared to 516.9784 by the LR method; the minimum is 15.24117 when the number of observation dates is 4, compared to 317.8014 by the LR method.

We observe that the LR method produces a larger standard error as the number of observation dates increases, this is because

- the number of likelihood score functions increases with the number of time steps, as more random variables I_i depend on the parameters of interest;
- we have fixed T, the inter-observation times $T_i - T_{i-1}$ reduces as the number of observation dates increases, and hence the simulated I_i's are more likely to be small and the individual score functions of I_i have larger variances.

Our OSRS, on the other hand, produces estimators for the gradient and the Hessian with relatively stable standard errors.

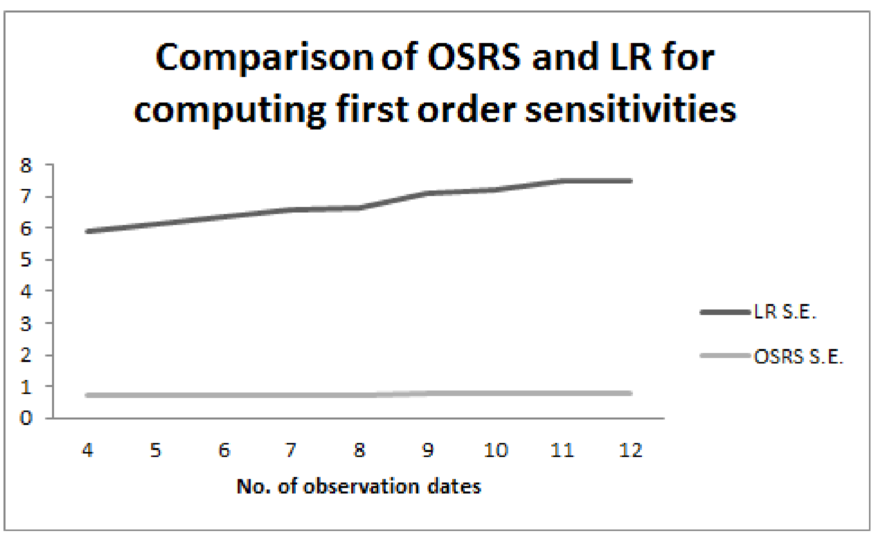

FIGURE 1. Barrier option with NIG underlying: the sum of standard errors of the first-order sensitivities computed by OSRS and LR with 20,000 path sample

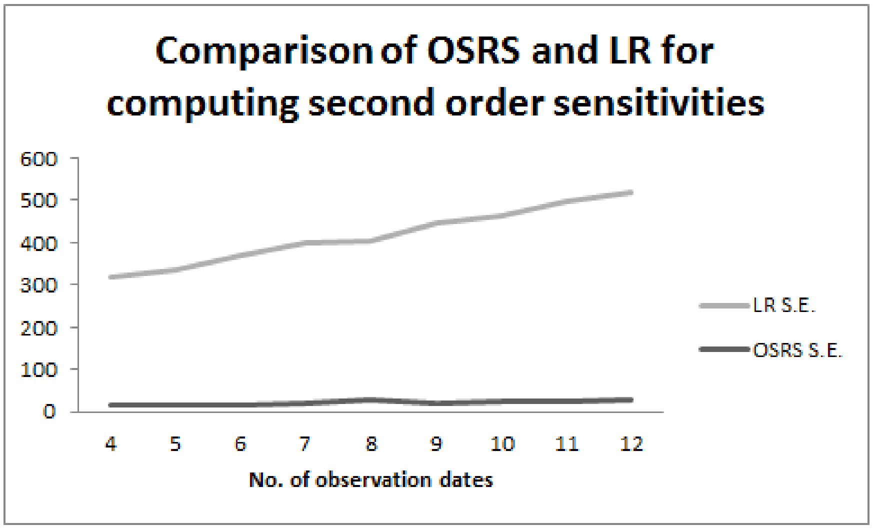

FIGURE 2. Barrier option with NIG underlying: the sum of standard errors of the Hessian computed by OSRS and LR with 20,000 path sample

For the VG process, we set the parameters $S = 100$, $K = 100$, $\theta = -0.15$, $r = 0.05$, $\sigma = 0.2$ and $v = 1$. For the Greeks of a call option, the payoff function is Lipschitz continuous everywhere and differentiable almost surely, which allows us to apply the pathwise method by Glasserman and Liu(2011) to compute the gradient with the smallest variances. We then combine their pathwise and LR approaches to compute the Hessian estimator, and compare it to our OSRS results. We consider the case with

- a small shape parameter, i.e. $\alpha = 0.7$, that is $T = 0.7$, the gamma random variate is simulated by the Ahrens-Dieter method and the results are summarized in Table 3 and 4;
- a large shape parameter, i.e. $\alpha = 4.5$, that is $T = 4.5$, the gamma random variate is simulated by the GKM1 method and the results are summarized in Table 5 and 6;
- the shape parameter α equals one, that is $T = 1$, the gamma random variate is simulated by Tanizaki's method and the results are summarized in Table 7 and 8.

The gamma random number generators are classified depending on the shape parameter α, this is because the probability density function of a gamma random variable with $\alpha \leq 1$ is very different from the PDF of a gamma random variable with $\alpha > 1$. The Ahrens-Dieter method deals with the set of gamma random variables with $\alpha \leq 1$ by an acceptance-rejection approach, and the expected number of re-calculations required per accepted sample increases as α approaches 1 from below. On the other hand, the GKM1 method deals with the set of gamma random variables with $\alpha > 1$ by a ratio-of-uniforms approach, the rectangle described by the gamma PDF narrows as α approaches 1 from above, thus an excessive number of regenerations is needed to obtain an pair of uniforms which lies within the target region. The efficacy of the OSRS algorithm depends on the performance of the rejection algorithm, so an exorbitant number of re-calculations in the rejection algorithms is clearly not desirable. Moreover, OSRS uses the behaviour of the simulation algorithm as α moves to the base point α_0, neither Ahrens-Dieter nor GKM1 allows α to move across the point $\alpha_0 = 1$ and both methods perform poorly on approach to it. To resolve this problem, we adopt the gamma generator for $\alpha \in [0.75, 3)$ by Tanizaki(2008).

The numerical results show that, the OSRS algorithm produces unbiased estimators of first and second-order Greeks of a call option with $\log(S_t)$ following the VG process. Observe that the gradient computed by OSRS converges to the values computed by the pathwise method and the Hessian computed by OSRS converges to the results computed by PWLR. OSRS outperforms the PWLR method in computing second-order sensitivities for the three shape parameters chosen. We note that since the payoff function of a call option is continuous, it might be possible to adapt the HOPP(1) method in Joshi and Zhu (2014) to computing Hessians for the VG process, however, this would be a non-trivial extension of that work and we do not perform that analysis here.

We also demonstrate the speed of the OSRS algorithm for computing Greeks for the VG process. For a sample of 20,000 paths,

- the time taken for computing the price is 0.119 by the Ahrens-Dieter method, and the corresponding time taken to compute all the Greeks is 0.260 seconds;
- the time taken for computing the price is 0.122 by the GKM1 method, and the corresponding time taken to compute all the Greeks is 0.532 seconds;
- the time taken for computing the price is 0.134 by Tanizaki's method, and the corresponding time taken to compute all the Greeks is 0.610 seconds.

In Figure 3, we vary the maturity date from 0.5 to 5.5 to compare the sum of standard errors of the Hessian calculated by OSRS and PWLR. For the same set of maturity dates, the maximum sum of standard errors of the Hessian calculated by OSRS is 91.8616 when $T = 5.5$ compared to 207.0194 by the PWLR method; the minimum is 17.89213 when $T = 0.5$, compared to 456.2566 by the PWLR method. The results show that as $\alpha = T$ gets smaller than 1, the sum of standard errors produced by PWLR increases dramatically. This is because the simulated G_T appears in the denominator of the score function, the probability of having extremely small G_T is very high for gamma random variables with $\alpha < 1$. The OSRS method, however, does not have such problems, and produces relatively stable standard errors for second-order sensitivities.

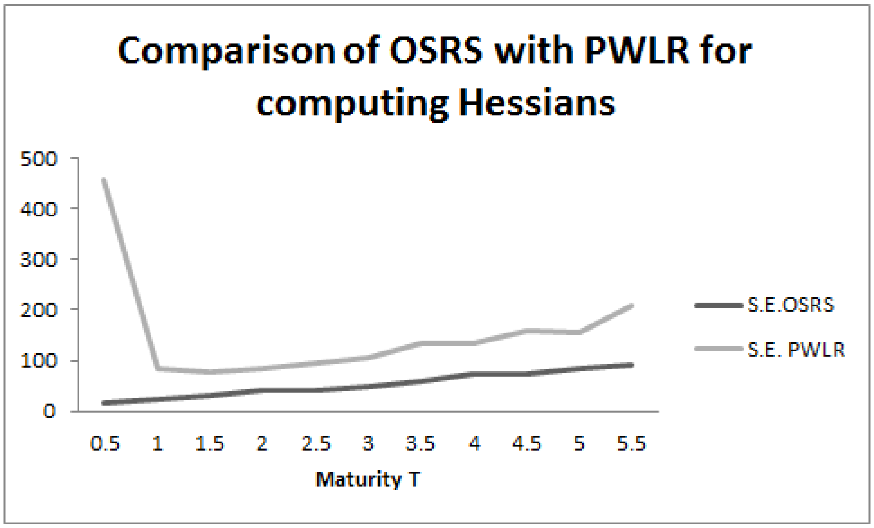

Figure 3. Call option with VG underlying: the sum of standard errors of the Hessian computed by OSRS and PWLR with 20,000 path sample

It is evident that OSRS does produce a smaller standard error than LR for computing first-order sensitivities. It also outperforms both the LR method and the PWLR method for computing second-order sensitivities. Although the pathwise method outperforms the OSRS method for computing first-order sensitivities, it would not be applicable when the payoff function is discontinuous.

4. Application to queues

In this section, we apply OSRS to computing sensitivities of performance measures of queueing systems. The motivation for this application arises from the growing emphasis on the development of increasingly sophisticated simulation techniques for the study of complex problems in operations research, which further requires researchers to develop methods for estimating sensitivities of models to variations in model parameters. Parameter estimation serves primarily two purposes in operational research:

- if a small change in a parameter results in relatively large changes in the outcomes, that parameter needs to be monitored frequently;
- if second-order sensitivities can be estimated, they can be used in optimization routines to find more efficient experimental outcomes.

The pathwise method, or IPA, has been introduced as an efficient way to compute parameter sensitivities of discrete-event systems. The main application of this approach so far is to compute sensitivities of steady state performance measures, for example, the steady state mean time spent in the system by a customer (Suri and Zazanis, 1988). Ho and Cao (1983) showed that IPA is exact in the expected value sense for a certain restricted class of queueing systems.

Performance measures calculated under finite-time horizon queueing systems often fail to fall into this restricted class. Glasserman (1991) provided a general formulation of IPA for a broad class of discrete-event systems, and stated sufficient conditions for these estimates to be unbiased. Essentially, for computing first-order sensitivities, the pathwise performance measure estimate is required to be a continuous function of θ and differentiable with probability one. In this paper, our focus is on considering queueing system performance measures simulated by rejection techniques which clearly fail to satisfy the conditions set out in literature for the application of IPA.

One interesting such example is the non-homogeneous Poisson process. It has wide applications in modelling queueing systems such as the arrival process at an intensive care unit (Lewis, 1972), the transaction processing in a data base management system (Lewis and Shedler, 1979), and bulk arrivals under parallel and series configuration (Suhasini et al, 2013). Efficient simulation algorithms of such point processes have been explored rigorously by the research community. The most popular simulation technique for non-homogeneous Poisson process was introduced by Lewis and Shedler (1979), and is called "thinning". Their method can be applied to simulating a given non-homogeneous Poisson process without the need for numerical integration. The thinning method is refined by Ross (2006) to produce a more efficient rejection algorithm. We apply OSRS to a queueing system with a non-homogeneous Poisson arrival process, where the interarrival time random variables are simulated by the thinning algorithm.

4.1. **Interarrival time.** Our first numerical example is computing sensitivities of the interarrival time with respect to the parameters of $\lambda(t)$. For benchmarking purposes, we have chosen the example so that the expected interarrival time has analytical solutions, and the first and second-order sensitivities can be derived from the analytical expected value. Define the random variable X_s as the arrival time of the next customer provided a customer just arrived at time s. Here we take the intensity function to be

$$\lambda(t) = \frac{\lambda}{1+t}, \text{ for } t \geq 0,$$

which is a continuously decreasing function bounded by λ.

To simulate X_s, we apply the thinning algorithm in Ross (2006) by adopting the majoring Poisson process with an intensity function $\lambda^*(t) = \lambda$, as follow

(1) *Let* $t = s$,
(2) *generate* $V = U(0,1)$, *set* $t = t - \frac{1}{\lambda} log(V)$;
(3) *generate* V^D,
(4) *if* $V^D \leq \frac{1}{1+t}$, *set* $X_s = t$ *and accept, else go back to step 2.*

For this simple case, the only discontinuities arise from the acceptance tests. The critical value function is

$$a(t_j, \lambda) = \frac{1}{1+t_j},$$

where t_j is the jth simulated outcome. The numerical experiment is performed by assuming $s = 2$ and $C = 3$, and the comparison with the analytical solution is shown in Table 9. The result illustrates that the OSRS applied to the thinning algorithm produces unbiased estimators, as the derivative estimates calculated by OSRS converge to the analytical solutions.

4.2. **Average time spent in an $M_t|M|1$ queue with a cyclical arrival intensity.** We now consider a more realistic example, an $M_t|M|1$ queue with the arrival intensity

$$\lambda(t) = \exp(a_0 + a_1 t + K \sin(w_0 t + \omega)), \text{ for } t \geq 0. \tag{4.1}$$

This is a log-linear function with additional cyclical behaviour. This model is found by Lewis (1972) for arrivals at an intensive care unit. The computation of $\Lambda^{-1}(x)$ in equation (1.2) is computationally demanding.

We consider the thinning technique with the following majoring function,

$$\lambda^*(t) = \exp(a_0 + a_1 t + K), \text{ for } t \geq 0,$$

for which $\Lambda^{-1}(x)$ can be calculated easily, and we obtain the cumulative density function, F_s^*, of such interarrival time random variables as well as its inverse. We simulate the interarrival time under such distributions by the inverse-transform method.

The performance measure function we consider here is the average time spent by a customer in the system within a finite-time horizon T,

$$g = \mathbb{I}_{A_{Q(\theta)}>T} \frac{1}{Q(\theta)-1} \sum_{i=1}^{Q(\theta)-1} (D_i(\theta) - A_i(\theta)) \mathbb{I}_{A_i(\theta)<T}, \text{ for } Q(\theta) \in \mathbb{Z},$$

where A_i is the arrival time and D_i is the departure time of the ith customer, $Q - 1$ is the number of customers arrived between $[0, T]$. The service time is distributed exponentially with intensity μ. This performance measure function clearly fails to satisfy Glasserman's conditions for the application of IPA. Further, the application of the LR method for computing sensitivities is intractable with the intensity function in equation (4.1), as the computation of score functions under such distributions is demanding (Rubinstein, 1992).

To simulate such discrete-event systems, we adopt the methodology suggested by Ross (2006). The subroutine for generating A_i is

(1) *let* $t = A_{i-1}$,
(2) *generate* V*, and obtain a new* t *by the inverse-transform method from* F_t^* *and the simulated* V;
(3) *generate* V^D,
(4) *if* $V^D < \exp(K(\sin(w_0 t + \omega) - 1))$*, accept* $A_i = t$,
(5) *else return to step* 2.

We remove the pathwise discontinuities caused by the rejection technique by performing measure changes on every V^D. The change of variable function is defined by the critical value function

$$a(K, w_0, t_{i,j}, \omega) = \exp(K(\sin(w_0 t_{i,j} + \omega) - 1)),$$

where $t_{i,j}$ is the jth simulated outcome of A_i.

To compute the gradient, we remove the pathwise discontinuities of g that arise from A_i passing across T. To compute the second-order sensitivities, we need to further consider the pathwise angularities that arise from A_i passing across D_{i-1}. It is important to notice that when the ith customer arrives before D_{i-1}, the time spent by him in the system equals the queueing time plus the service time, otherwise, it equals the service time. The pathwise discontinuities of g' stem from this feature of the $M_t|M|1$ system. To remove the pathwise discontinuities of g and g', we perform the change of measure on the first random uniform variate V in the simulation algorithm above, with the same approach as the HOPP(2) algorithm.

The numerical experiments are performed by taking $a_0 = 0.01$, $a_1 = 0.001$, $K = 0.654$, $W = 2\pi$, $\omega = 3.519$, $\mu = 5$ and $T = 1$. We benchmark our results with the Hessian and gradient calculated by the FD method. We adopt the symmetric FD estimators with different bump sizes, h_θ, for each parameter, see Glasserman(2004) for the detailed explanation. Here, we set $h_{a_0} = 0.001$, $h_{a_1} = 0.0009$, $h_K = 0.01$, $h_W = 0.1$, $h_\omega = 0.1$ and $h_\mu = 0.1$ for the FD estimators.

We compare the Hessian and gradient computed by OSRS using 100,000 paths and the results by the FD method with 10,000,000 paths, shown in Table 10 and 11. The FD method does not produce as good results as OSRS even with 100 times more paths, this is expected as the performance measure function is discontinuous in the parameters of interest and the bump sizes are very small. We refer the reader to detailed explanations in Glasserman (2004). OSRS produces unbiased estimators with tolerable standard errors for a reasonable number of paths.

To show the speed of the OSRS algorithm, we compare the computational time for computing sensitivities by the OSRS method and the computational time for calculating the expected performance measure with the same sample size of 20,000 paths. With the same set of parameter inputs as above, the time taken for computing the expected time spent in the system is 1.298 seconds, and the corresponding time for computing all the sensitivities is 2.258 seconds by the OSRS method.

5. Conclusion

We have introduced a simulation algorithm for computing first and second-order derivative estimates of performance measures simulated by rejection techniques. The method is to perform a measure change at each acceptance test on the decision uniform random variable, which ensures the limit does not move with the parameters of interest up to the third order; the change of variable function is also chosen to be optimal in terms of minimizing the variance of the likelihood ratio terms. As our change of measure is determined by the critical value functions not the performance measure, it has been made generic and applicable to a wide range of practical situations. We applied OSRS to computing sensitivities of options prices with Lévy-driven underlyings and the average time spent by a customer in an $M_t|M|1$ queue to demonstrate the superiority of OSRS comparing to the traditional methods of estimating derivatives. We saw that is very effective in both cases.

We would like to emphasize breadth of applicability of the new algorithm: it can be applied to situations in which the performance measure function is highly discontinuous, the underlying state variables have intractable distribution functions and none of the traditional methods are feasible.

Appendix One: Tables of Numerical Results

OSRS vs LR	S	r	u	δ	α	β
S	0.012 vs 0.027	1.172 vs 1.248	2.124 vs 2.195	-0.279 vs -0.28	6.06E-03 vs 4.46E-04	6.24E-03 vs 5.86E-03
r	1.172 vs 1.248	38.411 vs 40.184	117.052 vs 118.829	-32.22 vs -27.779	0.727 vs 0.732	0.748 vs 0.748
u	2.124 vs 2.195	117.052 vs 118.829	212.209 vs 214	-28.108 vs -22.845	0.615 vs 0.608	0.63 vs 0.628
δ	-0.279 vs -0.28	-32.22 vs -27.779	-28.108 vs -22.845	1.041 vs 16.802	-0.324 vs -0.321	-0.332 vs -0.3
α	6.09E03 vs 4.46E-03	0.727 vs 0.732	0.615 vs 0.608	-0.324 vs -0.321	0.018 vs 0.017	0.019 vs 0.018
β	6.24E-03 vs 5.86E-03	0.748 vs 0.748	0.630 vs 0.628	-0.332 vs -0.300	0.019 vs 0.018	0.021 vs 0.023
First Order	0.952 vs 0.947	78.641 vs 78.645	95.156 vs 95.171	4.113 vs 4.934	-0.112 vs -0.124	-0.118 vs -0.12

Table 1. Barrier Option with NIG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by LR using a 20,000 paths sample

OSRS vs LR	S	r	u	δ	α	β
S	1.1E-04 vs 0.023	0.014 vs 0.405	0.014 vs 0.417	0.005 vs 0.501	1.73E-04 vs 1.146E-03	1.18E-04 vs 5.34E-03
r	0.014 vs 0.405	1.468 vs 16.468	1.473 vs 16.911	0.383 vs 15.929	0.016 vs 0.151	0.014 vs 0.204
u	0.014 vs 0.417	1.473 vs 16.911	1.478 vs 17.377	0.442 vs 16.321	0.017 vs 0.155	0.015 vs 0.209
δ	4.54E-03 vs 0.501	0.383 vs 15.929	0.442 vs 16.321	0.288 vs 28.309	5.58E-03 vs 0.186	5.74E-03 vs 0.217
α	1.15E-04 vs 4.1E-03	0.016 vs 0.151	0.017 vs 0.155	5.58E-03 vs 0.186	3.1E-04 vs 2.28E-03	3.07E-04 vs 2.282E-03
β	9.39E-04 vs 5.34E-03	0.014 vs 0.204	0.015 vs 0.209	5.74E-03 vs 0.217	3.07E-04 vs 2.82E-03	2.25E-04 vs 3.11E-03
First Order	9.39E-04 vs 0.023	0.075 vs 0.705	0.092 vs 0.72	0.077 vs 0.923	2.25E-03 vs 7.73E-03	2.11E-03 vs 9.29E-03

Table 2. Barrier Option with NIG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by LR using a 20,000 paths sample

OSRS vs PWLR	S	r	σ	θ	v
S	0.017 vs 0.018	1.206 vs 1.262	-0.251 vs -0.251	-0.129 vs -0.190	0.073 vs 0.075
r	1.206 vs 1.262	52.981 vs 56.435	-30.789 vs -30.246	1.004 vs -3.229	4.93 vs 5.053
σ	-0.251 vs -0.251	-30.789 vs -30.246	34.242 vs 32.052	51.412 vs 51.737	-8.332 vs -8.035
θ	-0.129 vs -0.199	1.004 vs -3.229	51.412 vs 51.737	41.812 vs 44.064	-8.398 vs -8.057
v	0.073 vs 0.075	4.93 vs 5.053	-8.332 vs -8.035	-8.398 vs -8.057	-0.169 vs -0.358
First Order	0.731 vs 0.724	44.935 vs 44.516	18.835 vs 18.343	-14.352 vs -14.828	0.288 vs 0.294

Table 3. Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OHRS method and by PWLR using a 20,000 paths sample with $\alpha = 0.7$

OSRS vs PWLR	S	r	σ	θ	v
S	1.902E-04 vs 4.02E-03	0.013 vs 0.281	0.016 vs 0.066	0.018 vs 0.248	7.22E-03 vs 6.12E-03
r	0.013 vs 0.281	0.86 vs 17.431	0.875 vs 3.852	0.999 vs 15.358	0.434 vs 0.379
σ	0.016 vs 0.066	0.875 vs 3.852	2.305 vs 7.422	2.502 vs 4.909	0.713 vs 0.36
θ	0.018 vs 0.248	0.999 vs 15.358	2.502 vs 4.909	3.136 vs 15.574	0.538 vs 0.453
v	7.22E-03 vs 6.12E-03	0.434 vs 0.379	0.713 vs 0.36	0.538 vs 0.453	0.322 vs 0.304
First Order	5.00E-03 vs 3.73E-03	0.063 vs 0.045	0.147 vs 0.167	0.155 vs 0.132	0.054 vs 0.025

TABLE 4. Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 0.7$

OSRS vs PWLR	S	r	σ	θ	v
S	5.586E-03 vs 5.01E-03	2.457 vs 2.529	-0.235 vs -0.238	0.021vs 0.022	0.023 vs 0.023
r	2.457 vs 2.529	104.804 vs 127.357	-303.737 vs -314.878	148.647 vs 152.506	0.492 vs 0.536
σ	-0.235 vs -0.238	-303.737 vs -314.878	100.452 vs 129.214	116.11 vs 128.319	-14.21 vs -15.029
θ	0.021 vs 0.022	148.647 vs 152.506	116.11 vs 128.319	114.117 vs 108.343	-22.331 vs -22.727
v	0.023 vs 0.023	0.492 vs 0.536	-14.21 vs -15.029	-22.331 vs -22.727	-1.422 vs -1.695
First Order	0.789 vs 0.803	222.432 vs 225.786	43.952 vs 45.553	-32.273 vs -32.605	2.038 vs 1.915

TABLE 5. Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 4.5$

OSRS vs PWLR	S	r	σ	θ	v
S	5.96E-05 vs 2.23E-04	0.027 vs 0.104	0.024 vs 0.077	0.028 vs 0.051	6.54E-03 vs 6.25E-03
r	0.027 vs 0.104	10.044 vs 22.208	6.226 vs 11.84	7.324 vs 10.687	1.813 vs 1.612
σ	0.024 vs 0.077	6.226 vs 11.84	6.679 vs 38.673	7.595 vs 14.788	0.944 vs 1.07
θ	0.028 vs 0.051	7.324 vs 10.687	7.595 vs 14.788	10.847 vs 14.228	1.126 vs 1.763
v	6.54E-03 vs 6.25E-03	1.813 vs 1.612	0.944 vs 1.07	1.126 vs 1.763	0.535 vs 0.638
First Order	6.60E-03 vs 4.84E-03	1.807 vs 1.228	1.277 vs 1.562	1.451 vs 1.247	0.261 vs 0.063

TABLE 6. Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 4.5$

OSRS vs PWLR	S	r	σ	θ	v
S	0.014 vs 0.013	1.423 vs 1.344	-0.246 vs -0.193	-0.097 vs 2.77E-03	0.056 vs 0.065
r	1.423 vs 1.344	80.77 vs 73.953	-47.8 vs -43.501	7.475 vs 15.473	5.04 vs 5.85
σ	-0.246 vs -0.193	-47.8 vs -43.501	43.294 vs 48.746	63.428 vs 69.915	-8.307 vs -8.622
θ	-0.097 vs 2.77E-03	7.475 vs 15.473	63.428 vs 69.915	52.886 vs 52.869	-9.173 vs -10.089
v	0.056 vs 0.065	5.04 vs 5.85	-8.307 vs -8.622	-9.173 vs -10.089	-0.629 vs -0.443
First Order	0.728 vs 0.728	61.553 vs 61.487	23.16 vs 23.851	-17.186 vs -16.9	0.526 vs 0.513

TABLE 7. Call Option with VG underlying: Mean of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 1$

OSRS vs PWLR	S	r	σ	θ	v
S	1.52E-04 vs 1.21E-03	0.015 vs 0.121	0.017 vs 0.053	0.019 vs 0.103	3.5E-03 vs 5.87E-03
r	0.015 vs 0.121	1.371 vs 10.143	1.286 vs 3.841	1.500 vs 8.667	0.296 vs 0.496
σ	0.017 vs 0.053	1.286 vs 3.841	2.854 vs 10.288	3.153 vs 5.668	0.323 vs 0.407
θ	0.019 vs 0.103	1.500 vs 8.667	3.153 vs 5.668	3.992 vs 9.613	0.307 vs 0.534
v	3.5E-03 vs 5.87E-03	0.296 vs 0.496	0.323 vs 0.407	0.307 vs 0.534	0.143 vs 0.200
First Order	5.22E-03 vs 3.88E-03	0.433 vs 0.322	0.592 vs 0.600	0.541 vs 0.462	0.057 vs 0.028

TABLE 8. Call Option with VG underlying: Standard errors of the Hessian and first order sensitivities calculated by the OSRS method and by PWLR using a 20,000 paths sample with $\alpha = 1$

	Analytical C	Analytical T	OSRS C	OSRS T	OSRS S.E. C	OSRS S.E. T
C	0.75	-0.25	0.725	-0.246	0.045	0.019
T	-0.25	0	-0.246	8.3E-03	0.019	0.010
First Order	-0.75	1.5	-0.742	1.495	0.024	0.017

TABLE 9. Sensitivities of the expected interarrival time with a time-dependent intensity function: OSRS results with 20,000 paths sample compared with analytical value

OSRS vs FD	a_0	a_1	K	w_0	ω	μ
a_0	-0.402 vs -0.906	-1.284 vs -23.821	-0.368 vs -0.447	0.123 vs 0.028	0.053 vs 0.044	-0.024 vs -0.036
a_1	-1.284 vs -23.821	-3.693 vs -94.7	-1.173 vs -1.778	0.288 vs 1.246	0.078 vs 0.375	-7.49E-03 vs -0.566
K	-0.368 vs -0.447	-1.173 vs -1.778	-0.269 vs -8.241	0.105 vs 0.044	0.047 vs 0.056	-7.72E-03 vs -0.146
w_0	0.123 vs 0.028	0.288 vs 1.246	0.105 vs 0.044	-0.038 vs -0.061	-0.021 vs -6.59E-03	9.36E-04 vs 1.32E-03
ω	0.053 vs 0.044	0.078 vs 0.375	0.047 vs 0.056	-0.021 vs -6.59E-03	-8.06E-03 vs 0.019	-7.12E-04 vs -1.14E-03
μ	-0.024 vs -0.036	-7.49E-03 vs -0.566	-7.72E-03 vs -0.146	9.36E-04 vs 1.32E-03	-7.12E-04 vs -1.14E-03	0.013 vs 0.037
First Order	0.091 vs 0.103	0.025 vs 0.043	0.023 vs 0.043	-3.86E-03 vs -7.25E-03	2.57E-03 vs 1.60E-03	-0.033 vs -0.035

TABLE 10. Average Waiting time $M_t|M|1$ queue: Mean of the Hessian and first order sensitivities calculated by the OSRS method using a 100,000 path sample and by the FD method using a 10,000,000 path sample

OSRS vs FD	a_0	a_1	K	w_0	ω	μ
a_0	0.293 vs 4.893	0.983 vs 11.101	0.291 vs 0.998	0.061 vs 0.099	0.018 vs 0.098	1.47E-03 vs 0.098
a_1	0.983 vs 11.101	3.844 vs 603.503	0.968 vs 11.084	0.192 vs 1.109	0.041 vs 1.108	3.37E-03 vs 1.109
K	0.291 vs 0.998	0.968 vs 11.084	0.292 vs 4.895	0.060 vs 0.099	0.018 vs 0.099	1.53E-03 vs 0.099
w_0	0.061 vs 0.010	0.192 vs 1.109	0.060 vs 0.099	0.014 vs 0.049	6.99E-03 vs 0.010	3.782E-04 vs 9.98E-03
ω	0.018 vs 0.010	0.041 vs 1.108	0.018 vs 0.099	6.99E-03 vs 0.010	8.29E-03 vs 0.049	2.603E-04 vs 9.99E-03
μ	1.47E-03 vs 0.121	3.37E-03 vs 1.109	1.53E-03 vs 0.099	3.782E-04 vs 9.98E-03	2.603E-04 vs 9.99E-03	5.575E-05 vs 0.049
First Order	6.72E-03 vs 0.014	0.016 vs 0.157	7.04E-03 vs 0.014	1.71E-03 vs 1.41E-03	1.14E-03 vs 1.41E-03	1.39E-04 vs 1.42E-03

TABLE 11. Average Waiting time $M_t|M|1$ queue: Standard Errors of the Hessian and first order sensitivities calculated by the OSRS method using a 100,000 path sample and by the FD method using a 10,000,000 path sample

References

[1] Ahrens, J.H. and Dieter, U. (1974). *Computer methods for sampling from gamma, beta, Poisson and Binomial distribution*. Computing, Vol. 12, 223 - 246.

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

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

[4] Box, G. E. P., and Muller, M. E. (1958). *A Note on the Generation of Random Normal Deviates*. The Annals of Mathematical Statistics, Vol. 29(2), 610-611

[5] Chan, J. H., and Joshi, M. S. (2012). *Optimal Limit Methods for Computing Sensitivities of Discontinuous Integrals Including Triggerable Derivative Securities*. Available at SSRN 2011690.

[6] Cheng, R.C.H., and Feast, G.M. (1980) *Gamma variate generators with increased shape parameter range*. Communications of the ACM, Vol. 23, 389-394.

[7] Christianson, B. (1992). *Automatic Hessians by reverse accumulation*. Journal of Numerical Analysis, Vol.12, 135-150.

[8] Devroye, L. (1986). *Non-Uniform Random Variate Generation*. New York: Springer-Verlag.

[9] Fishman, G.S. (1996) *Monte Carlo. Concepts, Algorithms, and Applications*. New York: Springer

[10] Giles, M. (2008). Vibrato Monte Carlo sensitivities, *In Monte Carlo and Quasi-Monte Carlo Methods 2008*. Springer, New York.

[11] Glasserman, P. (1991). *Structural conditions for perturbation analysis derivative estimation: Finite-time performance indices*. Operations Research, Vol. 39(5), 724-738.

[12] Glasserman, P. (1992), *Smoothing complements and randomized score functions,* Annals of Operations Research, 39, 41–67

[13] Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer, New York.

[14] Glasserman, P., and Liu, Zongjian.(2011).*Estimating Greeks in simulating Lévy-driven models*. The Journal of Computational Finance, Vol.14, 3-56.

[15] Giles, M. and Glasserman, P. (2006). *Smoking adjoints: fast Monte Carlo Greeks*. Risk, January, 92-96.

[16] Griewank, A., and Walter, B. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. 2nd edition. SIAM, Philadelphia.

[17] Heidergott, B., Vazquez-Abad, F.J., Pflug, G.Ch. and Farenhorst-Yuan, T. (2010). Gradient estimation for discrete-event systems by measure-valued differentiation, *ACM Transactions on Modeling and Computer Simulation (TOMACS)* , Vol. 20 (1), Association for Computing Machinery

[18] Ho, Y.C., and Cao, X. R. (1983). *Perturbation analysis and optimization of queueing networks*, Journal of Optimization Theory and Applications, Vol. 40, 559-582.

[19] Hong, L. J., and Liu, G. (2011). Kernel estimation of the Greeks for options with discontinuous payoffs, *Operations Research,* 59, 96–108.

[20] Joshi, M. S., Yang, C. (2011) . *Algorithmic Hessians and the Fast Computation of Cross-Gamma Risk*, IIE Transactions, vol. 43 (12), 878 - 892.

[21] Joshi, M. S., and Zhu, D. (2014). *Optimal Partial Proxy Method for Computing Gammas of Financial Products with Discontinuous and Angular Payoff*. Available at SSRN 2431580.

[22] Kinderman, A. J., and Monahan, J. F. (1977). *Computer Generation of Random Variates Using Ratio of Uniform Deviates*. ACM Transaction in Mathematical Software, Vol. 3, 257-260.

[23] Lewis, P. A. W. (1972). *Recent Results in Statistical Analysis of Univeriate Point Processes*. Statistical Point Processes. New York, John Wiley & sons, 1-54.

[24] Lewis, P. A. W., and Shedler, G. S. (1979). *Simulation of nonhomogenous Poisson processes with degree-two exponential polynomial rate function*. Operations Research, Vol. 27(5):1026-1039.

[25] Lyuu, Y.D., and Teng, H. W. (2008), Efficient and unbiased Greeks of rainbow and path-dependent options using importance sampling, *Finance and Stochastics,* 15, Issue 1 (2011), 141–181

[26] Madan, D. B., and Seneta, E. (1990). *The Variance-Gamma (V.G.) model for share market returns*. Journal of Business, Vol. 63, 511-524.

[27] Madan, D., Carr, P., and Chang, E. (1998). *The variance gamma process and option pricing*. European Finance Review, Vol. 2, 79105

[28] Michael, J.R., Schucany, W. R., and Haas, R. W.(1976).*Generating Random Variates Using Transformations with Multiple Roots*. The American Statistician, Vol. 30(2), 88-90

[29] Neumann, J. V. (1951). *Various techniques used in connection with random digits. Monte Carlo methods*, Nat. Bureau Standards, Vol.12, 3638.

[30] Ortega, A. J., and Rheinboldt, W. C. (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, NY.

[31] Rubinstein, R. (1992). *Sensitivity analysis of discrete event systems by the score function method*. Annals of Operations Research, Vol. 39, 229-250.

[32] Ross, R. (2006). *Simulation*. Elsevier, Burlington, MA.

[33] Sato, K. (1999). *Lévy Processes and Infinitely Divisible Distributions*. Cambridge UniversityPress.

[34] Schoutens, W. (2003). *Lévy Processes in Finance: Pricing Financial Derivatives*. Wiley.

[35] Suhasini, A. V. S., Rao, K. S., and Reddy, P. R. S.(2013) *On parallel and series non homogeneous bulk arrival queueing model*. OPSEARCH, Vol. 50(4),521-548.

[36] Suri, R., and Zazanis, M. A. (1988) *Perturbation Analysis Gives Strongly Consistent Sensitivity Estimates for the* $M \setminus G \setminus 1$ *Queue*. Management Science, Vol. 34 (1), 39-64

[37] Tanizaki H (2008). *A Simple Gamma Random Number Generator for Arbitrary Shape Parameters*. Economics Bulletin, Vol. 3(7), 110.

[38] Y. Wang, M.C. Fu, S.I. Marcus, (2012). A new stochastic derivative estimator for discontinuous payoff functions with application to financial derivatives, *Operations Research,* Vol. 60, No. 2, March-April, 447–460

Centre for actuarial studies, Department of Economics, University of Melbourne, VIC3010, Australia