THE UNIVERSITY OF MELBOURNE

Department of Economics

Working Paper Series

Optimal Limit Methods for Computing Sensitivities of Discontinuous Integrals Including Triggerable Derivative Securities

Jiun Hong Chan and Mark Joshi

February 2012

Research Paper Number 1142

ISSN: 0819-2642
ISBN: 978 0 7340 4492 1

Department of Economics
The University of Melbourne
Parkville VIC 3010
www.economics.unimelb.edu.au

OPTIMAL LIMIT METHODS FOR COMPUTING SENSITIVITIES OF DISCONTINUOUS INTEGRALS INCLUDING TRIGGERABLE DERIVATIVE SECURITIES

JIUN HONG CHAN AND MARK JOSHI

Abstract. We introduce a new approach to computing sensitivities of discontinuous integrals. The methodology is generic in that it only requires knowledge of the simulation scheme and the location of the integrand's singularities. The methodology is proven to be optimal in terms of minimizing the variance of the measure changes caused by the elimination of the discontinuities for finite bump sizes. An efficient adjoint implementation of the small bump-size limit is discussed, and the method is shown to be effective for a number of natural examples involving triggerable interest rate derivative securities.

1. Introduction

Many problems in finance and management reduce to the evaluation of a high-dimensional integral. It is often the case that we do not want just its value but also the sensitivities to input parameters. This means that we must develop techniques to compute these sensitivities for integrals approximated by Monte Carlo simulation. Whilst there has been much progress on this problem, there are still unresolved issues. This is particularly the case when the integrand is a discontinuous function of a random variable, and doubly so when the random variable's density is singular. Here we present a new approach which is generic in that little handcrafting is required for particular problems whilst it is in a sense optimal for this amount of genericity. Whilst we focus on financial applications in this paper, we believe that there is potential for it to be applied to a wide range of problems whereever sensitivities of a discontinuous integral are required.

Typically, we write our integral in the form

$$\int g(x, \theta)\Phi(x, \phi)dx,$$

with g a pay-off and Φ a density, and we then wish to compute a sensitivity to either θ or ϕ. In the former case, we rely on the smoothness of g and the method is called the *pathwise* method or the infinitesimal perturbation method (IPA). We essentially need g to be differentiable almost surely and Lipschitz continuous everywhere. For the second case, it is Φ that is differentiated. We typically then multiply and divide by Φ so we still have a density. This results in the integrand being multipled by $(\log \Phi)'$. This method is called the likelihood ratio method and it is the smoothness of Φ that is important. These two methods were introduced to derivatives pricing by Broadie and Glasserman (1996). The unity of the two methods is discussed as far back as L'Ecuyer (1990).

Date: February 27, 2012.
Key words and phrases. Price Sensitivities, Monte-Carlo Greeks, Partial Proxy Simulation Scheme, Minimal Partial Proxy Simulation Scheme, Optimal Partial Proxy Simulation Scheme, Discontinuous Pay-offs, Digital Options, Target Redemption Notes, LIBOR Market Model.

One often change coordinates before differentiating to shift dependence from Φ to g or vice-versa. However, when both g and Φ are singular, neither of these methods applies.

A natural example of a case where g has a jump discontinuity and Φ has a singular density is the case of a triggerable interest rate product priced using the low-factor LIBOR market model. Whilst Greeks that are tangent to the density's support can be computed using the likelihood ratio method, general Greeks are problematic. Even when the likelihood ratio method does apply it can lead to large variances; this is particularly the case for volatility sensitivities of products close to expiry. Other approaches which have been suggested are the vibrato method of Giles (2008) and Malliavin techniques, eg Benhamou (2003). These appear to suffer the same limitations as the likelihood ratio method, however. Another variant of such approaches is to use measure-valued differentation. It is the measure itself that is differentiated and an expression as a difference of two integrals is then obtained. See Heidergott et al (2010), Heidergott and Vasquez-Abad (2006), Volk-Makarewicz et al (2008), Heidergott and Leahu (2010), Vasquez-Abad and Heidergott (2008).

The fundamental problem with applying pathwise methods to discontinuous pay-offs is that the distributional derivative contains a delta distribution supported on the hypersurface of singularity. (See, for example, Friedlander–Joshi (1999).) This delta distribution must either be removed via a regularization technique or integrated explicitly. Whilst explicit integration works, it requires a great deal of handcrafting: Joshi and Kainth (2003), Rott and Fries (2005), and Brace (2007). This method is sometimes called the weak derivative (or WD) method. An alternate approach which reduces handcrafting, Hong and Liu (2008), Lyuu and Teng (2008), is to approximate the delta distribution by a Gaussian supported in a small neighbourhood of its support. This results in an order of convergence of less than 1/2, however.

Since pathwise methods tend to lead to lower variances, and likelihood ratios can be used to cope with singularities of the pay-off, one solution is to use a mix of the two. We differentiate the pay-off where possible and where it is not, we use a likelihood ratio. This is the essence of *proxy methods*. Here bumped paths are simulated using schemes that are designed so that the bump does not cause the crossing of a singularity. Instead, some constraints are imposed and the bumped scheme is modified to leave these invariant. As long as the discontinuity arises from a function of the constrained quantity, jumps are eliminated. Fries and Joshi (2008) and Chan and Joshi (2011) developed such schemes and they have the feature that the finite difference estimate of the derivative does not blow up for small bumps. The essential difference between the two schemes is that the Chan–Joshi method, known as *minimal partial proxy*, chooses the measure change to minimize the variance of the likelihood ratio amongst all Gaussian changes of density that do not cause crossing of the discontinuity, whereas the Fries–Joshi approach is to use the likelihood ratio along the normal line to the proxy constraint's level sets.

Since the proxy methods lead to stable behaviour for small bumps, it is possible to analyze the behaviour as the bump size goes to zero and derive a limit method. This is in similar vein to how the pathwise method is related to finite differencing in the continuous case. These limit proxy methods were studied in Chan and Joshi (2012) and shown to be effective. However, the solution there is not satisfactory for a number of reasons.

- The limit of the Fries–Joshi (2008) scheme leads to high variances for short-dated volatility sensitivities.
- The minimal partial proxy scheme is minimal amongst Gaussian schemes but need not be amongst all schemes.
- The minimal partial proxy scheme requires there to be only one discontinuity per step and so cannot always be used. For example, a double digital option is not within its purview.
- The limit version of the minimal partial proxy scheme was not derived in the case that the discontinuities were determined by a non-linear function.
- In some cases, where the discontinuity was mild, the previous schemes led to higher variance than naive bumping.

Here we present a new limit proxy scheme which has none of these defects. We call it the limit optimal partial proxy scheme. In particular, we show that the variance of the likelihood ratio is minimal amongst all measure changes and that it is therefore in a certain sense optimal. It also works for functions with multiple discontinuities per step, and we show how to adapt the method so that it works with non-linear functions.

One important feature of the method presented here is that it is largely pay-off independent. It is the choice of the proxy constraint function that determines the numerical scheme not the pay-off. We can think of a general pay-off that is a piecewise constant function of the proxy constraint times a general function. Ensuring that the proxy constraint does not pass through jump levels of the discontinuous part is sufficient. When we speak of optimality, we mean that it minimizes the variance of the likelihood ratio weight amongst all proxy schemes that have this property. We show that for some natural examples a substantial reduction is achieved and that it is robust across a wide range of cases. Clearly, further variance reduction could be achieved by further analyzing the interaction with the pay-off. However, that would appear to be too pay-off specific to be generally useful.

For the pathwise method in the continuous case, there has been much interest in adjoint methods following Giles and Glasserman (2006); they applied automatic differentiation techniques to computing Greeks in the LIBOR market model. In particular, they showed that all the first order Greeks could be computed with the same computational order as the price. We show here that the same holds true for the limit optimal proxy scheme and we are thus able not just to achieve low variances but also a high computational speed.

This paper is organized as follows: In section 2, we present a simple example to illustrate the different between the existing proxy simulation methods and our new approach. We outline the numerical scheme and the notations used in this paper in section 3. Section 4 shows the detailed construction of the optimal partial proxy simulation scheme. In section 5, we demonstrate that, under the optimal partial proxy simulation scheme, the order of differentiation (i.e. limit) and expectation (i.e. integration) can be interchanged even when computing pricing sensitivities for financial products with discontinuous pay-offs and using this result, we develop the pathwise OPP method. In section 6, we present our numerical test specifications and numerical results. Lastly, we conclude in section 7.

2. A Simple Example

To illustrate the differences between existing proxy methods and our proposed method, we consider computing the delta of a digital call option under the Black-Scholes model using Monte-Carlo simulations. Assuming zero interest rates, an exact weak solution of the Black-Scholes asset price process at time T is given by

$$S_T = S_0 \exp\left(-0.5\sigma^2 T + \sigma\sqrt{T}Z\right), \tag{2.1}$$

where

- S_0 is the initial asset price,
- σ is the volatility of the log of asset price process,
- Z is a standard normal random variable.

The price of a digital call option maturing at time T with a strike of K is

$$\mathbb{E}\left[g(S_T)\right], \tag{2.2}$$

and the delta is defined to be the price sensitivity with respect to (w.r.t.) S_0, i.e.

$$\frac{\partial}{\partial S_0}\mathbb{E}\left[g(S_T)\right], \tag{2.3}$$

where

$$g(S_T) = \begin{cases} 1 & \text{for } S_T \geq K, \\ 0 & \text{for } S_T < K. \end{cases} \tag{2.4}$$

Here, our interest lies in estimating the delta using Monte-Carlo simulations. While pricing using Monte-Carlo simulations is straight-forward, computing the delta can be rather challenging due to the discontinuous pay-off.

2.1. **Finite Difference Approaches.**

2.1.1. *Naive Bump-and-Revalue Method.* One possible way to estimate the delta is by applying finite differences directly to the simulated prices. This approach is also known as the *bump-and-revalue* method. In order to emphasize the dependency on the perturbation size, we rewrite equation (2.1) as follows

$$S_T(x) = S_0(x) \exp\left(-0.5\sigma^2 T + \sigma\sqrt{T}Z\right), \tag{2.5}$$

where $S_0(x) = S_0 + x$ and $x \in \mathbb{R}$. Under the bump-and-revalue method, the delta is estimated using

$$\frac{\tilde{\mathbb{E}}\left[g(S_T(h))\right] - \tilde{\mathbb{E}}\left[g(S_T(0))\right]}{h} \tag{2.6}$$

where h is a small positive constant and $\tilde{\mathbb{E}}$ represents the expectation approximated via a Monte-Carlo simulation. In general, if $S_T(h)$ and $S_T(0)$ are highly correlated, the variance of the delta estimator can be reduced. Hence, a common set of Z is usually used when sampling $S_T(h)$ and $S_T(0)$.

This method, however, does not work well for products with discontinuous pay-offs. This is because the expression in equation (2.6) can be written as

$$\tilde{\mathbb{E}}\left[\frac{g(S_T(h)) - g(S_T(0))}{h}\right] \tag{2.7}$$

(due to the linearity property of expectation) and we are effectively applying finite difference on a path-by-path basis. Observe that, for most of the paths, the digital option has a zero pathwise sensitivity as both $S_T(h)$ and $S_T(0)$ will either finish above or below the strike. However, occasionally it has a pathwise sensitivity of $1/h$ when both $S_T(h)$ and $S_T(0)$ end up on the different side of the strike (i.e. pathwise discontinuity). Therefore, the Monte-Carlo Greeks obtained using the bump-and-revalue method can have high variance particularly as $h \to 0$.

2.1.2. *Partial Proxy Simulation Scheme.* Fries and Joshi (2008) used a 'smart' finite difference approach to compute Greeks for financial products with discontinuous pay-offs. They defined a new scheme, the *partial proxy simulation scheme*, P, to sample the underlying process. Using this scheme, measure changes (importance samplings) are performed at each simulation time step to fix quantities that give rise to pathwise discontinuities. Hence, pathwise discontinuities are eliminated.

In our example, instead of using equation (2.5), under the partial proxy approach, the asset price process is sampled using

$$S_T^{\mathrm{P}}(x) = S_0(x) \exp\left(-0.5\sigma^2 T + \sigma\sqrt{T}(Z - \nu^{\mathrm{P}}(Z, x))\right), \tag{2.8}$$

with $\nu^{\mathrm{P}}(Z, 0) = 0$. The superscript P indicates that the asset price is generated using the partial proxy simulation scheme. Applying this new scheme can be loosely viewed as sampling the underlying asset price process in a new measure where Z has a drift of $-\nu^{\mathrm{P}}(Z, x)$. Note that when $x = 0$, the scheme of (2.8) reduces to that of (2.5).

The key to the Fries and Joshi approach is that, at each path, the perturbed and unperturbed asset price paths, i.e. $S_T^{\mathrm{P}}(h)$ and $S_T^{\mathrm{P}}(0)$, are sampled simultaneously with the solution of $\nu^{\mathrm{P}}(Z, h)$ determined so that

$$S_T^{\mathrm{P}}(h) = S_T^{\mathrm{P}}(0), \tag{2.9}$$

and thus giving[1]

$$\nu^{\mathrm{P}}(Z, h) = \frac{\log\left(S_0^{\mathrm{P}}(h)\right) - \log\left(S_0^{\mathrm{P}}(0)\right)}{\sigma\sqrt{T}}. \tag{2.10}$$

This ensures that pathwise discontinuities are eliminated as both $S_T^{\mathrm{P}}(h)$ and $S_T^{\mathrm{P}}(0)$ always finish on the same side of the discontinuity. As importance samplings are performed when sampling $S_T^{\mathrm{P}}(h)$, the pay-off at each path must be scaled by the Monte-Carlo weight,

$$w^{\mathrm{P}}(h) = \left(1 - \frac{\partial \nu^{\mathrm{P}}(Z, h)}{\partial Z}\right) \exp\left(Z \cdot \nu^{\mathrm{P}}(Z, h) - 0.5 \cdot \nu^{\mathrm{P}}(Z, h)^2\right), \tag{2.11}$$

[1]In this example, the solution of ν does not depend on Z. However, such a dependency is important when it comes to computing vegas.

so that the Monte-Carlo expectation of the weighted pay-off is an unbiased estimator of the (perturbed) price, i.e.

$$\mathbb{E}\left[w^{\mathrm{P}}(h)g(S_T^{\mathrm{P}}(h))\right] = \mathbb{E}\left[g(S_T(h))\right]. \tag{2.12}$$

The Monte-Carlo weight in equation (2.11) can be easily derived as both Z and $Z - \nu(Z, h)$ are defined on a same probability space and we have

$$\int_{\mathbb{R}} q(Z)\phi(Z)dZ = \int_{\mathbb{R}} \left(1 - \frac{\partial \nu(Z,h)}{\partial Z}\right) \frac{\phi(Z - \nu(Z,h))}{\phi(Z)} q(Z - \nu(Z,h))\phi(Z)dZ.$$

where $q : \mathbb{R} \to \mathbb{R}$ and ϕ is the standard normal density function. Hence, under the partial proxy approach, the delta of digital call options can be estimated using

$$\tilde{\mathbb{E}}\left[\frac{w^{\mathrm{P}}(h)g(S_T^{\mathrm{P}}(h)) - w^{\mathrm{P}}(0)g(S_T^{\mathrm{P}}(0))}{h}\right] \tag{2.13}$$

where, by construction, $w^{\mathrm{P}}(0) = 1$ for all paths.

Note that, in a one-dimensional setting, the partial proxy simulation scheme is equivalent to the full proxy simulation scheme of Fries and Kampen (2007). Here, we have not fully illustrated the strength of the partial proxy simulation scheme where it can be applied to factor-reduced models when other methods fail. We refer the interested reader to Fries and Joshi (2007) and Chan and Joshi (2012).

2.1.3. *Minimal Partial Proxy Simulation Scheme.* The minimal partial proxy (MPP) simulation scheme was introduced by Chan and Joshi (2011). This scheme is similar to the PP scheme and it differs in how the solution of $\nu(Z, h)$ is determined. Chan and Joshi made the weaker constraint that both $S_T(h)$ and $S_T(0)$ only had to end up on the same side of the trigger. This gave them an extra degree of freedom and they were therefore able to select the measure change which minimizes the variance of the Monte-Carlo weight within the class of Gaussian measure changes.

Under the MPP approach, the asset price process in this example is sampled using

$$S_T^{\mathrm{M}}(x) = S_0(x) \exp\left(-0.5\sigma^2 T + \sigma\sqrt{T}(Z - \nu^{\mathrm{M}}(Z, x))\right), \tag{2.14}$$

where

$$\nu^{\mathrm{M}}(Z, x) = (1 - \alpha^{\mathrm{M}}(x))Z - \beta^{\mathrm{M}}(x) \tag{2.15}$$

with $\alpha^{\mathrm{M}}(0) = 1$ and $\beta^{\mathrm{M}}(0) = 0$ giving $\nu^{\mathrm{M}}(Z, 0) = 0$. Here, we have the superscript M indicating the MPP simulation scheme.

When computing price sensitivities, the solutions of $\alpha^{\mathrm{M}}(h)$ and $\beta^{\mathrm{M}}(h)$ are determined such that the following conditions hold:

Condition (1): Eliminate pathwise discontinuities i.e.

$$S_T^{\mathrm{M}}(h) \geq K \iff S_T^{\mathrm{M}}(0) \geq K \tag{2.16}$$

Condition (2): Minimize the variance of the Monte-Carlo weight where the Monte-Carlo weight is given by

$$w^{\mathrm{M}}(h) = \alpha^{\mathrm{M}}(h) \exp\left(Z \cdot \nu^{\mathrm{M}}(Z,h) - 0.5 \cdot \nu^{\mathrm{M}}(Z,h)^2\right). \tag{2.17}$$

Substituting equation (2.14) into condition (1) and rearranging gives

$$\alpha^{\mathrm{M}}(h)Z + \beta^{\mathrm{M}}(h) \geq Z^*(h) \iff Z \geq Z^*(0) \tag{2.18}$$

where

$$Z^*(x) = \frac{\log K - \log S_0(x) + 0.5\sigma^2 T}{\sigma\sqrt{T}}$$

Hence, when $Z = Z^*(0)$, we must also have

$$\alpha^{\mathrm{M}}(h)Z^*(0) + \beta^{\mathrm{M}}(h) = Z^*(h) \tag{2.19}$$

to prevent pathwise discontinuities. The solutions of $\alpha^{\mathrm{M}}(h)$ and $\beta^{\mathrm{M}}(h)$ must therefore satisfy equation (2.19). Since there are two variables and one equation, the additional degree of freedom can be used to minimize the variance of the Monte-Carlo weight within this class. It turns out that, for conditions (1) and (2) to hold, the solution of $\alpha^{\mathrm{M}}(h)$ must be greater than $1/\sqrt{2}$ and must satisfy

$$A\alpha^{\mathrm{M}}(h)^4 + B\alpha^{\mathrm{M}}(h)^3 + C\alpha^{\mathrm{M}}(h)^2 + D\alpha^{\mathrm{M}}(h) + E = 0$$

where

$$\begin{aligned} A &= 2 \\ B &= 2 \cdot Z^*(h) \cdot Z^*(0) \\ C &= -(3 + 2 \cdot Z^*(h)^2 + Z^*(0)^2) \\ D &= Z^*(h) \cdot Z^*(0) \\ E &= 1 \end{aligned} \tag{2.20}$$

Solving for $\alpha^{\mathrm{M}}(h)$ is straight-forward as closed-form solutions exist for the roots of a 4^{th} order polynomial while the solution for $\beta^{\mathrm{M}}(h)$ is given by

$$\beta^{\mathrm{M}}(h) = Z^*(h) - \alpha^{\mathrm{M}}(h)Z^*(0).$$

Hence, under the MPP approach, the delta of digital call options can be estimated using

$$\tilde{\mathbb{E}}\left[\frac{w^{\mathrm{M}}(h)g(S_T^{\mathrm{M}}(h)) - w^{\mathrm{M}}(0)g(S_T^{\mathrm{M}}(0))}{h}\right]. \tag{2.21}$$

While the MPP approach generally performs better than the PP approach, it is not as generic as the PP approach. The MPP approach can only be applied to cases with one discontinuity per time step. However, in a practical financial setting, it is common for pay-off functions to have multiple discontinuities, for example, a double digital caplet. Also, the MPP computational cost can be very high when the analytical solution of Z^* does not exist as numerical methods are usually required to solve for Z^*.

2.1.4. *Optimal Partial Proxy Simulation Scheme.* Here, we present a simple application of the optimal partial proxy (OPP) simulation scheme before formalizing the underlying theoretical framework of this scheme in the next section. Observe that, when sampling Z, we typically generate a standard uniform random variable U, and Z can then be obtained via the inverse transformation method. Unlike the PP and MPP approaches where measure changes transform Z, the OPP simulation scheme works directly with U.

We stick with the Black–Scholes digital call option example. With the OPP approach, the asset price process is sampled using

$$S_T^{\mathrm{L}}(x) = S_0(x) \exp\left(-0.5\sigma^2 T + \sigma\sqrt{T}\Psi^{-1}\left(f^{\mathrm{L}}(U,x)\right)\right), \tag{2.22}$$

where Ψ^{-1} is the inverse of the standard normal distribution function and $f^{\mathrm{L}}(U,x) : [0,1) \times \mathbb{R} \to [0,1)$ is a Lipschitz continuous function of U and x with $f^{\mathrm{L}}(U,0) = U$. We make the additional assumption that $f^L(U,x)$ is a continuously differentiable function of U except at a finite number of points $\lambda_j(x)$ which are smooth[1] functions of x.

The function f^{L} is constructed so that pathwise discontinuities are eliminated. Similar to the condition (1) in the MPP approach, pathwise discontinuities are eliminated when

$$f^{\mathrm{L}}(U,h) \geq U^*(h) \iff f^{\mathrm{L}}(U,0) = U \geq U^*(0) \tag{2.23}$$

where

$$U^*(x) = \Psi\left(\frac{\log K - \log S_0(x) + 0.5\sigma^2 T}{\sigma\sqrt{T}}\right) = \Psi(Z^*(x)).$$

Under the OPP approach, this requirement is satisfied by setting

$$f^{\mathrm{L}}(U,h) = \begin{cases} \frac{1-U^*(h)}{1-U^*(0)} \cdot (U - U^*(0)) + U^*(h) & \text{for } U \geq U^*(0), \\ \frac{U^*(h)}{U^*(0)} \cdot U & \text{for } U < U^*(0). \end{cases} \tag{2.24}$$

with the Monte-Carlo weight given by

$$w^{\mathrm{L}}(h) = \frac{\partial f^{\mathrm{L}}(U,h)}{\partial U} = \begin{cases} \frac{1-U^*(h)}{1-U^*(0)} & \text{for } U \geq U^*(0), \\ \frac{U^*(h)}{U^*(0)} & \text{for } U < U^*(0). \end{cases} \tag{2.25}$$

Similar to the PP and MPP approaches, under the OPP approach, the delta of digital call options can be estimated using

$$\tilde{\mathbb{E}}\left[\frac{w^{\mathrm{L}}(h)g(S_T^{\mathrm{L}}(h)) - w^{\mathrm{L}}(0)g(S_T^{\mathrm{L}}(0))}{h}\right]. \tag{2.26}$$

Our choice of f^{L} ensures that the variance of the Monte-Carlo weight is minimized across all possible choices of f^{L} (see Section 4.2 for the proof). In contrast, the MPP approach only minimizes

[1]We shall take *smooth* to mean infinitely differentiable in this paper. However, we note that C^1 or C^2 would suffice for our results.

the variance subject to ν^M taking the following form

$$\nu^{\mathrm{M}}(Z,x) = (1-\alpha^{\mathrm{M}}(x))Z - \beta^{\mathrm{M}}(x), \tag{2.27}$$

which is equivalent to setting

$$f^{\mathrm{L}}(U,x) = \Psi(\alpha^{\mathrm{M}}(x)\Psi^{-1}(U) + \beta^{\mathrm{M}}(x)).$$

2.2. **Pathwise Approaches.**

2.2.1. *Standard Pathwise Method.* The pathwise method was introduced to derivatives pricing by Broadie and Glasserman (1996). This approach requires the pay-off function to be Lipschitz continuous. Provided that the pay-off function $\bar{g}$ satisfies the Lipschitz condition, the order of the differentiation (i.e. limit) and the expectation can be interchanged as follows:

$$\begin{aligned}\frac{\partial}{\partial S_0}\mathbb{E}\left[\bar{g}(S_T))\right] &= \lim_{h\to 0}\left(\frac{\mathbb{E}\left[\bar{g}(S_T(h))\right] - \mathbb{E}\left[\bar{g}(S_T(0))\right]}{h}\right)\\ &= \mathbb{E}\left[\lim_{h\to 0}\frac{\bar{g}(S_T(h)) - \bar{g}(S_T(0))}{h}\right]\\ &= \mathbb{E}\left[\frac{\partial\bar{g}(S_T)}{\partial S_T}\frac{\partial S_T}{\partial S_0}\right]\end{aligned} \tag{2.28}$$

and, hence, price sensitivities can be obtained by differentiating each simulated outcome w.r.t. the parameter of interest. Note that by taking the limit as h approaches zero, the bump-and-revalue method converges to the pathwise method provided that $\bar{g}$ is Lipschitz continuous. The pathwise method, if applicable, can reduce the computational cost substantially as all sensitivities can be evaluated simultaneously in one simulation. However, in our example, the pathwise method is not applicable as digital call options have discontinuous pay-offs. The failure to satisfy the Lipschitz condition is reflected by the fact that the standard error of the bump-and-revalue method blows up as h goes zero.

2.2.2. *Pathwise PP method.* Chan and Joshi (2012) showed that the construction of the PP simulation scheme permits the interchange of differentiation (i.e. limit) and expectation operators even when computing price sensitivities for products with discontinuous-pay-offs. Using this result and taking the limit as h goes zero, Chan and Joshi developed the pathwise PP method where price sensitivities can be evaluated as follow:

$$\lim_{h\to 0}\tilde{\mathbb{E}}\left[\frac{w^{\mathrm{P}}(h)g(S_T^{\mathrm{P}}(h)) - w^{\mathrm{P}}(0)g(S_T^{\mathrm{P}}(0))}{h}\right] = \tilde{\mathbb{E}}\left[g(S_T^{\mathrm{P}}(0))\frac{\partial w^{\mathrm{P}}(0)}{\partial S_0(0)} + w^{\mathrm{P}}(0)\frac{\partial g(S_T^{\mathrm{P}}(0))}{\partial S_0(0)}\right] \tag{2.29}$$

In particular, the delta estimate of digital call options can be calculated using

$$\tilde{\mathbb{E}}\left[g(S_T)\frac{Z}{S_0\sigma\sqrt{T}}\right]$$

and, for completeness, we also present the estimate for vega which is given by

$$\tilde{\mathbb{E}}\left[g(S_T)\left(\frac{Z^2-1}{\sigma} - Z\sqrt{T}\right)\right]$$

Observe that, we have $S_T^{\mathrm{P}}(0) = S_T(0) = S_T$. Therefore, when using the pathwise PP approach, the underlying asset price process can instead be sampled using the original numerical scheme.

As discussed by Chan and Joshi (2012), the pathwise PP method is a hybrid approach between the standard pathwise method and the likelihood ratio method (LRM) where the standard pathwise method is used to compute the sensitivities in the non-discontinuous directions while the sensitivities in the discontinuous directions are evaluated using the likelihood ratio method. In fact, in the one-dimensional setting considered above, the pathwise PP method is the LRM.

2.2.3. *Pathwise MPP method.* Chan and Joshi (2012) also developed the pathwise MPP method by taking the limit as h approaches zero and interchanging the differentiation and the expectation. In particular, under this approach, the delta estimate for a digital call option is given by

$$\tilde{\mathbb{E}}\left[g(S_T)\left(\frac{2Z + Z^2 Z^*(0) - Z^*(0)}{2 + Z^*(0)^2}\right)\frac{1}{S_0\sigma\sqrt{T}}\right]$$

while the vega estimate is given by

$$\tilde{\mathbb{E}}\left[g(S_T)\left(\frac{2Z + Z^2 Z^*(0) - Z^*(0)}{2 + Z^*(0)^2}\right)\left(\frac{Z^*(0)}{\sigma} - \sqrt{T}\right)\right]$$

Again, this method is a hybrid approach. For the discontinuous directions, sensitivities are evaluated by selecting a smoothly varying mix of likelihood ratio method and pathwise method such that the impact of the likelihood ratio method is minimized amongst Gaussian measure changes.

2.2.4. *Pathwise OPP method.* Building on the work of Chan and Joshi (2012), it is straight-forward to develop a similar pathwise method for the OPP simulation scheme. Under the pathwise OPP method, price sensitivities w.r.t. S_0 can be evaluated as follow

$$\tilde{\mathbb{E}}\left[g(S_T^{\mathrm{L}}(0))\frac{\partial w^{\mathrm{L}}(0)}{\partial S_0(0)} + w^{\mathrm{L}}(0)\frac{\partial g(S_T^{\mathrm{L}}(0))}{\partial S_0(0)}\right] \tag{2.30}$$

where

$$\frac{\partial w^{\mathrm{L}}(0)}{\partial S_0(0)} = \begin{cases} \frac{-\phi(Z^*(0))}{1-U^*(0)}\frac{-1}{S_0(0)\sigma\sqrt{T}} & \text{for } U \geq U^*(0), \\ \frac{\phi(Z^*(0))}{U^*(0)}\frac{-1}{S_0(0)\sigma\sqrt{T}} & \text{for } U < U^*(0). \end{cases} \tag{2.31}$$

In particular, the delta estimate of a digital call option is given by

$$\tilde{\mathbb{E}}\left[\left(\frac{\phi(Z^*(0))}{(1-U^*(0))}\frac{1}{S_0\sigma\sqrt{T}}\right)\mathbb{I}_{[U>U^*(0)]}\right]$$

while the vega estimate is given by

$$\tilde{\mathbb{E}}\left[\left(\frac{-\phi(Z^*(0))\sqrt{T}}{(1-U^*(0))}\left(1 - \frac{Z^*(0)}{\sigma\sqrt{T}}\right)\right)\mathbb{I}_{[U>U^*(0)]}\right]$$

This method is also a hybrid approach where sensitivities in the discontinuous directions are evaluated using a combination of the standard pathwise method and the LRM.

	Pathwise PP				Pathwise MPP				Pathwise OPP			
	deltas		vegas		deltas		vegas		deltas		vegas	
Maturity	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
0.01	8.0%	11.7%	-2.0%	195.1%	8.0%	11.8%	-2.0%	3.0%	8.0%	8.1%	-2.0%	2.0%
0.10	2.5%	3.7%	-6.3%	184.8%	2.5%	3.9%	-6.3%	9.6%	2.5%	2.7%	-6.3%	6.7%
0.25	1.6%	2.3%	-9.9%	176.5%	1.6%	2.5%	-9.9%	15.6%	1.6%	1.7%	-9.9%	10.9%
0.50	1.1%	1.7%	-13.9%	167.6%	1.1%	1.8%	-13.9%	22.5%	1.1%	1.3%	-13.9%	16.0%
1.00	0.8%	1.2%	-19.3%	156.2%	0.8%	1.3%	-19.3%	32.7%	0.8%	0.9%	-19.3%	23.6%

Table 2.1. Digital Calls

2.3. **Numerical Results.** Here, we present the deltas and vegas for digital call options with maturities, $T = 0.01, 0.1, 0.25, 0.5, 1.0,$. We set $S_0 = 100$, $K = 100$ and $\sigma = 0.5$. As we are working in a one-dimensional state space, the deltas and the vegas can be evaluated via a direct integration instead of using a Monte-Carlo simulation. The results are summarized in Table 2.1. As we see from the table, the means for deltas and vegas are the same for all three methods. However the pathwise OPP method gives the lowest standard deviations. In particular, the pathwise OPP method is much better than the pathwise MPP and is significantly better than the pathwise PP method which is equivalent to the LRM in this one-dimensional case.

3. Numerical Scheme and Product Specifications

3.1. **Numerical Scheme Specifications.** In this and the following sections, we formalize the underlying theoretical framework of the OPP simulation scheme. Suppose that the underlying quantities of a financial contract at time t_i is denoted by $K_i(\theta) = (K_{i,1}(\theta), K_{i,2}(\theta), \ldots, K_{i,n}(\theta))$ where $\theta \in \mathbb{R}^k$ is a vector of initial inputs. Our set-up is that the vectors K_i are determined recursively by a numerical scheme. In particular, we take

$$K_i(\theta) = F_i(K_{i-1}(\theta), \theta, U_i) \tag{3.1}$$

where F_i is a smooth mapping function and $U_i \in \mathbb{R}^d$ is a vector of standard uniform random variables. Note that, here, we are working in a post-discretization state space and we make no assumption on the SDE of K. Continuous diffusive processes discretized using an Euler or log-Euler scheme generally satisfy the above form.

When computing price sensitivities, we are interested in the rate of change in price when perturbing some base inputs θ_0. Under the bump-and-revalue approach, the rate of change in price can be determined by applying finite difference to the base price and the bumped price obtained using θ_0 and a nearby value, θ_b respectively. Typically, this approach does not work well for financial products with discontinuous pay-offs due to pathwise discontinuities (See, Fries and Joshi (2012)).

3.2. **Product Specifications.** In this paper, we consider a financial product which depends on the underlying quantities K at times $t_1, t_2, \ldots t_m$. In particular, we assume that, at time t_i, the product will generate a cash-flow $g_{i,j}(\theta) \equiv g_{i,j}(K_1(\theta), \ldots, K_i(\theta))$ conditional on particular events $E_{i,j}$ occurring for $j = 0, 1, 2, \ldots, r-1$. We take the event $E_{i,j}$ to be of the form

$$H_{i,j}(\theta) \leq p_i(K_i(\theta)) < H_{i,j+1}(\theta) \tag{3.2}$$

where $H_{i,s}(\theta)$ is a $\mathcal{F}_{t_{i-1}}$-measurable trigger level for $s = 0, 1, 2, \ldots, r$ such that

$$H_{i,0}(\theta) < H_{i,1}(\theta) < \ldots < H_{i,r}(\theta)$$

with $H_{i,0}(\theta) \to -\infty$ and $H_{i,r}(\theta) \to \infty$ while the function $p_i : \mathbb{R}^n \to \mathbb{R}$ represents the quantity which can give rise to pathwise discontinuities at time t_i. We shall call this function the *proxy constraint function*. The conditional cash-flow $g_{i,j}(\theta)$ is $\mathcal{F}_{t_i}$-measureable and it is a continuous smooth function of model inputs. We write the cash-flow generated at time t_i as

$$g_i(\theta) = \sum_{j=0}^{r-1} g_{i,j}(\theta)\mathbb{I}_{[H_{i,j}(\theta) \leq p_i(K_i(\theta)) < H_{i,j+1}(\theta)]}$$

and write the accumulated cash-flow generated from time t_1 to t_m as

$$g(\theta) = \sum_{i=1}^{m} g_i(\theta)$$

where $\mathbb{I}$ is an indicator function.

For instance, suppose that we are working with the LIBOR market model and we wish to compute the price sensitivities of a double digital CMS call option with strike k_1 and k_2 maturing at time t_i. Under this scenario, we will have K_i being the forward rates while $p_i(K_i)$ represents the underlying CMS swap rate with $H_{i,1} = k_1$ and $H_{i,2} = k_2$.

4. Optimal Partial Proxy Simulation Scheme

The optimal partial proxy (OPP) simulation scheme, L, bears some resemblance to the MPP scheme. We observe the uniforms $U_{i,j}$ with $j > 1$ and use them to determine the measure change for $U_{i,1}$ i.e. the measure change for $U_{i,1}$ is determined based on the realization of $U_{i,j}$ for $j > 1$. For convenience, let

$$U_i^{'} = (U_{i,2}, U_{i,3}, \ldots, U_{i,d}).$$

Our scheme, L, takes the following form

$$\begin{aligned}
K_0^{\mathrm{L}}(\theta) &= K_0(\theta) \\
K_i^{\mathrm{L}}(\theta) &= F_i^{\mathrm{L}}(K_{i-1}^{\mathrm{L}}(\theta), \theta, U_i) \\
&= F_i(K_{i-1}^{\mathrm{L}}(\theta), \theta, V_i^{\mathrm{L}}(\theta, U_i))
\end{aligned} \tag{4.1}$$

where

$$V_i^{\mathrm{L}}(\theta, U_i) = (f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i^{'}), U_i^{'})$$

and F_i is the mapping function as defined in equation (3.1). The function $f_i^{\mathrm{L}} : \mathbb{R}_\theta^k \times [0,1]_{U_{i,1}} \to [0,1]$ is a Lipschitz continuous function of θ and U_i and will be smooth as a function of $U_{i,1}$ except at a finite number of values λ_j which vary smoothly with θ and $U_i^{'}$. We also impose the following constraints on f_i^L :

$$\begin{aligned}
f_i^{\mathrm{L}}(\theta, 0, U_i^{'}) &= 0, \\
f_i^{\mathrm{L}}(\theta, 1, U_i^{'}) &= 1,
\end{aligned}$$

and f^{L} is monotone increasing in $U_{i,1}$. We also restrict

$$f_i^{\mathrm{L}}(\theta_0, U_{i,1}, U_i') \quad = \quad U_{i,1}. \tag{4.2}$$

The Monte-Carlo weight across time steps t_{i-1} and t_i is given by

$$w_i^{\mathrm{L}}(\theta) = \frac{\partial f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')}{\partial U_{i,1}}. \tag{4.3}$$

This follows from a change of variables since

$$\begin{aligned} \int_{[0,1)^d} g(U_i)\vartheta(U_i)dU_i &= \int_{[0,1)^d} \left|\det\left(\frac{\partial V_i^{\mathrm{L}}(\theta, U_i)}{\partial U_i}\right)\right| g(V_i^{\mathrm{L}}(\theta, U_i))\vartheta(V_i^{\mathrm{L}}(\theta, U_i))dU_i \\ &= \int_{[0,1)^d} \frac{\partial f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')}{\partial U_{i,1}} g(V_i^{\mathrm{L}}(\theta, U_i))\vartheta(U_i)dU_i \end{aligned} \tag{4.4}$$

where ϑ is the density function of a d-dimensional standard uniform random variable with $\vartheta(.) = 1$ Note that, due to equation (4.2), when our simulation inputs are given by θ_0, the OPP simulation scheme is the same numerical scheme as in (3.1) since we have

$$w_i^L(\theta_0) = 1. \tag{4.5}$$

The crux of the scheme lies in the choice of f^L. We have two objectives. The first is to eliminate pathwise discontinuities, thereby enabling the application of the finite difference method and the possibility of evaluating the limit as the bump-size goes to zero. The second is to minimize the variance of the likelihood ratio weight amongst all such measure changes.

We achieve the first by constructing $f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')$ so that both $p_i(K_i^{\mathrm{L}}(\theta_0))$ and $p_i(K_i^{\mathrm{L}}(\theta))$ always trigger the same events $E_{i,j}$ for all θ i.e.

$$H_{i,j}(\theta_0) \le p_i(K_i^{\mathrm{L}}(\theta_0)) < H_{i,j+1}(\theta_0) \tag{4.6}$$

if and only if

$$H_{i,j}(\theta) \le p_i(K_i^{\mathrm{L}}(\theta)) < H_{i,j+1}(\theta). \tag{4.7}$$

4.1. **Construction of f_i^{L}.** Constructing $f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')$ directly such that pathwise discontinuities are eliminated can be rather difficult as f_i^{L} depends on the realization of $U_{i,1}$ and U_i'. Similar to the MPP approach, here, we assume that we observe the realization of U_i' before $U_{i,1}$ at time t_i. Making such an assumption reduces the complexity in constructing f_i^{L} significantly. In particular, conditional on knowing the realization of U_i', the function $f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')$ effectively becomes a one-dimensional function of $U_{i,1}$. Similarly, this allows us to regard p_i as a function of $U_{i,1}$; the occurrence of event $E_{i,j}$ is then determined by the value of $U_{i,1}$. We can therefore reduce this complicated problem to a one-dimensional problem. In all the following discussions, we assume that the realization of U_i' is already determined and we let $f_i^{\mathrm{L}}(\theta, U_{i,1})$ represent $f_i^{\mathrm{L}}(\theta, U_{i,1}, U_i')$ conditional on knowing U_i'.

We define

$$K_i^*(\theta) \quad = \quad F_i(K_{i-1}^{\mathrm{L}}(\theta), \theta, U_i). \tag{4.8}$$

Suppose that p_i is monotone increasing in $U_{i,1}$. We let $U_{i,1}^{*j}(\theta)$ be the value of $U_{i,1}$ which yields

$$p_i(K_i^*(\theta)) = H_{i,j}(\theta)$$

for $j = 1, \ldots, r-1$. We also set $U_{i,1}^{*0}(\theta) = 0$ and $U_{i,1}^{*r}(\theta) = 1$ for convenience in what follows, as these correspond to the minimum and the maximum values attainable by p_i. Conditional on the information already observed, the event $E_{i,j}$ can now be expressed as

$$U_{i,1}^{*j}(\theta) \leq f^{\mathrm{L}}(\theta, U_{i,1}) < U_{i,1}^{*j+1}(\theta). \tag{4.9}$$

Consequently, the equations (4.6) and (4.7) can be written as follows:

$$U_{i,1}^{*j}(\theta_0) \leq U_{i,1} < U_{i,1}^{*j+1}(\theta_0) \tag{4.10}$$

if and only if

$$U_{i,1}^{*j}(\theta) \leq f_i^{\mathrm{L}}(\theta, U_{i,1}) < U_{i,1}^{*j+1}(\theta) \tag{4.11}$$

as $f_i^{\mathrm{L}}(\theta_0, U_{i,1}) = U_{i,1}$ by construction. Therefore, we can conclude that pathwise discontinuities will be eliminated provided

$$f_i^{\mathrm{L}}(\theta, U_{i,1}^{*j}(\theta_0)) \quad = \quad U_{i,1}^{*j}(\theta) \tag{4.12}$$

$$\int_{U_{i,1}^{*j}(\theta_0)}^{U_{i,1}^{*j+1}(\theta_0)} \frac{\partial f_i^{\mathrm{L}}(\theta, u)}{\partial u} \vartheta(u) du \quad = \quad U_{i,1}^{*j+1}(\theta) - U_{i,1}^{*j}(\theta) \tag{4.13}$$

for $j = 0, 1, \ldots, r-1$.

Our second objective was to minimise the variance of the Monte-Carlo weight which is equivalent to minimising its second moment, as its mean is fixed. So we must minimize

$$\int_0^1 \left(\frac{\partial f_i^{\mathrm{L}}(\theta, u)}{\partial u} \right)^2 du = \sum_{j=0}^{r-1} \left(\int_{U_{i,1}^{*j}(\theta_0)}^{U_{i,1}^{*j+1}(\theta_0)} \left(\frac{\partial f_i^{\mathrm{L}}(\theta, u)}{\partial u} \right)^2 du \right) \tag{4.14}$$

subject to these conditions in equations (4.12) and (4.13) . We will see in Section 4.2, that the minimum is attained by setting

$$\frac{\partial f_i^{\mathrm{L}}(\theta, U_{i,1})}{\partial U_{i,1}} = \frac{U_{i,1}^{*j+1}(\theta) - U_{i,1}^{*j}(\theta)}{U_{i,1}^{*j+1}(\theta_0) - U_{i,1}^{*j}(\theta_0)} \tag{4.15}$$

when $U_{i,1}^{*j}(\theta_0) \leq U_{i,1} < U_{i,1}^{*j+1}(\theta_0)$. We therefore have

$$f_i^{\mathrm{L}}(\theta, U_{i,1}) = \left(\frac{U_{i,1}^{*j+1}(\theta) - U_{i,1}^{*j}(\theta)}{U_{i,1}^{*j+1}(\theta_0) - U_{i,1}^{*j}(\theta_0)} \right) \left(u - U_{i,1}^{*j}(\theta_0) \right) + U_{i,1}^{*j}(\theta) \tag{4.16}$$

when $U_{i,1}^{*j}(\theta_0) \leq U_{i,1} < U_{i,1}^{*j+1}(\theta_0)$ i.e. f_i^{L} is a piecewise linear function of $U_{i,1}$. Hence, under the OPP approach, price sensitivities can be evaluated taking the difference of

$$\mathbb{E}\left[g^{\mathrm{L}}(\theta_b) \prod_{i=1}^{m} w_i^{\mathrm{L}}(\theta_b) \right]$$

and

$$\mathbb{E}\left[g^{\mathrm{L}}(\theta_0)\prod_{i=1}^{m} w_i^{\mathrm{L}}(\theta_0)\right]$$

where

$$g^{\mathrm{L}}(\theta) \equiv g(K_1^{\mathrm{L}}(\theta), K_2^{\mathrm{L}}(\theta), \ldots, K_m^{\mathrm{L}}(\theta))$$

represents the accumulated deflated cash-flows generated by a financial product based on the realization of K under the OPP simulation scheme.

4.2. **Minimality of Measure Change.** In this section, we derive the results required to construct f_i^{L} such that the variance of the Monte-Carlo weight is minimised. We first derive some general results before restricting to our specific case. Given a value t, we wish to find w that minimizes

$$\int_a^b w(u)^2 du$$

subject to

$$\int_a^b w(u) du = t.$$

We first work under the mild condition that g is measurable so that these integrals make sense. Using the Cauchy–Schwartz inequality, we have

$$t = \int_a^b w(u) du, \tag{4.17}$$

$$= \int_a^b w(u).1 du, \tag{4.18}$$

$$\leq \left(\int_a^b w(u)^2 du\right)^{1/2} \left(\int_a^b 1 du\right)^{1/2}, \tag{4.19}$$

$$= (b-a)^{1/2} \left(\int_a^b w(u)^2 du\right)^{1/2}. \tag{4.20}$$

This immediately implies

$$\int_a^b w(u)^2 du \geq t^2/(b-a). \tag{4.21}$$

If we let

$$w(u) = t/(b-a),$$

then this minimum is attained.

Thus a global minimum exists and is attained. It remains to show that the minimum is unique. We employ a calculus of variations argument. We make a stronger but still mild assumption.

Theorem: If w is continuously differentiable on $[a, b]$ except at a finite number of points where it is continuous and w is not constant, and

$$\int_a^b w(u)du = t,$$

then w does not minimize

$$\int_a^b w(u)^2 du$$

within this class of functions.

Proof. If w is not constant then there must be some point $r \in (a, b)$ where $w'(r) \neq 0$ and some $\delta > 0$ such that the sign of $w'(s)$ is constant for $|x - r| < \delta$.

Let ϕ be a bump function which is non-negative everywhere, even, identically one for $|u| < 1/4$, and zero for $|u| > 1/2$. (See for example Hörmander (1983) Theorem 1.41) Let

$$h_{\delta,r}(x) = \sin\left(\frac{x-r}{\delta}\right)\phi\left(\frac{x-r}{\delta}\right).$$

Since ϕ is even and sin is odd, we have

$$\int h_{\delta,r}(x) = 0,$$

and so

$$I_{1,\epsilon} = \int w(u) + \epsilon h_{\delta,r}(u)du = t$$

for any ϵ.

Now

$$\begin{aligned} I_{2,\epsilon} &= \int (w(u) + \epsilon h_{\delta,r}(u))^2 du, \\ &= \int w(u)^2 du + \epsilon 2 \int w(u) h_{\delta,r}(u) du + \epsilon^2 \int h_{\delta,r}(u)^2 du. \end{aligned}$$

Differentiating with respect to ϵ and setting it to zero. We have

$$I'_{2,0} = 2 \int w(u) h_{\delta,r}(u) du.$$

Since h is odd about r, and w has non-zero derivative on its support, this will be non-zero. If w were minimal, 0 would be a critical point and we would get zero. We conclude that w is not minimal as asserted. □

From equation (4.22), we can write our objective function as follows

$$\text{minimise}\left(\sum_{j=0}^{r-1}\left(\int_{U_{i,1}^{*j}(\theta_0)}^{U_{i,1}^{*j+1}(\theta_0)}\left(\frac{\partial f_i^{\mathrm{L}}(\theta,u)}{\partial u}\right)^2 du\right)\right) \tag{4.22}$$

$$= \sum_{j=0}^{r-1}\left(\text{minimise}\left(\int_{U_{i,1}^{*j}(\theta_0)}^{U_{i,1}^{*j+1}(\theta_0)}\left(\frac{\partial f_i^{\mathrm{L}}(\theta,u)}{\partial u}\right)^2 du\right)\right) \tag{4.23}$$

as the only restriction we have on the Monte-Carlo weight is that

$$\int_{U_{i,1}^{*j}(\theta_0)}^{U_{i,1}^{*j+1}(\theta_0)}\frac{\partial f_i^{\mathrm{L}}(\theta,u)}{\partial u}du \quad = \quad U_{i,1}^{*j+1}(\theta) - U_{i,1}^{*j}(\theta) \tag{4.24}$$

for $j = 0, 1, \dots, r-1$. Therefore, we can minimise the inner summation terms individually in order to minimise the overall variance, and using the result above, we get

$$\frac{\partial f_i^{\mathrm{L}}(\theta,u)}{\partial U_{i,1}} = \frac{U_{i,1}^{*j+1}(\theta) - U_{i,1}^{*j}(\theta)}{U_{i,1}^{*j+1}(\theta_0) - U_{i,1}^{*j}(\theta_0)} \tag{4.25}$$

when $U_{i,1}^{*j}(\theta_0) \le u < U_{i,1}^{*j+1}(\theta_0)$.

4.3. **Linearizing proxy constraint functions.** For cases where the closed-form solution of $U_{i,1}^{*j}$ does not exist, solving for $U_{i,1}^{*j}$ can be time consuming as one typically has to rely on numerical methods. Here, we propose an efficient alternative approach to address this issue. In particular, we will work with the approximated $U_{i,1}^{*j}$ (denoted by $\tilde{U}_{i,1}^{*j}$) obtained via linearizing the proxy constraint function.

Under this approach, we first write $U_{i,1} = \eta(\eta^{-1}(U_{i,1}))$ where $\eta : \mathbb{R} \to [0,1)$ is a continuous monotone function. We then linearize the proxy constraint function around $\eta^{-1}(U_{i,1})$ and we get

$$p_i(K_i^*(\theta)) + \frac{\partial p_i(K_i^*(\theta))}{\partial \eta^{-1}(U_{i,1})}\left(\eta^{-1}\left(U_{i,1}^{*j}(\theta, U_{i,1})\right) - \eta^{-1}(U_{i,1})\right) \approx H_{i,j}(\theta).$$

We choose to linearize around $\eta^{-1}(U_{i,1})$ instead of $U_{i,1}$ as this guarantees that our estimate of the critical point is bounded between zero and one. Solving for this (approximated) critical point is now straight-forward and rearranging the equation above gives

$$\tilde{U}_{i,1}^{*j}(\theta, U_{i,1}) = \eta\left(\left(\frac{\partial p_i(K_i^*(\theta))}{\partial U_{i,1}}\frac{\partial U_{i,1}}{\partial \eta^{-1}(U_{i,1})}\right)^{-1}\left(H_{i,j}(\theta) - p_i(K_i^*(\theta))\right) + \eta^{-1}(U_{i,1})\right). \tag{4.26}$$

with

$$\frac{\partial}{\partial U_{i,1}}\tilde{U}_{i,1}^{*j}(\theta,U_{i,1}) = \eta'\left(\tilde{U}_{i,1}^{*j}(\theta,U_{i,1})\right)\times\left(\frac{\partial^2 p_i(K_i^*(\theta))}{\partial(U_{i,1})^2}\left(\frac{\partial U_{i,1}}{\partial\eta^{-1}(U_{i,1})}\right)^2+\frac{\partial p_i(K_i^*(\theta))}{\partial(U_{i,1})}\frac{\partial^2 U_{i,1}}{\partial\left(\eta^{-1}(U_{i,1})\right)^2}\right)$$
$$\times\left(p_i(K_i^*(\theta))-H_{i,j}(\theta)\right)\Big/\left(\left(\frac{\partial p_i(K_i^*(\theta))}{\partial(U_{i,1})}\right)^2\left(\frac{\partial U_{i,1}}{\partial\eta^{-1}(U_{i,1})}\right)^3\right) \tag{4.27}$$

There are many possible choices for the function η. Here we suggest using the standard normal distribution function. This is a natural choice especially when using the log-Euler discretization scheme where one has to map the uniform random variables to Gaussian random variables.

Similar to equation (4.16), once all the approximate critical points have been determined, we set

$$f_i^{\mathrm{L}}(\theta,U_{i,1}) = \left(\frac{\tilde{U}_{i,1}^{*j+1}(\theta,U_{i,1})-\tilde{U}_{i,1}^{*j}(\theta,U_{i,1})}{\tilde{U}_{i,1}^{*j+1}(\theta_0,U_{i,1})-\tilde{U}_{i,1}^{*j}(\theta_0,U_{i,1})}\right)\left(U_{i,1}-\tilde{U}_{i,1}^{*j}(\theta_0,U_{i,1})\right)+\tilde{U}_{i,1}^{*j}(\theta,U_{i,1}) \tag{4.28}$$

when $\tilde{U}_{i,1}^{*j}(\theta_0,U_{i,1}) \leq U_{i,1} < \tilde{U}_{i,1}^{*j+1}(\theta_0,U_{i,1})$. Note the difference between equations (4.16) and (4.28). Due to the linearization of the proxy constraint function, the approximate critical point $\tilde{U}_{i,1}^{*j}$ is now a function of $\tilde{U}_{i,1}$. Consequently, computing the Monte-Carlo weight becomes slightly more complicated and it is given by

$$w_i^{\mathrm{L}}(\theta) = \frac{\partial}{\partial U_{i,1}}\left[\left(\frac{\tilde{U}_{i,1}^{*j+1}(\theta,U_{i,1})-\tilde{U}_{i,1}^{*j}(\theta,U_{i,1})}{\tilde{U}_{i,1}^{*j+1}(\theta_0,U_{i,1})-\tilde{U}_{i,1}^{*j}(\theta_0,U_{i,1})}\right)\left(U_{i,1}-\tilde{U}_{i,1}^{*j}(\theta_0,U_{i,1})\right)+\tilde{U}_{i,1}^{*j}(\theta,U_{i,1})\right]. \tag{4.29}$$

Since we are working with approximate critical points rather than exact ones, there is still the possibility that pathwise discontinuities could occur, albeit with a much reduced probability. We remove these remaining discontinuities by redefining the pay-off of the bumped path so that it is determined by the events $E_{i,j}$ for the unbumped path. This will introduce a small bias. A similar approach was used in Fries and Joshi (2011). We prove in Appendix A that these discontinuities generate a bias in the value of a bumped path that is of order $(\theta-\theta_0)^2$. In consequence, when working with the pathwise method as $\theta \to \theta_0$, it disappears even after division by $\theta-\theta_0$.

Note that our linearization approach can be viewed as the first step in a Newton–Raphson search for the critical value. We could achieve a smaller bias by performing more steps, and this would, in fact, be necessary when considering higher order derivatives, since a second order bias would not disappear in the limit.

Note that the efficiency gained from linearizing the proxy constraint function comes at a cost - our choice of measure change at each time step will no longer be fully optimal. However, if before linearization, we apply a simple map that makes the proxy constraint close to linear then it will be close to optimal. For example, in interest rate applications, we can use the log of the swap-rate as the proxy constraint function instead of using the swap-rate directly.

5. Pathwise OPP method

Whilst the OPP method is effective, it still suffers from the same defects as finite differencing does in the Lipschitz continuous case: a small bias from the finite bump size; and the necessity of one simulation per sensitivity. We therefore analyze the small bump limit in this section and thereby develop a pathwise OPP method which does not have either of these problems. We use a limit result from Chan and Joshi (2011).

The pathwise OPP method relies on the interchange of the differentiation and the expectation operators, i.e.

$$\frac{\partial}{\partial\theta}\mathbb{E}\left[\left(g^{\mathrm{L}}(\theta)\prod_{i=1}^{m}w_i^{\mathrm{L}}(\theta)\right)\right]\Big|_{\theta=\theta_0}=\mathbb{E}\left[\frac{\partial}{\partial\theta}\left(g^{\mathrm{L}}(\theta)\prod_{i=1}^{m}w_i^{\mathrm{L}}(\theta)\right)\right]\Big|_{\theta=\theta_0} \tag{5.1}$$

We write

$$\alpha(\theta,U)=W(\theta,U)(g^{\mathrm{L}}\circ K)(\theta,U), \tag{5.2}$$

where W is the accumulated weights for the path. We recall the main result from Chan and Joshi (2012) on interchanging limits for derivatives of weighted Monte Carlos. Let Θ be a small open neighbourhood of θ_0 and V be open subset of Euclidean space that contains all possible realizations of U. The assumptions are

A1: $W(\theta,U)$ is an infinitely differentiable function of (θ,U) on $\Theta\times V$.

A2: $K(\theta,U)$ is an infinitely differentiable function of (θ,U) on $\Theta\times V$.

A3: There exists a measurable function $G(U)$ on V such

$$|\alpha(\theta,U)-\alpha(\theta_0,U)|\leq G(U)||\theta-\theta_0||$$

for $\theta\in\Theta$ and $U\in V$, such that $\mathbb{E}(G(Z))$ is finite.

A4: The probability of the event that $(g^{\mathrm{L}}\circ K)(\theta,U)$ is differentiable as a function of θ at θ_0 is 1.

The main result is

Theorem 1. *Suppose assumptions A1 to A4 hold, then*

$$Y'(\theta)=\mathbb{E}\left(\frac{\partial\alpha}{\partial\theta}(\theta_0,U)\right).$$

The key to applying this result is to establish that A3 holds. The OPP simulation scheme is constructed so that K^{L} are sampled in such a way that pathwise discontinuities are eliminated for every U. Therefore, $g^{\mathrm{L}}(\theta,U)$ will be a continuously differentiable function of θ locally around θ_0 for every U. There is no issue with constructing uniform bounds as the Monte-Carlo weights are constructed to have minimal second moments. Computing the price sensitivities at θ_0 is thus simplified to computing the expectation of the pathwise derivatives, i.e.

$$\mathbb{E}\left[\frac{\partial}{\partial\theta}g^{\mathrm{L}}(\theta)+g^{\mathrm{L}}(\theta)\sum_{i=1}^{m}\frac{\partial}{\partial\theta}w_i^{\mathrm{L}}(\theta)\right]\Big|_{\theta=\theta_0} \tag{5.3}$$

5.1. **Computing pathwise derivatives.** Whilst we have now derived an expression for the sensitivity. This still involves evaluating a complicated expression for each path of the simulation. It is therefore important to carry out this evaluation efficiently. For the ordinary pathwise method work has been done on using methods of algorithmic differentation to speed up such computations. In particular, Giles and Glasserman (2006) suggested using an adjoint approach to apply a general result that gradients can always be computed with at most 4 times as many operations as values, see Griewank and Walther (2008). We now show how to apply the adjoint method to this case. There are many different ways to implement such adjoint methods, we adopt the backward algorithmic differentiation approach introduced by Joshi and Yang (2011) and extended by Chan and Joshi (2010).

Using this approach, one can calculate the gradient of the (weighted) pay-off w.r.t the state variables at t_{m-1} back to t_0 easily by specifying all the simple mapping functions that govern the evolution of the state variables. Specifically, given the gradient ∇_g of a function $g : \mathbb{R}^n \to \mathbb{R}$, Joshi and Yang (2011) showed that $\nabla_{g \circ P}$ for a sufficiently simple function $P : \mathbb{R}^n \to \mathbb{R}^n$ can be obtained from ∇_g by overwriting the elements in ∇_g in a simple way. The important observation made by Joshi and Yang is that if P is the identity map in all coordinates except one and that particular coordinate depends on only one or two coordinates then the new gradient will agree except in one or two coordinates.

Chan and Joshi (2010) suggested working with a more complicated mapping function, instead of working with multiple simple functions, to reduce the computational time. They let the function $P : \mathbf{x} \to \mathbf{y}$, where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, be an identity map in all coordinates except for the s^{th} coordinate. However, instead of restricting that the s^{th} coordinate to depend only on one or two coordinates (as proposed by Joshi and Yang (2011)), they allowed it to depend on more than two coordinates as necessary. Therefore, under the Chan and Joshi approach, the mapping function P gives

$$y_j = x_j \quad \text{for} \quad j = 1, 2, \ldots, s-1, s+1, \ldots, n, \tag{5.4}$$

with

$$y_s = P_s(\mathbf{x}), \tag{5.5}$$

where $P_s : \mathbb{R}^n \to \mathbb{R}$. Based on the algorithm provided by Chan and Joshi (2010), as long as one can compute the gradient of P, the new gradient $\nabla_{g \circ P}$ can be obtained as follows:

(1) First, overwrite the gradient ∇_g as follows

$$(\nabla_g)_j = (\nabla_g)_j + (\nabla_{P_s})_j \cdot (\nabla_g)_s \tag{5.6}$$

for all j except for the s^{th} element.

(2) Then, the s^{th} element is updated as follows

$$(\nabla_g)_s = (\nabla_{P_s})_s \cdot (\nabla_g)_s, \tag{5.7}$$

and the new gradient $\nabla_{g \circ P}$ is given by the updated ∇_g.

In order to reduce the computational time, they also suggested that there is no need to overwrite the l^{th} element when computing the new gradient if y_s has no dependency on x_l (except when $l = s$) for some index l.

Before we proceed to evaluating the pathwise derivatives, we wish to emphasize that, when applying pathwise OPP method, there is no need to evolve the underlying state variable K using the OPP simulation scheme. This is because all the pathwise derivatives are evaluated at θ_0 and, given θ_0, the mapping function F_i is equivalent to F_i^{L} by construction.

In order to compute the pathwise derivatives, we perform the following mapping at the beginning of the simulation

$$\begin{bmatrix} \theta \end{bmatrix} \xrightarrow{\mathcal{H}_0} \begin{bmatrix} \theta \\ W_0^{\mathrm{L}} \\ K_0^{\mathrm{L}} \\ H_0 \end{bmatrix}$$

where W_i^{L} represents the accumulated Monte-Carlo weight up to time t_i with $W_0^{\mathrm{L}} = 1$ and $H_i = (H_{i,1}, H_{i,2}, \ldots, H_{i,r-1})$. Note that, for simplicity, here, we no longer express the state variables as a function of θ. At each time step, we can evolve the underlying state variables from t_{i-1} to t_i using the composition of the following mapping functions

$$\begin{bmatrix} \theta \\ W_{i-1}^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \end{bmatrix} \xrightarrow{\mathcal{H}_{i,1}} \begin{bmatrix} \theta \\ W_{i-1}^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \\ K_i^* \end{bmatrix} \xrightarrow{\mathcal{H}_{i,2}} \begin{bmatrix} \theta \\ W_{i-1}^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \\ K_i^* \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \end{bmatrix} \xrightarrow{\mathcal{H}_{i,3}} \begin{bmatrix} \theta \\ W_{i-1}^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \\ K_i^* \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \\ \tilde{U}_{i,1}^{*j} \\ \nabla_u \tilde{U}_{i,1}^{*j+1} \\ \nabla_u \tilde{U}_{i,1}^{*j} \end{bmatrix} \xrightarrow{\mathcal{H}_{i,4}} \begin{bmatrix} \theta \\ W_i^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \\ K_i^* \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \\ \tilde{U}_{i,1}^{*j} \end{bmatrix} \xrightarrow{\mathcal{H}_{i,5}} \begin{bmatrix} \theta \\ W_i^{\mathrm{L}} \\ K_{i-1}^{\mathrm{L}} \\ H_{i-1} \\ f_i^L \end{bmatrix} \xrightarrow{\mathcal{H}_{i,6}} \begin{bmatrix} \theta \\ W_i^{\mathrm{L}} \\ K_i^{\mathrm{L}} \\ H_{i-1} \end{bmatrix} \xrightarrow{\mathcal{H}_{i,7}} \begin{bmatrix} \theta \\ W_i^{\mathrm{L}} \\ K_i^{\mathrm{L}} \\ H_i \end{bmatrix}$$

where ∇_u^k represents the k^{th} order derivative w.r.t $U_{i,1}$. Note that θ, K_i^{L}, H_i, K_i^* are vectors while the remaining variables are scalar quantities. The mapping functions $\mathcal{H}_{i,1}$, $\mathcal{H}_{i,2}$, $\mathcal{H}_{i,6}$ and $\mathcal{H}_{i,7}$, can be decomposed into a composition of multiple simple mapping functions once the exact forms are known. In particular, the mapping functions $\mathcal{H}_{i,1}$ and $\mathcal{H}_{i,6}$ will depend on the underlying numerical scheme used. For example, the mapping functions for the LMM has been well studied by Joshi and Yang (2009). The mapping function $\mathcal{H}_{i,2}$ will depend on our choice of proxy constraint functions and the mapping function $\mathcal{H}_{i,7}$ will be determined by the financial product itself.

The remaining mapping functions $\mathcal{H}_{i,3}$, $\mathcal{H}_{i,4}$ and $\mathcal{H}_{i,5}$ can also be decomposed easily. In particular one can decompose $\mathcal{H}_{i,3}$ as follows

$$\mathcal{H}_{i,3} = \mathcal{H}_{i,3}^4 \circ \mathcal{H}_{i,3}^3 \circ \mathcal{H}_{i,3}^2 \circ \mathcal{H}_{i,3}^1$$

$$
\begin{bmatrix} \vdots \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \end{bmatrix}
\xrightarrow{\mathcal{H}_{i,3}^1}
\begin{bmatrix} \vdots \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \end{bmatrix}
\xrightarrow{\mathcal{H}_{i,3}^2}
\begin{bmatrix} \vdots \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \\ \tilde{U}_{i,1}^{*j} \end{bmatrix}
\xrightarrow{\mathcal{H}_{i,3}^3}
\begin{bmatrix} \vdots \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \\ \tilde{U}_{i,1}^{*j} \\ \nabla_u \tilde{U}_{i,1}^{*j+1} \end{bmatrix}
\xrightarrow{\mathcal{H}_{i,3}^4}
\begin{bmatrix} \vdots \\ p_i \\ \nabla_u p_i \\ \nabla_u^2 p_i \\ \tilde{U}_{i,1}^{*j+1} \\ \tilde{U}_{i,1}^{*j} \\ \nabla_u \tilde{U}_{i,1}^{*j+1} \\ \nabla_u \tilde{U}_{i,1}^{*j} \end{bmatrix}
$$

with

$$
\begin{aligned}
\tilde{U}_{i,1}^{*k} &= f_u\left(p_i, \nabla_u p_i, H_{i,k}\right) = \eta\left(\left(\frac{H_{i,k} - p_i}{\nabla_u p_i \cdot c_1}\right) + c_2\right) \\
\nabla_u \tilde{U}_{i,1}^{*k} &= f_{u'}\left(p_i, \nabla_u p_i, \nabla_u^2 p_i, H_{i,k}, \tilde{U}_{i,1}^{*k}\right) \\
&= \frac{\eta\left(\tilde{U}_{i,1}^{*k}\right) \cdot \left(\nabla_u^2 p_i \cdot (c_1)^2 + \nabla_u p_i \cdot c_3\right) \cdot \left(p_i - H_{i,k}\right)}{\left(\nabla_u p_i\right)^2 \cdot (c_1)^3}
\end{aligned}
\tag{5.8}
$$

where c_1, c_2 and c_3 are constants given by

$$
\begin{aligned}
c_1 &= \frac{\partial U_{i,1}}{\partial \eta^{-1}\left(U_{i,1}\right)} \\
c_2 &= \eta^{-1}\left(U_{i,1}\right) \\
c_3 &= \frac{\partial^2 U_{i,1}}{\partial\left(\eta^{-1}\left(U_{i,1}\right)\right)^2}
\end{aligned}
$$

for $k = j, j+1$. Since computing the gradients of f_u and $f_{u'}$ are straight-forward, one can update the gradient of the weighted price easily using the algorithm provided by Chan and Joshi (2010).

The accumulated weight W_i^{L} can be updated as follows

$$
\begin{aligned}
W_i^{\mathrm{L}} &= f_W\left(\tilde{U}_{i,1}^{*j}, \tilde{U}_{i,1}^{*j+1}, \nabla_u \tilde{U}_{i,1}^{*j}, \nabla_u \tilde{U}_{i,1}^{*j+1}, W_{i-1}^{\mathrm{L}}\right) \\
&= W_{i-1}^{\mathrm{L}} \cdot \left[c_4\left(\left(\nabla_u \tilde{U}_{i,1}^{*j+1} - \nabla_u \tilde{U}_{i,1}^{*j}\right) \cdot c_5 + \left(\tilde{U}_{i,1}^{*j+1} - \tilde{U}_{i,1}^{*j}\right) \cdot c_6 - \left(\tilde{U}_{i,1}^{*j+1} - \tilde{U}_{i,1}^{*j}\right) \cdot c_7\right) + \nabla_u \tilde{U}_{i,1}^{*j}\right]
\end{aligned}
\tag{5.9}
$$

where the values of c_4, c_5, c_6 and c_7 are given by

$$
\begin{aligned}
c_4 &= \left(\tilde{U}_{i,1}^{*j+1}(\theta_0, U_{i,1}) - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right)^{-2} \\
c_5 &= \left(U_{i,1} - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right) \cdot \left(\tilde{U}_{i,1}^{*j+1}(\theta_0, U_{i,1}) - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right) \\
c_6 &= \left(1 - \frac{\partial}{\partial U_{i,1}} \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right) \cdot \left(\tilde{U}_{i,1}^{*j+1}(\theta_0, U_{i,1}) - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right) \\
c_7 &= \left(U_{i,1} - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right) \cdot \left(\frac{\partial}{\partial U_{i,1}} \tilde{U}_{i,1}^{*j+1}(\theta_0, U_{i,1}) - \frac{\partial}{\partial U_{i,1}} \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})\right)
\end{aligned}
\tag{5.10}
$$

Since the gradient of f_W can be evaluated easily, one can work with the mapping function $\mathcal{H}_{i,4}$ directly.

Lastly, there is also no need to decompose the mapping function $\mathcal{H}_{i,5}$ as we have

$$f_i^{\mathrm{L}} = f_\mu\left(\tilde{U}_{i,1}^{*j+1}, \tilde{U}_{i,1}^{*j}\right) = c_8 \cdot \left(\tilde{U}_{i,1}^{*j+1} - \tilde{U}_{i,1}^{*j}\right) + \tilde{U}_{i,1}^{*j} \tag{5.11}$$

where the value of c_8 is given by

$$c_8 \quad = \quad \frac{U_{i,1} - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})}{\tilde{U}_{i,1}^{*j+1}(\theta_0, U_{i,1}) - \tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1})}. \tag{5.12}$$

Hence, computing the gradient of f_μ is straight-forward and one can easily update the gradient of the weighted price using the Chan and Joshi (2010) algorithm.

6. Numerical Tests and Results

For our numerical tests, we consider applying the pathwise OPP method to compute the deltas and vegas of digital caplets, double digital caplets, digital CMSes, double digital CMSes, LIBOR TARNs and CMS TARNs under the LIBOR market model. In particular, we will compare the standard errors of Greeks obtained using the pathwise OPP method against those obtained using the naive bump-and-revalue method, the pathwise PP method and the pathwise MPP method (when applicable).

6.1. **The LIBOR Market Model Specifications.** The LIBOR market model (Brace et al. (1997)) assigns dynamics to n contiguous forward rates $f_1, f_2, \ldots, f_n$ with the corresponding tenor structure of $t_0 = 0 < t_1 < \ldots < t_{n+1}$. Under this model, the forward rates are assumed to follow

$$\frac{df_j(t)}{f_j(t)} = \mu_j(f,t)dt + \sigma_j(t)dW(t), \tag{6.1}$$

where $\sigma_j(t)$ is a deterministic d-dimensional row vector and $W(t)$ is a d-dimensional column vector of uncorrelated Brownian motions. Under the spot-measure, the drift term is

$$\mu_j(f,t) = \sum_{h=\eta(t)}^{j} \frac{f_h(t)\tau_h}{1+f_h(t)\tau_h}\sigma_j(t)\sigma_h(t)^{\mathrm{T}}, \tag{6.2}$$

where $\tau_i = t_{i+1} - t_i$ and $\eta(t)$ gives the index of the next forward rate to reset at time t. Under the log-Euler discretization (see Joshi, 2003a) with $f_j(t_i) \equiv f_j(i)$, we have

$$\log f_j(i+1) = \log f_j(i) + \tilde{\mu}_j(f(i)) - \frac{1}{2}C_{jj}(i) + a_j^r(i)Z(i+1) \tag{6.3}$$

with

$$\tilde{\mu}_j(f(i)) = \sum_{h=\eta(t)}^{j} \frac{f_h(i)\tau_h}{1+f_h(i)\tau_h}C_{jh}(i)$$

where $C_{jh}(i)$ is the covariance between $\log f_j$ and $\log f_h$ from t_i to t_{i+1}, $Z(i+1)$ is a d-dimensional column vector of standard normal random variables and $a_j^r(i)$ is the j^{th} row of the $n \times d$ pseudo-root, $A(i) = (a_{js}(i))$ such that

$$A(i)A(i)^{\mathrm{T}} = C(i).$$

We can easily express the LMM discretization scheme above in the required form as specified by equation (3.1) and this permits the application of the pathwise OPP methods. For our numerical tests, we model semi-annual forward rates with a tenor structure of

$$T_0 = 0 < T_1 < \ldots < T_{n+1}$$

where $T_j = 0.5 \times j$. We assume that all the forward rates have a flat volatility structure of 20%. We also assume that the Brownian driver of the forward rates has a correlation of $\rho_{ij} = 0.5 + 0.5\exp(-0.2|T_i - T_j|)$ and reduced to the first 5 factors. For our numerical tests, we simulate the log of the forward rates instead of the forward rates directly.

6.2. **Product Specifications.**

6.2.1. *Digital Caplets.* For our digital caplet numerical tests, we compute the price sensitivities of a 5-year digital caplet with the forward rate resetting on year 5 as the underlying (i.e. $f_{10}(T_{10})$). We set the strike of the digital caplet to 10% and we assume that the initial LIBOR curve is flat at 10%. The log of the forward rate (i.e. a linear proxy constraint function) resetting at year 5 is used as the proxy constraint function.

6.2.2. *Digital CMSes.* A digital CMS is similar to a digital caplet except that a constant maturity swap-rate (CMS) is used as the underlying quantity of the contract. For our numerical tests, we consider a 5-year digital CMS with the 5-year CMS as the underlying. This digital CMS is assumed to have a strike of 10% and we further assume that the initial LIBOR curve is flat at 10% with the log of the CMS resetting at year 5 used as the proxy constraint function. Note that, the pathwise MPP method is not applicable as digital CMSes require a non-linear proxy constraint function (see Chan and Joshi (2012)).

6.2.3. *Double Digital Caplets.* A double digital caplet is similar to a digital caplet except that this product will pay 1 only when the underlying forward rate finishes between the lower and upper strikes. For the double digital caplet numerical test, we use the same setup as for the digital caplet numerical test except that we set the lower and upper strike to be 5% and 10% respectively. Note that, the pathwise MPP method is not applicable as the pay-off function of a double digital caplet has two discontinuities at each time step.

6.2.4. *Double Digital CMSes.* A double digital CMS is similar to a digital CMS except that this product will pay 1 only when the underlying CMS rate finishes between the lower and upper strikes. For the double digital CMS numerical test, we use the same setup as for the digital CMS numerical test except that we set the lower and upper strike to be 5% and 10% respectively. Note that, the pathwise MPP method is not applicable as double digital CMSes have two discontinuities at each time step.

6.2.5. *LIBOR TARNs.* Here, we consider the TARN definition in Piterbarg (2004) where the TARN is structured like an exotic swap that knocks out when the total sum of structured coupons exceeds the target coupon. Specifically, we consider computing the sensitivities of a 5.5-year LIBOR TARN with a zero initial coupon and a target coupon of 9%. This TARN is assumed to pay semi-annual inverse floating coupons of $\max(10\% - 2f_j(T_j), 0)$ and the coupons are determined at the beginning of the reset date and payable at the end of the reset period. Since LIBOR TARNs are known to have strong pay-off discontinuity effects in an upward-sloping interest-rate environment (see Fries and Joshi (2008)), the forward rates are assumed to increase linearly from $f_1 = 2.5\%$ to $f_{10} = 11.5\%$ and hence the probability of early redemption is approximately 0.5. At each time step, we set the log of the forward rate resetting at each tenor date (i.e. a linear proxy constraint function) as the proxy constraint function.

6.2.6. *CMS TARNs.* Similar to the LIBOR TARN, a CMS TARN has all the same features. However, instead of using forward rates, the coupon payments are determined using the constant maturity swap-rate (CMS). For our numerical tests, we consider computing the price sensitivities of a 5.5-year CMS TARN with a zero initial coupon and a target coupon of 9%. This TARN is assumed to pays semi-annual inverse floating coupons of $\max(10\% - 2\,\mathrm{CMS}_j, 0)$ where CMS_j is the 5-year constant maturity swap-rate resetting at time T_j with the coupons determined at the beginning of the reset date and payable at the end of the reset period. Similar to LIBOR TARN, CMS TARNs are known to have strong pay-off discontinuity effects in an upward-sloping interest rate environment. We therefore assume that the LIBOR forward rates increase linearly from $f_1 = 1\%$ to $f_{19} = 10\%$. The log of the constant maturity swap rate resetting at each tenor date is used as the proxy constraint in order to prevent pathwise discontinuities. The pathwise MPP method is not applicable in this case as CMS TARNs require a non-linear proxy constraint function (see Chan and Joshi (2012)).

6.3. **Numerical Results.** Here, we present the numerical results for deltas and vegas evaluated using the naive bump-and-revalue method, the pathwise PP method, the pathwise MPP method and the pathwise OPP method. For vegas, we first compute all the elementary vegas which are the derivatives of price with respect to pseudo-root elements (see Joshi and Kwon (2011)). We then convert them to the price sensitivities with respect to the volatility of each underlying forward rate by summing the weighted elementary vegas

$$\frac{\partial\,\mathrm{Price}}{\partial \sigma_j} = \sum_s \sum_i \frac{a_{js}(i)}{\sigma_j} \frac{\partial\,\mathrm{Price}}{\partial a_{js}(i)},$$

where σ_j is the volatility of the underlying. Under the naive bump-and-revalue method, deltas and vegas are calculated by applying finite differences to prices obtained using the base inputs and the bumped inputs. For deltas, we shift the relevant initial forward rate by 1 basis point while the perturbed inputs for vegas are obtained by shifting the base volatility by 10 basis points (i.e. $\sigma_j + 0.1\%$). Note that, under the bump and revalue method, vegas are calculated directly instead of taking the weighted sum of the elementary vegas.

The means and the standard errors of deltas and vegas for all the numerical test products are evaluated using 5000 batches of simulations, each with 5000 paths. Here, we do not present the results for the means of deltas and vegas as they are within the Monte-Carlo sampling error. The results for the standard errors are presented in tables 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6.

Overall the pathwise OPP method gives the lowest standard errors followed by the pathwise MPP method (when applicable), the pathwise PP method and the bump-and-revalue method. As we see from the numerical results for digital caplets (table 6.1), the standard error of vegas (for f_{10})

	Digital Caplet							
	Std Errors of Deltas				Std Errors of Vegas			
Forward Rates	Bump Revalue	Pathwise PP	Pathwise MPP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise MPP	Pathwise OPP
f_1	6.14%	0.08%	0.08%	0.08%	1.43%	0.02%	0.02%	0.02%
f_2	8.21%	0.10%	0.09%	0.08%	2.01%	0.04%	0.03%	0.03%
f_3	10.64%	0.11%	0.10%	0.09%	2.49%	0.05%	0.04%	0.04%
f_4	12.86%	0.14%	0.11%	0.10%	2.94%	0.07%	0.06%	0.05%
f_5	14.54%	0.17%	0.13%	0.11%	3.30%	0.09%	0.07%	0.06%
f_6	16.35%	0.20%	0.15%	0.13%	3.69%	0.11%	0.08%	0.07%
f_7	17.78%	0.24%	0.18%	0.14%	4.05%	0.14%	0.10%	0.08%
f_8	19.44%	0.29%	0.20%	0.17%	4.44%	0.17%	0.12%	0.10%
f_9	21.32%	0.34%	0.24%	0.19%	4.82%	0.21%	0.14%	0.11%
f_{10}	245.65%	35.61%	24.40%	17.38%	16.50%	6.30%	1.08%	0.76%

TABLE 6.1. Digital Caplets: Standard errors of deltas and vegas.

obtained using the pathwise OPP method is approximately 87% and 30% lower than the pathwise PP method and the pathwise OPP method respectively. Standard error reductions of a similar magnitude can be also be observed for all other financial products considered in our numerical tests and, in some extreme cases, the standard errors of Greeks obtained using the pathwise OPP method are more than 90% lower than the pathwise PP method e.g. deltas of LIBOR TARNs.

An interesting observation for LIBOR TARNs and CMS TARNs is that the naive bump-and-revalue method produces stable deltas and vegas for forward rates resetting close to the maturity of the contract. The is because, as explained by Chan and Joshi (2012), the coupons will go quickly to zero in an upward-sloping yield curve environment. Therefore if the TARN does not trigger early, the coupon payments will most likely reduce to zero and the target coupon will not be reached until final maturity. Hence, under this setting, this product is discontinuous in the first few rates but effectively continuous in the last few. Even if a trigger does occur close to the end, the size of the discontinuity is small as the difference in timing of the principal return is slight. From the LIBOR TARN and CMS TARN numerical results, we can see that both the pathwise PP method and the pathwise MPP method produce deltas and vegas with high standard errors for forward rates resetting close to the maturity of the contract. The pathwise OPP method on the other hand produces stable deltas and vegas even for forward rates resetting close to the maturity where the discontinuity effects are insignificant.

7. Conclusion

In this paper, we have presented a new approach known as the pathwise OPP method. This method is generic and can be easily applied to compute price sensitivities for financial product with discontinuous payoffs. Our new approach is in a certain sense optimal as measure changes are selected such that the variance of the likelihood ratio is minimal amongst all possible measure changes and this method also works for functions with multiple discontinuities per step as well as for cases with non-linear proxy constraint functions. Our numerical results suggested that the pathwise OPP method is significantly better than the pathwise PP method and the pathwise

	Double Digital Caplet					
	Std Errors of Deltas			Std Errors of Vegas		
Forward Rates	Bump Revalue	Pathwise PP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise OPP
f_1	5.82%	0.21%	0.21%	1.36%	0.01%	0.01%
f_2	7.26%	0.21%	0.22%	2.01%	0.02%	0.02%
f_3	9.88%	0.21%	0.23%	2.48%	0.03%	0.03%
f_4	12.24%	0.22%	0.23%	2.94%	0.04%	0.03%
f_5	14.05%	0.22%	0.24%	3.32%	0.05%	0.04%
f_6	15.70%	0.23%	0.25%	3.66%	0.06%	0.05%
f_7	16.93%	0.23%	0.26%	4.03%	0.07%	0.06%
f_8	19.27%	0.24%	0.27%	4.39%	0.08%	0.07%
f_9	21.41%	0.25%	0.28%	4.80%	0.09%	0.08%
f_{10}	238.40%	19.07%	13.23%	39.77%	10.71%	5.00%

TABLE 6.2. Double Digital Caplets: Standard errors of deltas and vegas.

	Digital CMS					
	Std Errors of Deltas			Std Errors of Vegas		
Forward Rates	Bump Revalue	Pathwise PP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise OPP
f_1	6.60%	0.06%	0.05%	1.53%	0.02%	0.02%
f_2	9.70%	0.08%	0.06%	2.12%	0.04%	0.03%
f_3	11.83%	0.10%	0.07%	2.61%	0.06%	0.04%
f_4	13.30%	0.13%	0.08%	3.06%	0.08%	0.05%
f_5	15.29%	0.16%	0.09%	3.42%	0.10%	0.06%
f_6	16.81%	0.20%	0.11%	3.75%	0.12%	0.07%
f_7	18.27%	0.24%	0.13%	4.09%	0.14%	0.08%
f_8	19.74%	0.29%	0.15%	4.45%	0.17%	0.09%
f_9	21.26%	0.34%	0.17%	4.81%	0.21%	0.11%
f_{10}	90.71%	5.45%	2.36%	7.41%	1.16%	0.21%
f_{11}	88.78%	5.16%	2.24%	6.71%	1.10%	0.18%
f_{12}	86.52%	4.85%	2.13%	6.01%	1.02%	0.14%
f_{13}	84.91%	4.55%	2.03%	5.57%	0.95%	0.12%
f_{14}	83.27%	4.27%	1.94%	5.38%	0.88%	0.11%
f_{15}	81.83%	4.01%	1.86%	5.32%	0.82%	0.12%
f_{16}	80.00%	3.76%	1.78%	5.44%	0.78%	0.12%
f_{17}	78.14%	3.52%	1.71%	5.79%	0.73%	0.14%
f_{18}	76.08%	3.29%	1.64%	6.23%	0.70%	0.16%
f_{19}	74.46%	3.07%	1.57%	6.57%	0.66%	0.18%

TABLE 6.3. Digital CMSes: Standard errors of deltas and vegas.

MPP methods. Our new method can therefore be applied to evaluate sensitivities for discontinuous integrals effectively.

Appendix A. The smallness of the bias arising from linearization of the proxy constraint

In this appendix, we show that the error produced by using unbumped events for bumped paths is of order $(\theta - \theta_0)^2$ for the OPP scheme. The main trickiness lies in the fact even if we have the true critical value for the discontinuity at θ_0, it will not map to the true critical value at θ. It is sufficient to study the case where θ is one-dimensional since the derivatives can be computed one coordinate at a time holding the other coordinates of θ fixed.

Note the following:

	Double Digital CMS					
	Std Errors of Deltas			Std Errors of Vegas		
Forward Rates	Bump Revalue	Pathwise PP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise OPP
f_1	7.02%	0.19%	0.19%	1.59%	0.01%	0.01%
f_2	11.20%	0.19%	0.20%	2.26%	0.02%	0.02%
f_3	13.15%	0.19%	0.21%	2.80%	0.03%	0.03%
f_4	14.71%	0.20%	0.22%	3.19%	0.04%	0.03%
f_5	16.06%	0.20%	0.23%	3.50%	0.05%	0.04%
f_6	17.26%	0.21%	0.24%	3.90%	0.06%	0.05%
f_7	19.25%	0.22%	0.25%	4.20%	0.07%	0.06%
f_8	21.13%	0.23%	0.26%	4.57%	0.08%	0.07%
f_9	22.52%	0.24%	0.27%	4.84%	0.10%	0.08%
f_{10}	91.49%	3.06%	2.05%	13.79%	1.45%	0.56%
f_{11}	88.97%	2.92%	1.85%	13.56%	1.36%	0.54%
f_{12}	85.74%	2.80%	1.75%	13.35%	1.30%	0.52%
f_{13}	83.99%	2.73%	1.66%	13.20%	1.24%	0.51%
f_{14}	82.18%	2.69%	1.58%	13.14%	1.20%	0.49%
f_{15}	79.73%	2.65%	1.50%	12.91%	1.17%	0.48%
f_{16}	77.96%	2.63%	1.43%	12.61%	1.15%	0.46%
f_{17}	75.25%	2.62%	1.37%	12.52%	1.12%	0.45%
f_{18}	73.69%	2.65%	1.32%	12.08%	1.10%	0.44%
f_{19}	72.53%	2.68%	1.28%	11.70%	1.08%	0.42%

TABLE 6.4. Double Digital CMSes: Standard errors of deltas and vegas.

	LIBOR TARN							
	Std Errors of Deltas				Std Errors of Vegas			
Forward Rates	Bump Revalue	Pathwise PP	Pathwise MPP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise MPP	Pathwise OPP
f_1	147.78%	132.28%	46.96%	16.89%	4.93%	2.33%	0.69%	0.28%
f_2	147.79%	89.23%	42.48%	17.41%	5.10%	3.19%	1.63%	0.30%
f_3	129.96%	67.05%	29.06%	15.91%	4.47%	4.12%	1.02%	0.22%
f_4	45.53%	53.72%	13.87%	4.04%	3.35%	4.58%	1.16%	0.21%
f_5	18.32%	45.16%	8.33%	1.36%	1.97%	5.06%	1.24%	0.17%
f_6	8.59%	39.30%	5.67%	0.66%	1.20%	5.42%	1.27%	0.14%
f_7	4.14%	33.82%	4.06%	0.37%	0.74%	5.77%	1.25%	0.11%
f_8	2.06%	29.16%	3.02%	0.24%	0.40%	6.09%	1.21%	0.10%
f_9	0.75%	26.02%	2.35%	0.18%	0.20%	6.30%	1.16%	0.09%
f_{10}	0.16%	16.86%	1.84%	0.17%	0.10%	4.84%	1.11%	0.10%

TABLE 6.5. LIBOR TARNs: Standard errors of deltas and vegas.

- from (4.28), when $\theta = \theta_0$, f_i^{L} is the identity map in u;
- the true critical points will be smooth functions of θ;
- in each interval between critical points f_i^{L} is a smooth function and will be smoothly extendible to the closed interval;
- the estimate of the critical point is correct on $\theta = \theta_0$ at the critical point, that is

$$\tilde{U}_{i,1}^{*j}(\theta_0, U_{i,1}^{*j}(\theta_0)) = U_{i,1}^{*j}(\theta_0).$$

Now we see how to focus behaviour close to the critical points. The difference between the location of $U_{i,1}^{*j}$ at θ and θ_0 is of size $\theta - \theta_0$, and f is smooth so f will not be able to map u across a critical point for θ close to θ_0 unless u is in a neighbourhood of size of order $|\theta - \theta_0|$ about $U_{i,1}^{*j}(\theta_0)$. Since

	CMS TARN					
	Std Errors of Deltas			Std Errors of Vegas		
Forward Rates	Bump Revalue	Pathwise PP	Pathwise OPP	Bump Revalue	Pathwise PP	Pathwise OPP
f_1	19.29%	6.30%	0.75%	0.46%	0.04%	0.00%
f_2	27.05%	5.82%	1.46%	0.74%	0.08%	0.01%
f_3	32.94%	5.10%	2.19%	0.99%	0.12%	0.03%
f_4	35.84%	4.81%	2.43%	1.12%	0.17%	0.04%
f_5	36.26%	4.77%	2.37%	1.21%	0.23%	0.04%
f_6	36.18%	4.64%	2.35%	1.24%	0.27%	0.05%
f_7	35.86%	4.50%	2.31%	1.20%	0.33%	0.05%
f_8	35.39%	4.37%	2.26%	1.21%	0.38%	0.06%
f_9	34.91%	4.32%	2.21%	1.25%	0.44%	0.07%
f_{10}	34.60%	3.74%	2.16%	1.32%	0.38%	0.08%
f_{11}	29.19%	3.12%	1.52%	1.33%	0.36%	0.05%
f_{12}	23.34%	2.62%	1.01%	1.25%	0.36%	0.04%
f_{13}	15.50%	2.02%	0.57%	1.00%	0.37%	0.03%
f_{14}	7.94%	1.72%	0.27%	0.66%	0.35%	0.03%
f_{15}	4.18%	1.56%	0.14%	0.41%	0.33%	0.02%
f_{16}	1.96%	1.40%	0.08%	0.27%	0.31%	0.01%
f_{17}	1.01%	1.25%	0.04%	0.15%	0.29%	0.01%
f_{18}	0.42%	1.15%	0.02%	0.06%	0.27%	0.01%
f_{19}	0.01%	1.09%	0.01%	0.00%	0.26%	0.00%

TABLE 6.6. CMS TARNs: Standard errors of deltas and vegas.

this is true we can handle the mapping about each critical point, individually. We only consider the case where u is just above or on a critical point since the case of below follows by symmetry.

Whilst the estimate of the critical point $\tilde{U}^{*j}_{i,1}(\theta_0, U^{*j}_{i,1}(\theta_0))$ is correct, $\tilde{U}^{*j}_{i,1}(\theta, U^{*j}_{i,1}(\theta_0))$ will not equal $U^{*j}_{i,1}(\theta)$. However, the fact that $U^{*j}_{i,1}(\theta) - U^{*j}_{i,1}(\theta_0)$ is of order $\theta - \theta_0$, and that the Newton–Raphson method is of second order (See Kreyszig (1988)) implies that

$$\tilde{U}^{*j}_{i,1}(\theta, U^{*j}_{i,1}(\theta_0)) - U^{*j}_{i,1}(\theta) = O((\theta - \theta_0)^2).$$

In order to ease notation, and since we are only considering a single critical point, we now translate to make the true $U^{*j}_{i,1}(\theta_0)$ equal to zero. We also translate θ_0 to 0. We also rescale our uniform to be supported on $[-1/2, 1/2]$; since we are only trying to prove that the probability of u mapping to a point below 0 is $O(\theta^2)$ this does not spoil our result.

We also drop the subscripts from our notation. We also write $E_l(u, \theta)$ for $\tilde{U}^{*l}_{i,1}(\theta, u)$. We write $F_l(\theta)$ for $U^{*l}_{i,1}(\theta)$. We have

$$F_0(0) = 0 = E_0(u, 0).$$

We have

$$f(u, \theta) = \frac{E_1(u, \theta) - E_0(u, \theta)}{E_1(u, 0) - E_0(u, 0)}(u - E_0(u, 0)) + E_0(u, \theta).$$

We have to show that

$$\mathbb{P}(f(u, \theta) < F_0(u, \theta), u > 0) = O(\theta^2).$$

We can rewrite

$$f(u,\theta) = \left(1 + \frac{(E_1(u,\theta) - E_1(u,0)) - (E_0(u,\theta) - E_0(u,0))}{E_1(u,0) - E_0(u,0)}\right)(u - E_0(u,0)) + E_0(u,\theta),$$
$$= u + (E_0(u,\theta) - E_0(u,0)) + \frac{(E_1(u,\theta) - E_1(u,0)) - (E_0(u,\theta) - E_0(u,0))}{E_1(u,0) - E_0(u,0)}(u - E_0(u,0)).$$

We can regard this expression as a sum of three terms. The last term is smooth and vanishes on both $\theta = 0$ and $u = 0$, and so can be written as $u\theta G(u,\theta)$ for a smooth function G. So,

$$f(u,\theta) - F(\theta) = u + (E_0(u,\theta) - E_0(u,0) - F(\theta)) + u\theta G(u,\theta).$$

Now, we analyze the second term. It is zero on $\theta = 0$. We also have that at $E_0(0,0) = 0$ and $E_0(0,\theta) - F(\theta)$ vanishes to second order in θ. Using Taylor's theorem, we therefore have that it can be written as

$$\theta^2 H(u,\theta) + \theta u K(u,\theta),$$

with H and K smooth.

Collecting terms and setting $M = G + K$, we need to show that the probability of

$$u + \theta^2 H(u,\theta) + \theta u M(u,\theta) < 0, \ u > 0$$

is of order θ^2. Since M is smooth, in a small enough neighbourhood in θ, $\theta u M > -u/2$, so on that set, our event is contained in

$$u/2 + \theta^2 H(u,\theta) < 0, \ u > 0.$$

The function H will be a bounded near $(0,0)$, so for some C our event is smaller than

$$u < C\theta^2, \ u > 0.$$

This is clearly of $O(\theta^2)$.

Having established that the set on which the wrong event occurs is of $O(\theta^2)$ probability, it follows immediately from the integrability of the two pay-offs that the difference between the two integrals is also $O(\theta^2)$, and we are done.

References

[1] Benhamou E., (2003). Optimal Malliavin Weighting Function for the Computation of the Greeks, *Mathematical Finance*, Volume 13, Issue 1, 37–53.

[2] Brace, A., Gatarek, D., and Musiela, M. (1997). The market model of interest-rate dynamics, *Mathematical Finance* **7**, 127–155.

[3] Brace, A., (2007). *Engineering BGM,* Chapman and Hall.

[4] Broadie, M., and Glasserman, P. (1996). Estimating security price derivatives using simulation. *Management Science* **42**(2), 269-285.

[5] Denson, N., and Joshi, M. S. (2009). Flaming Logs. *Wilmott Journal* Volume **1** Issue **5-6**, 259-262.

[6] Chan, J. H., and Joshi, M. S., (2010). First and Second Order Greeks in the Heston Model. http://ssrn.com/abstract=1718102.

[7] Chan, J. H., and Joshi, M. S. (2011). Minimal Partial Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks, *Journal of Computational Finance*, Vol 15, No 2, 77–109

[8] Chan, J. H., and Joshi, M. S., (2012). Fast Monte-Carlo Greeks for Financial Products with Discontinuous Pay-Offs. *Mathematical Finance*, DOI: 10.1111/j.1467-9965.2011.00509.x

[9] Friedlander, F. G., and Joshi, M. S. (1999). *Introduction to the Theory of Distributions* . Cambridge University Press, Cambridge.

[10] Fries, C. P (2007a). Localized Proxy Simulation Schemes for Generic and Robust Monte-Carlo Greeks. http://ssrn.com/abstract=984744

[11] Fries, C. P (2007b). *Mathematical Finance. Theory, Modelling, Implementation.* Wiley, New York.

[12] Fries, C. P., and Joshi, M. S. (2008). Partial proxy simulation schemes for generic and robust Monte Carlo Greeks. *The Journal of Computational Finance* **11**(3),79-106.

[13] Fries, C. P., and Joshi, M. S. (2011), Perturbation stable conditional analytic pricing scheme for auto-callable products, *International Journal of Theoretical and Applied Finance,* Vol 14, Iss 2, 197–219

[14] Fries, C. P., and Kampen, J. (2007) Proxy simulation schemes for generic robust Monte Carlo sensitivities, process oriented importance sampling and high accuracy drift approximation. *The Journal of Computational Finance* **10**(2), 97-128.

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

[16] Giles, M., and Glasserman, P. (2006). Smoking Adjoints: fast Monte Carlo Greeks. *Risk, January 2006*, 92–96.

[17] Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering (Stochactic Modelling and Applied Probability).* Springer, New York.

[18] Griewank, A. and, Walter, B. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* SIAM.

[19] Heidergott, B. and Vazquez-Abad, F.J. (2006). Measure-valued differentiation for random horizon problems. Markov Processes Related Fields, Vol. 12, pp. 509-536

[20] 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

[21] Heidergott, B., Vazquez-Abad, F.J. and Volk-Makarewicz, W. (2008). Sensitivity estimation for gaussian systems. European Journal of Operations Research, Vol. 187, pp. 193-207

[22] Kreyszig, E. (1988). *Advanced Engineering Mathematics.* Wiley, New York.

[23] Hong, L. J., and Liu, G. (2008). Kernel estimation of the Greeks for options with discontinuous payoffs. To appear in *Operations Research*

[24] L. Hörmander, The analysis of linear partial differential operators I, Springer–Verlag (1983)

[25] Jäckel, P. (2002). *Monte Carlo Methods in Finance.* Wiley, New York.

[26] Joshi, M. S. (2003a). *The Concepts and Practice of Mathematical Finance.* Cambridge University Press, Cambridge.

[27] Joshi, M. S. (2003b). Rapid computation of drifts in a reduced factor LIBOR market model. *Wilmott Magazine*, May.

[28] Joshi, M.S., and Kainth, D.S. (2004), Rapid computation of prices and deltas of nth to default swaps in the Li Model, *Quantitative Finance,* Vol **4**, No **3**, 266-275

[29] Joshi, M. S., and Kwon, O. K. (2011). Monte Carlo market Greeks in the displaced diffusion LIBOR market model, Vol 14, No 2, 23–37

[30] Joshi, M. S., and Yang, C. (2011). Fast Delta Computations in the Swap-Rate Market Model, *Journal of Economic Dynamics and Control,* 35 (2011), 764–775

[31] L'Ecuyer, P. (1990), A unified view of the IPA, SF, and LR gradient estimation techniques, *Management Science,* volume 3 **6** no. **11** 1364-1383.

[32] Lyuu, Y.D., and Teng, H. W. (2008). Effcient and Unbiased Greeks of Rainbow and Path-Dependent Options Using Importance Sampling. http://www.mfa-2008.com/papers/Greeks-mfa.pdf

[33] Piterbarg V. V. (2004). TARNs: models, valuation, risk sensitivities. *Wilmott Magazine* **14**,62-71.

[34] Rott, M. G., and Fries, C. P. (2005) Fast and Robust Monte Carlo CDO Sensitivities and their Efficient Object Oriented Implementation. http://ssrn.com/abstract=73256

[35] Vazquez-Abad, F.J. and Heidergott, B. (2008). Measure-valued differentia- tion for Markov Chains. Journal of Optimization Theory and Applications, Vol. 136, pp. 187-209

Centre for Actuarial Studies, Department of Economics, University of Melbourne, Victoria 3010, Australia

E-mail address: `jiunhong@hotmail.com`

E-mail address: `mark@markjoshi.com`