Operations Research Letters 40 (2012) 195–202

Contents lists available at SciVerse ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

Denoising Monte Carlo sensitivity estimates

Wanmo Kang [a], Kyoung-Kuk Kim [b], Hayong Shin [b,*]

[a] Korea Advanced Institute of Science and Technology, Department of Mathematical Sciences, Republic of Korea
[b] Korea Advanced Institute of Science and Technology, Industrial and Systems Engineering, Republic of Korea

ARTICLE INFO

Article history:
Received 11 October 2011
Accepted 7 January 2012
Available online 16 January 2012

Keywords:
Monte Carlo methods
Sensitivity estimation
Denoising
Near-interpolation

ABSTRACT

Monte Carlo methods are in widespread use both in academia and industry. We are, in particular, interested in improving sensitivity estimates obtained from Monte Carlo experiments with respect to given parameter values, motivated by, but not restricted to, financial applications. Denoising and interpolation methods, which have been used for a long time in many different areas, are proposed in a new form which is quadratic, easy to implement, and tailored to our objectives. This heuristic approach is supported by numerical experiments.

1. Introduction

Many complex systems do not render themselves to concrete mathematical analysis or intuitive dynamics, which often make us resort to simulation methods. Especially for systems with stochastic elements or in random environments, Monte Carlo simulation methods have been popular approaches for decades, finding applications across almost all scientific disciplines. In particular, with a relatively short history, such methods have been accepted by financial communities which include academics and financial practitioners for the purpose of pricing financial products, among which products with highly complex payoff structures find Monte Carlo methods most useful. More importantly from the financial risk management point of view, the main problem is sensitivities of such products with respect to underlying financial variables which are stochastic in nature, by which a risk manager can adjust the risk exposure of a portfolio by entering into appropriate transactions. However, for complex stochastic models with uncertainties due to Monte Carlo sampling errors, sensitivity estimates can often be quite *unstable*, in the sense that estimates might fluctuate heavily for close values of underlying variables. This is even more so when sensitivities of order higher than one are required. Our motivation to search for a heuristic but effective numerical method for stable sensitivity estimates starts from these observations.

We note that this question is not just limited to financial products but also applies to general complex stochastic systems. The particular situation we have in mind is when limited computational resources are available at hand and thus we have a budget allocation problem such as how to select computational grid points for higher-order sensitivities of a certain function of a system (or an expected payoff) at a fixed value of a continuous parameter. Of course, there are several well known methods of estimating sensitivities such as direct gradient estimation, using a pathwise method, likelihood ratio method, or other variants. See, for instance, Asmussen and Glynn [3] and Fu [7] for a brief introduction. However, such methods require certain structural properties of the underlying stochastic model. For example, those methods are applicable only when we can interchange the order of limit and integration (possibly after some manipulations), and moreover, pathwise derivatives of integrands or density functions of associated probability measures should be available. We note that there have been some approaches to extend the applicability of such methods, for example, see Liu and Hong [14] where the authors apply the pathwise method to estimate the sensitivities of financial products with discontinuous payoffs. Nevertheless, the complexity of stochastic models often hinders us from utilizing such well established methods, and we are led to consider more "naive" approaches like finite difference schemes with or without common random numbers.

Our proposed method treats Monte Carlo sampling errors as noise. And its final product out of noisy data are *denoised* multiple function values which we eventually use for the computation of sensitivity estimates. In spirit, our approach resembles simulation metamodels because we hope to get information about system behaviors as parameters change and metamodels approximate the unknown true function by assuming certain functional forms and by considering explicit noise terms. See Barton and Meckesheimer [4] and references therein for more details about

* Corresponding author.
E-mail addresses: wanmo.kang@kaist.edu (W. Kang), catenoid@kaist.ac.kr (K.-K. Kim), hyshin@kaist.ac.kr (H. Shin).

0167-6377/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2012.01.006

metamodels. Actually as explained later, it turns out that our approach is a variant of spline approximations. However, ours differs from those works in that we do not start from presumed functional forms, and additionally the proposed method is tailored to compute sensitivities of certain orders. We note that there are two other streams of papers that this research can be related to. One is from the computer design community where the objective is to produce visually satisfactory graphics from noisy data, e.g., Sapidis and Farin [17], Zhang et al. [20]. The other is the mathematical interpolation theories for noisy data such as Hutchinson and de Hoog [10] and Kersey [12]. More is to be explained as we describe the method below in Section 2. In the following section, several numerical examples are given for a demonstration of its performance. Section 4 concludes.

2. Method and formulation

2.1. Set-up

Consider parameterized random variables $\{Y(\theta): \theta \in \Theta \subset \mathbb{R}\}$ where Θ is an open domain. Let us denote its expectation by

$$\alpha(\theta) = \mathsf{E}[Y(\theta)], \tag{1}$$

which is obtained via Monte Carlo simulation. We are interested in the derivatives of the expected value at a fixed parameter θ_0, $(d^k/d\theta^k)\alpha(\theta)\big|_{\theta=\theta_0} =: \alpha^{(k)}(\theta_0)$. In particular, the first and second order derivatives, $\alpha'(\theta_0)$ and $\alpha''(\theta_0)$, are the primary target and our exposition is focused on them. Sometimes direct gradient estimation approaches are applicable, e.g., in the (s, S) system from inventory management and the sensitivity of the performance function with respect to re-order point s, or in the call option from financial engineering and the sensitivity of the call price with respect to initial stock price. However, the applicability of such methods is problem dependent, and hence we consider finite difference schemes (FDS). General background materials on FDS can be found in Asmussen and Glynn [3] or Glasserman [8]. In our consideration of FDS, the use of common random numbers is not excluded.

The usual approach of FDS would be to select two or three points for the computation of $\alpha'(\theta_0)$ or $\alpha''(\theta_0)$. But, as often noted in the practice of hedging risks of financial products, the so called delta ($\alpha'(\cdot)$) or gamma ($\alpha''(\cdot)$), which are essential ingredients in risk management, can be quite unreliable due to the simulation errors. Our first question is then: Is there a way of obtaining more reliable sensitivity estimates by simulating at more than 2 or 3 points near θ_0, which presumably would provide us with more information about the true function $\alpha(\cdot)$ but with less accuracy due to decreased simulation efforts at each grid point? One possible solution is to approximate the unknown function $\alpha(\cdot)$ by a certain functional form as in the metamodeling framework. For example, one can consider polynomial regressors or splines to interpolate simulation outputs, but it should be taken into account that they contain sampling errors. Other examples that apply metamodeling ideas to random simulation include Ankenman et al. [2] and van Beers and Kleijnen [18], both of which are based on Kriging.

Now, suppose that we estimate $\alpha(\theta)$ by n number of simulation trials, i.e., $\hat{\alpha}_n(\theta) = n^{-1}\sum_{i=1}^n Y_i(\theta)$ where $\{Y_i(\theta)\}_{i=1}^n$ is the i.i.d. sample of $Y(\theta)$. An additional outcome of Monte Carlo simulation is the confidence interval based on sample standard deviation $s_n(\theta) = \left((n-1)^{-1}\sum_{i=1}^n (Y_i(\theta) - \hat{\alpha}_n(\theta))^2\right)^{1/2}$. We denote the $(1-a)$-th percentile confidence interval as follows:

$$CI(\theta) = \left(\hat{\alpha}_n(\theta) - z_{a/2}\frac{s_n(\theta)}{\sqrt{n}}, \hat{\alpha}_n(\theta) + z_{a/2}\frac{s_n(\theta)}{\sqrt{n}}\right) \tag{2}$$

where $z_{a/2}$ is the $(1-a/2)$-quantile of the standard normal distribution. If we run simulation at multiple points, say $\theta_{-m}, \ldots, \theta_m$, then $\{CI(\theta_i)\}_{i=-m}^m$ is obtained in addition to $\{\hat{\alpha}_n(\theta_i)\}_{i=-m}^m$. This leads us to our second question: Can we utilize this information in some way? The next two subsections document the solution approach we take to address the above two questions, including some background.

2.2. Denoising techniques

In the simulation experiment, $\hat{\alpha}_n(\theta) - \alpha(\theta)$ is an error which is a random variable that is approximately normally distributed for large n values. If we treat this term as a noise ϵ so that $\hat{\alpha}_n(\theta) = \alpha(\theta) + \epsilon$, then one natural idea to smooth out random noises ϵ from observed values is to apply denoising techniques well developed in the signal/image processing community. The following is one typical denoising formulation which can be found, for example, in Chan and Chen [6] and Rudin et al. [16]: for a d-dimensional function $u(\cdot)$ and a parameter space $\mathcal{X} \subset \mathbb{R}^d$,

$$\begin{aligned} &\min_{u\in\mathcal{C}^2(\mathcal{X})} \int_{\mathcal{X}} \|\Delta u(x)\|^2 dx \\ &s.t. \quad \int_{\mathcal{X}} u(x)dx = \int_{\mathcal{X}} u_0(x)dx, \\ &\quad \int_{\mathcal{X}} (u(x) - u_0(x))^2 dx \le \text{constant} \end{aligned} \tag{3}$$

where $u_0(\cdot)$ is the observed data and $\Delta u(\cdot)$ is the Laplacian, i.e., $\Delta u(x) = \sum_{j=1}^d (\partial^2/\partial x_j^2)u(x)$. The objective function allows various interpretations such as (i) sum of curvature if we regard it as a thin shell or (ii) the coefficients of high order terms. This optimization problem seeks a twice differentiable function that does not deviate from the observation $u_0(\cdot)$ too much by the constraints on its mean and variance. Hence, it is as if the optimal solution minimizes the bending energy in the case of (i) or weakens the higher frequency terms in (ii), i.e., low-pass filtering. In either case, a noisier shape increases both quantities and we can get a smoothed shape after solving the optimization problem. One general guidance regarding denoising is that if we want to have a denoised function with smooth r-th derivatives, then we minimize the squared sum of $(r+2)$-th derivatives under some reasonable conservation constraints.

Assume now that there are $2m+1$ parameter values, say $\theta_{-m}, \ldots, \theta_m$, with corresponding noisy observations $\hat{\alpha}_n(\theta_i)$ for $i = -m, \ldots, m$. With this amount of data rather than $\hat{\alpha}_n(\theta)$ for all $\theta \in \Theta$, variants of the above denoising formulation have been suggested with higher order derivatives as the objective function (see Berghaus and Cannon [5], Hutchinson and de Hoog [10], or Woltring [19]):

$$\begin{aligned} &\min_{u\in\mathcal{C}^k(\Theta)} \int_{\Theta} |u^{(k)}(\theta)|^2 d\theta \\ &s.t. \quad \sum_{i=-m}^m (u(\theta_i) - \hat{\alpha}_n(\theta_i))^2 \le B \end{aligned} \tag{4}$$

where B is a given positive constant or

$$\min_{u\in\mathcal{C}^k(\Theta)} \int_{\Theta} |u^{(k)}(\theta)|^2 d\theta + \rho \sum_{i=-m}^m (u(\theta_i) - \hat{\alpha}_n(\theta_i))^2 \tag{5}$$

where $\rho > 0$ is a Lagrange multiplier. Reinsch [15] states some conditions under which the existence and the uniqueness of the solutions to both (4) and (5) are guaranteed and that the solutions, in fact, are splines of degree $2k-1$. We, however, note that these formulations control the deviations of $u(\cdot)$ from $\hat{\alpha}_n(\theta_i)$ collectively, the main reason being that the observed information $\hat{\alpha}_n(\theta_i)$'s is not enough to build a separate bound for each parameter value θ_i.

2.3. Near-interpolation with discrete approximation

As we noted earlier, the Monte Carlo runs generate information on confidence intervals as in (2), i.e. $CI(\theta_i)$ for each parameter value θ_i. We start by modifying the denoising formulation (4) with this additional information $\{s_n(\theta_i)\}_{i=-m}^{m}$ in contrast with the ordinary FDS that uses the center values $\{\hat{\alpha}_n(\theta_i)\}_{i=-m}^{m}$ only. Instead of controlling $u(\theta_i) - \hat{\alpha}_n(\theta_i)$ in an aggregating fashion, we insert multiple constraints as follows:

$$\begin{aligned} &\min_{u\in\mathcal{C}^k(\Theta)} \quad \int_{\Theta} |u^{(k)}(\theta)|^2 d\theta \\ &s.t. \quad |u(\theta_i) - \hat{\alpha}_n(\theta_i)| \le \delta \frac{s_n(\theta_i)}{\sqrt{n}}, \quad i = -m, \ldots, m \end{aligned} \tag{6}$$

where δ is some positive real number. It is also convenient to use equally spaced parameter values, say $\theta_i = \theta_0 + i \times h$ for a positive value h, and to have the domain $\Theta = [\theta_{-m} - 0.5h, \theta_m + 0.5h]$. Then, h is the control variable by which we fix grid points and obtain local information about $\alpha(\cdot)$ near θ_0. The parameter δ is obviously one that determines the level of deviations of $u(\cdot)$ from $\hat{\alpha}(\theta_i)$ values. For example, if $\delta = 0$, then it is the usual interpolation problem with given finitely many function values. In this sense, the above optimization problem is called *near-interpolation*. The existence and uniqueness of the solution to (6) together with a numerical algorithm is shown in Kersey [12,13]. Indeed, its optimal solution turns out to be a polynomial spline curve of degree $2k-1$. Following the general rule suggested by image denoising literature, we use $k = r+2$ for smooth r-th derivatives, and thus the solution curves become splines of degrees 5 and 7, respectively for the first and the second derivatives. From now on, we focus on $k = 4$ case for an illustration and also because that case is more relevant to our motivation.

Toward the final formulation, we introduce a discrete approximation to $u^{(4)}(\theta)$ in (6), which eases the implementation. This is simply a higher-order FDS and given by

$$\begin{aligned} \int_{\Theta} |u^{(4)}(\theta)|^2 d\theta \approx \frac{1}{h^4} \sum_{i=-m+2}^{m-2} \Big(& u(\theta_{i-2}) - 4u(\theta_{i-1}) + 6u(\theta_i) \\ & - 4u(\theta_{i+1}) + u(\theta_{i+2}) \Big)^2. \end{aligned}$$

And denoting $u(\theta_i)$ by u_i for notational convenience, we have a quadratic programming with linear inequality constraints

$$\begin{aligned} &\min_{\{u_i\}_{i=-m}^{m}} \quad \sum_{i=-m+2}^{m-2} \Big(u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2} \Big)^2 \\ &s.t. \quad |u_i - \hat{\alpha}_n(\theta_i)| \le \delta \frac{s_n(\theta_i)}{\sqrt{n}}, \quad i = -m, \ldots, m. \end{aligned} \tag{7}$$

This is a convex optimization with linear constraints, and thus attains a unique local minimum which is also the global optimum. The solution, which can be obtained quite easily by applying built-in functions in software packages such as MATLAB, is a vector of size $2m+1$ rather than a $\mathcal{C}^4(\Theta)$ function. Our sensitivity estimates are, then, defined by

$$\begin{aligned} \widehat{\alpha'}(\theta_0) = \nabla u_0 &:= \frac{u_1 - u_{-1}}{2h}, \\ \widehat{\alpha''}(\theta_0) = \Delta u_0 &:= \frac{u_1 - 2u_0 + u_{-1}}{h^2}. \end{aligned} \tag{8}$$

There still remain some technical questions in this approach. With a fixed computational budget, there is a trade-off between the numbers m and n, which are the number of grid points and the number of simulation runs per point. If we increase m, then we have more information about the true function $\alpha(\cdot)$ at least locally. However, this action widens the confidence intervals due to smaller n, which means the decreased reliability of each simulation output. (Later in the numerical experiments, we demonstrate the performance differences for different m values.) In addition, we need to determine the parameter values of δ and h. Even though any practical choice would be a case-by-case basis, we state some general properties in the next paragraph.

Glynn [9] briefly discusses the asymptotically optimal choice of h for FDS. This ideal increment is given as

$$h_n = h_* \times n^{-\eta}, \tag{9}$$

where h_* and η have to be determined repeatedly for individual problems, depending on some higher derivative which is not known at the time of simulation. Nevertheless, the optimal choice of h yields the convergence rate of FDS equal to $O(n^{-\eta})$ for $0 < \eta < 0.5$ as n increases. Now, let us consider ∇u_0 and Δu_0 above. We denote the usual first and second derivatives at θ_0 from FDS by $\nabla\alpha_0 = (\hat{\alpha}_n(\theta_1) - \hat{\alpha}_n(\theta_{-1}))/(2h)$ and $\Delta\alpha_0 = (\hat{\alpha}_n(\theta_1) - 2\hat{\alpha}_n(\theta_0) + \hat{\alpha}_n(\theta_{-1}))/h^2$. Due to the constraints in the optimization problem (7), we obtain

$$\begin{aligned} |\nabla u_0 - \nabla\alpha_0| &\le \delta \frac{s_n(\theta_1) + s_n(\theta_{-1})}{2h\sqrt{n}}, \\ |\Delta u_0 - \Delta\alpha_0| &\le \delta \frac{s_n(\theta_1) + 2s_n(\theta_0) + s_n(\theta_{-1})}{h^2\sqrt{n}}. \end{aligned}$$

Therefore, their asymptotic deviations are bounded by $O(\delta h^{-1} n^{-0.5})$ and $O(\delta h^{-2} n^{-0.5})$, respectively. If we choose δ dynamically as $\delta_n = \text{constant} \times h_n$ or $\delta_n = \text{constant} \times h_n^2$, then we have the asymptotic deviation rate $O(n^{-0.5})$. However, this is the upper bound of any convergence order for FDS as argued in Glynn [9]. We state this observation as follows.

Proposition 1. *Suppose that h_n is the asymptotically optimal increment as in (9) with convergence rate of $n^{-\eta}$ for $0 < \eta < 0.5$. For fixed m, if we set $\delta_n = O(h_n)$ or $O(h_n^2)$, respectively, then the denoising estimates (8) have the same convergence rate as FDS.*

Different problems would require separate evaluations of optimal h and δ values, and such an evaluation is found to be quite a difficult task in practice. We, instead, choose h values that are intuitively appealing and practically relevant, e.g., 1% changes of parameter θ. As for δ, we set it equal to one. This choice makes the outcome u_0 of (7) stay in the original confidence interval $CI(\theta_0)$. In Section 3.1, we demonstrate the effects of δ and support the use of $\delta = 1$ in our numerical tests.

Before moving onto the next section, we comment on possible extensions of the near-interpolation approach to higher dimensional settings. Unfortunately, there is no existence result for the solution to (6) for functions with several variables. However, the modified formulation (7) can be easily extended to higher dimensions, and the solution can be found by any convex quadratic optimization solver. In this paper, we restrict our exhibition to the one dimensional case for illustrative purposes. But, in the remark below, we briefly mention a possible solution approach.

Remark. Suppose that the function of interest is $\alpha(\theta, \phi)$ and we want to estimate

$$\frac{\partial^2}{\partial\theta^2}\alpha(\theta,\phi)|_{\theta=\theta_0,\phi=\phi_0}, \qquad \frac{\partial^2}{\partial\phi^2}\alpha(\theta,\phi)|_{\theta=\theta_0,\phi=\phi_0}$$

at the same time. Let us denote their Monte Carlo estimates with sample size n by $\hat{\alpha}_n(\theta_i, \phi_0)$ and $\hat{\alpha}_n(\theta_0, \phi_j)$ where $i = -l, \ldots, l$ and $j = -m, \ldots, m$. Their standard deviations can be similarly defined. Then, the near-interpolation scheme in this setting takes

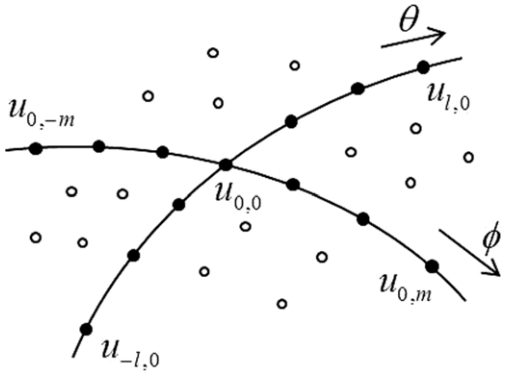

Fig. 1. Graphical illustration of changing θ and ϕ.

the following form: with $u_{i,0} := u(\theta_i, \phi_0)$ and $u_{0,j} := u(\theta_0, \phi_j)$, we minimize

$$A \sum_{i=-l+2}^{l-2} \left(u_{i-2,0} - 4u_{i-1,0} + 6u_{i,0} - 4u_{i+1,0} + u_{i+2,0}\right)^2 + B \sum_{j=-m+2}^{m-2} \left(u_{0,j-2} - 4u_{0,j-1} + 6u_{0,j} - 4u_{0,j+1} + u_{0,j+2}\right)^2$$

for some positive numbers A, B, by changing $\{u_{i,0}\}_{i=-l}^{l}$ and $\{u_{0,j}\}_{j=-m}^{m}$ under similar linear inequality constraints as in (7). Constants A and B are relative weights between the smoothness measures in θ and ϕ directions. Note that, in this formulation, we change the value of each variable θ and ϕ separately, as shown in Fig. 1. In this figure, the $u_{i,0}$ and the $u_{0,j}$ are denoted by solid dots. Different types of formulations are also possible, e.g., tilting both variables simultaneously (denoted by empty dots in Fig. 1), which is better suited for cross derivatives. We leave this issue as a topic to be more fully explored in the future.

Our last comment is that the above formulation (7) can also be extended to estimation of sensitivities at multiple points simultaneously. Actually, we consider this problem in Kang et al. [11] where, however, a different numerical method is employed to address this issue. Even so, the central theme remains the same and we refer the interested reader to the paper for more discussions.

3. Numerical results

While the methods introduced in the previous section are quite general in terms of applications, our primary focus is on the risk management of financial derivatives products. As noted in the literature, e.g., Andersen and Piterbarg [1], this task requires sensitivity computations of such products which are fed back into risk management systems to produce profit and loss prediction and analysis, for instance. However, complex contracts are often quite difficult to obtain stable sensitivity estimates (the so called Greeks that include delta/gamma introduced in Section 2.1 above) and thus they provide us with a natural setting in which the near-interpolation scheme (NIS) becomes potentially useful.

In this section, we work with a particular financial product that is called an equity linked security (ELS) and it is quite a popular type of equity derivative contract that is actively traded in the Korean financial market. A detailed payoff structure is explained below to exhibit its features and we note that many financial contracts indeed contain payoff features with a similar level of complexity. For the ease of exposition, we restrict ourselves to the estimation of delta ($\alpha'(\theta_0)$) and gamma ($\alpha''(\theta_0)$) values. In (1), θ corresponds to the price of the underlying asset, $Y(\theta)$ stands for the discounted random payoff, and $\alpha(\theta)$ is the value of the ELS.

The payoff function of the ELS depends on the relative value of the underlying asset with respect to the initial value, i.e., $\hat{S}_t = S_t/S_0, t \in [0, T]$, S_t being the stock price at time t. Here, T indicates the contract maturity and we set it equal to 3 years in numerical experiments. In addition, the contract has six early redemption dates $t_i = iT/6$, $i = 1, \ldots, 6$, which mean that the contract ends at time t_i if a certain condition on $\hat{S}$ is met at that time. Note that T is also regarded as an early redemption date because this feature precedes others. More specifically,

1. if $\hat{S}_{t_i} > LB_i$ for some i for the first time, then the contract expires at time t_i with payoff $1 + 0.1i$ to the investor (where $LB = (0.9, 0.9, 0.85, 0.85, 0.8, 0.8)$ in this example),
2. if there was no early redemption and $\min_{0\le t\le T} \hat{S}_t > 0.6$, the payoff is 1,
3. the final payoff is $\hat{S}_T$ otherwise.

Here $\min_{0\le t\le T} \hat{S}_t$ is the minimum of daily closing prices until maturity.

The computation of Greeks for this product is non-trivial by any of the existing methods for sensitivity estimation. The most challenging estimation problem is when we are close to the last early redemption date other than the maturity T because, first, its dynamics are similar to that of a digital option when we are close to T (hence, relatively well understood), and second, demand for hedging is little when we are far away from T. Therefore, we consider two cases for the current time t, namely, two weeks and one month from the fifth early redemption date t_5. In terms of remaining time-to-maturity, they are 6.5 months and 7 months. We also assume that risk-free interest rate is 0.05 and the volatility is 40%. As for θ_0, we estimate delta and gamma values at $S_t = 90, \ldots, 110$ while $S_0 = 125$. Moreover, at each θ_0, we set $h = 1$ which is the distance between nearby points.

Remark. This choice of the perturbation parameter h is made because, first, it is a convenient and intuitive value to be used in practice, and second, the optimal h stated in (9) is not known until we find out h_* and η for the problem at hand. Fig. 5 and the left panel of Fig. 6 in Appendix plot the relative root mean square errors (RRMSEs) of gamma values for digital options when the FDS is used. The first two have an optimal h around 8.1 and 2.2, respectively, while the optimal value of the third is around 1. As illustrated, this optimal h varies a lot, depending on parameter values. The right panel of Fig. 6 shows the RRMSEs of the same product as the left one when the NIS is implemented. By having $n = 20,000$ for each θ_i, we have the same computational budget for both cases. In that figure, we observe that the NIS is less sensitive to the h value, thereby making the method a potentially useful tool when the optimal h is practically not convenient to find.

Recall that the FDS uses three points $\{\theta_{-1}, \theta_0, \theta_1\}$ for the first two sensitivities and the NIS (also DNS in (10)) needs $\{\theta_i\}_{i=-m}^{m}$. For a fair comparison, we allocate a total budget equally among those points, and then estimate $\alpha(\theta_0)$, $\alpha'(\theta_0)$, and $\alpha''(\theta_0)$. In the tests below, the budget is set to be either 3×10^5 or 3×10^6 simulation trials. As a measure of performance, after obtaining such estimates for all of 21 S_t values, we look at the RRMSE in % denominated by the average of true values. Since the price of the ELS does not have any closed form expression, the estimates from the 10^7 Monte Carlo runs are regarded as the true values. Lastly, our computations are based on the geometric Brownian motion for the model of the stock price dynamics.

3.1. Selection of δ value for NIS

There is yet one additional parameter δ to be specified for the implementation of NIS as in (7). From many tests with various

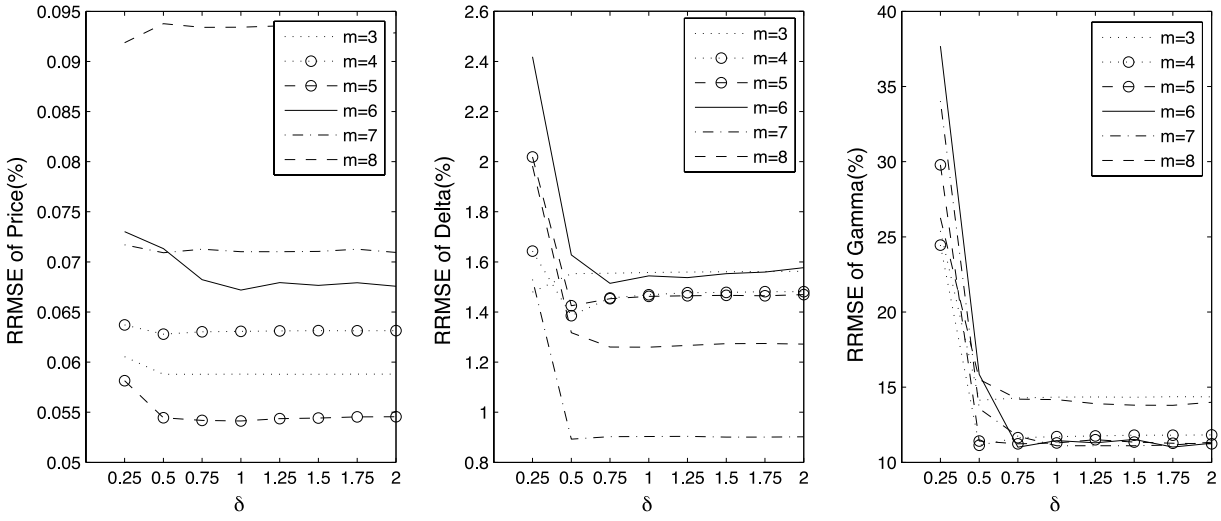

Fig. 2. RRMSE of price, delta, and gamma values for NIS in δ with 3×10^5 runs.

Table 1
DNS versus NIS (number of simulation runs $= 3 \times 10^5$).

m	Time-to maturity (M)	RRMSE: price (%)		RRMSE: delta (%)		RRMSE: gamma (%)	
		DNS	NIS	DNS	NIS	DNS	NIS
3	6.5	0.0587	0.0588	1.4961	1.5577	12.3621	14.3418
	7	0.0674	0.0677	1.3358	1.3989	15.3927	19.1838
4	6.5	0.0633	0.0631	1.3808	1.4690	15.3414	11.7054
	7	0.0837	0.0841	1.1409	1.2033	14.8832	12.3571
5	6.5	0.0556	0.0541	1.4475	1.4625	16.1546	11.3016
	7	0.0662	0.0644	1.3708	1.3874	15.9390	9.9429
6	6.5	0.0700	0.0675	1.7836	1.5437	18.7276	11.5289
	7	0.0891	0.0874	1.6232	1.5580	17.3683	9.5770
7	6.5	0.0714	0.0711	1.1042	0.9023	16.4597	11.0838
	7	0.1091	0.1057	1.4690	1.3992	18.3096	12.4892
8	6.5	0.0935	0.0937	1.4759	1.2569	19.8245	14.0457
	7	0.1358	0.1333	1.9572	1.6432	39.7813	20.2947

m and δ values, we found that $\delta = 1$ is a reasonable choice, and this means that the price estimates would stay within one standard error bound from the original estimates even after solving the optimization problem. For the rest of this paper, we use this parameter value. In Fig. 2, we summarize the RRMSE plots for price, delta, and gamma values along eight different δ values ($\delta = 0.25, 0.5, \ldots, 1.75, 2$). Here, time t is two weeks from t_5, and the total number of Monte Carlo runs that are used for estimates at θ_0 is fixed at 3×10^5.

3.2. Comparison of NIS and DNS

In this subsection, we compare the performance of NIS with the original denoising scheme (DNS) introduced in Section 2.2. As we have done for NIS, we can discretize the function $u^{(k)}(\cdot)$ in (4) and (5). However, in the first case, (4) becomes a quadratically constrained quadratic program which requires more efforts to solve than a linearly constrained quadratic program. Thus, in this subsection, we focus on the second formulation which is quite easy to solve and thus the workloads of (5) and (7) should be comparable. The discretized version of this program can be written as

$$\min_{\{u_i\}_{i=-m}^{m}} \sum_{i=-m+2}^{m-2} \left(u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2}\right)^2 + \rho \sum_{i=-m}^{m} (u_i - \hat{\alpha}_n(\theta_i))^2. \tag{10}$$

Obviously, we need to specify the parameter ρ in this formulation. Fig. 3 exhibits the behaviors of estimates as ρ changes from value 0 to 0.5. Here, the time t is again two weeks before t_5 and the budget is set to be 3×10^5 simulation runs. However, as seen from graphs, we cannot pick a single ρ value with which the DNS works well for all m values at the same time. Hence, rather than fixing ρ, we calculated the smallest RRMSE from DNS after applying various ρ values in each cell of Table 1. In the table, we observe that the RRMSE of DNS and NIS are quite similar in their magnitudes for price and delta. Except for $m = 3$, the gamma RRMSEs of NIS are about 50%–83% of those of DNS.

3.3. Comparison of NIS and FDS

The RRMSE values from FDS are reported in Table 2. Comparing Tables 1 and 2, we observe that the NIS increases the RRMSE of price estimates by at most 0.07% of the average of true prices with respect to the FDS. Note that this difference is very small when $m = 3$ where we put 10^5 runs at each of $\{\theta_{-1}, \theta_0, \theta_1\}$. Such difference increases as m increases, the reason being that we have a less number of runs at each point θ_i. Regarding delta, we see that the RRMSE of NIS is greater than that of FDS by at most 0.7%.

Despite such increases in errors for price and delta estimates, the benefit of NIS lies in its reduction of the RRMSE of gamma estimates. From two tables, it can be seen that the RRMSE is decreased by 20%–40% of the average of true gamma values. It is as if the number of Monte Carlo runs is increased by 10 times. Therefore, if such small changes in prices and deltas are acceptable

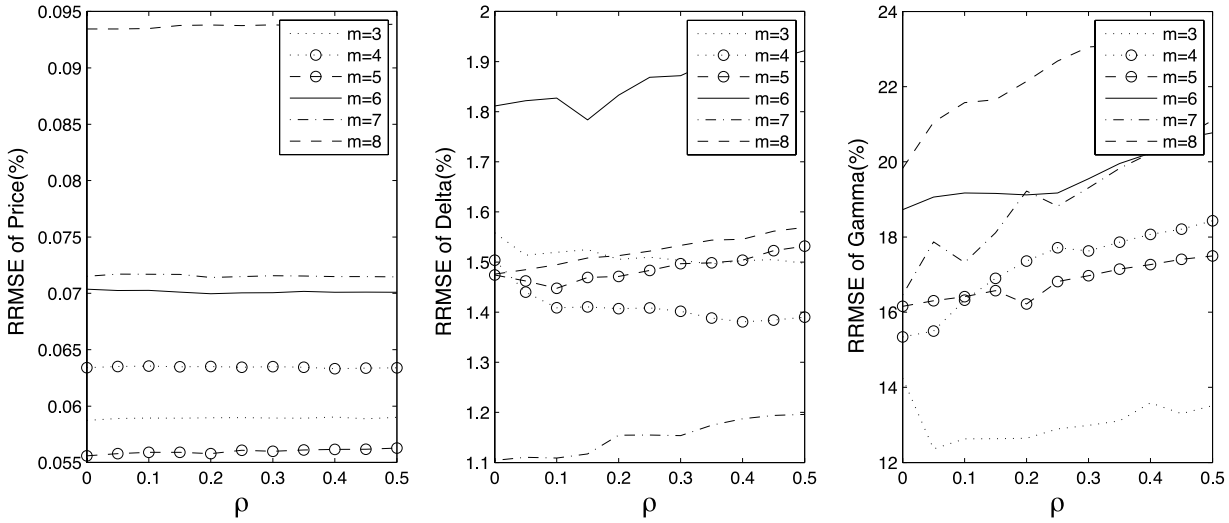

Fig. 3. RRMSE of price, delta, and gamma values for DNS in ρ with 3×10^5 runs.

Table 2
RRMSE of FDS (number of simulation runs $= 3 \times 10^5$).

Time-to-maturity (M)	RRMSE: price (%)	RRMSE: delta (%)	RRMSE: gamma (%)
6.5	0.0447	1.1265	43.0554
7	0.0504	0.9532	39.7813

Table 3
FDS versus NIS (number of simulation runs $= 3 \times 10^6$).

m	Time-to maturity (M)	RRMSE: price (%)		RRMSE: delta (%)		RRMSE: gamma (%)		
		FDS	NIS	FDS	NIS	FDS	NIS	Ratio
6	6.5	0.0144	0.0360	0.3206	0.5550	19.5747	4.8630	4.03
	7	0.0146	0.0299	0.4004	0.5981	17.5910	5.4502	3.23
	6.5	0.0146	0.0328	0.3191	0.5714	24.2696	5.9839	4.06
	7	0.0127	0.0303	0.2329	0.3605	14.6968	6.6813	2.2

(which indeed should be the case in practice), then this is a huge gain compared to the very little workload that the NIS requires. If we compare the results of FDS and DNS, then we can see that the RRMSE reduction of gamma from DNS is only 25%–33% effective compared to the NIS.

We also found that the case of $m = 6$ yields the best performance. For this number, we ran the same experiment but with the number of simulation runs 3×10^6. The results are reported in Table 3. Clearly, the above observations seem to be still valid. For a more graphical exposition of NIS, we plot the graphs of $\alpha''(\theta_0)$ for the range of θ_0 values that have been considered. In Fig. 4, the dashed curve is the result of our NIS and the solid one is from FDS. The level of fluctuation is greatly reduced.

4. Concluding remarks

In this paper, we looked at the problem of finding reliable sensitivity estimates of the performance function of complex stochastic systems. Motivating problems are the computations of the first and the second derivatives of the prices of financial products with respect to financial variables, which are an essential part of the risk management procedure. However, the proposed method should be applicable as long as the problem at hand has similar issues such as finite resources and unstable Monte Carlo sensitivity outputs from finite difference schemes. Our key solution approach is to treat the Monte Carlo sampling errors as noises, then to apply denoising techniques which have been well developed in the image processing or interpolation theory literature. The main differences from existing works are, first, the optimization problem is a quadratic programming in finite dimension which is rapidly solvable, and second, the information about the confidence intervals from original Monte Carlo estimates is utilized. In our numerical experiments, we used a highly complex financial product linked to a single stock. For this example, the performance of this heuristic method is demonstrated and compared to the finite difference scheme and a well known denoising technique. We observed that the additional gains in terms of root mean square errors are noticeable compared to the negligible computational efforts to solve the quadratic optimization problem. Even though our numerical tests are based on the geometric Brownian motion, the method is model-free and thus should be applicable to models with more complex dynamics such as stochastic volatility.

Several questions remain to be studied in the future. In the original near-interpolation problem, the constraints can include derivative information as well. In Monte Carlo experiments, it is possible to obtain lower-order derivative information by applying direct gradient estimation methods. Incorporating this possibility into our formulation is an ongoing research question. Also, with a constrained computational budget, we need to have a better idea about the trade-off between the values of m and n as well as the determination of other parameters δ and h. Any possible solution would depend on some detailed information about the true performance function such as differentiability or shape. Lastly, the application of this approach or its variants to simulation optimization, extending the line of works using splines, would be an interesting research direction as well.

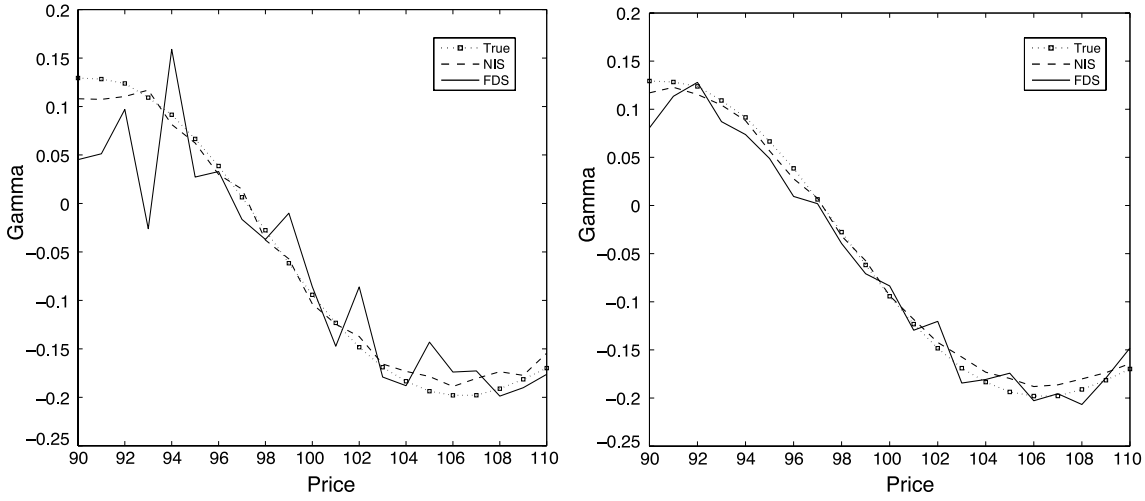

Fig. 4. Gamma estimates with the number of simulation runs (left) 3×10^5 (right) 3×10^6.

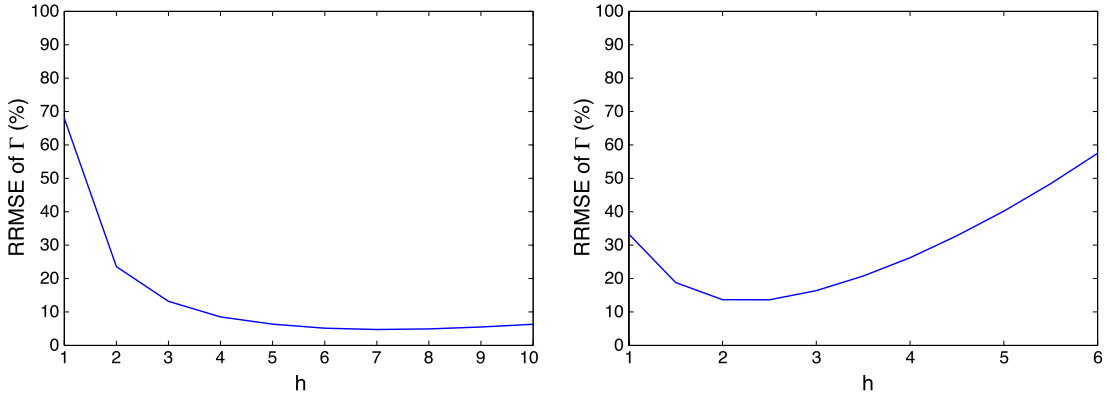

Fig. 5. RRMSE of digital option gamma when the FDS is used with $K = 120$, $T = 0.3$, $n = 100K$, and (left) $\sigma = 0.3$ (right) $\sigma = 0.1$.

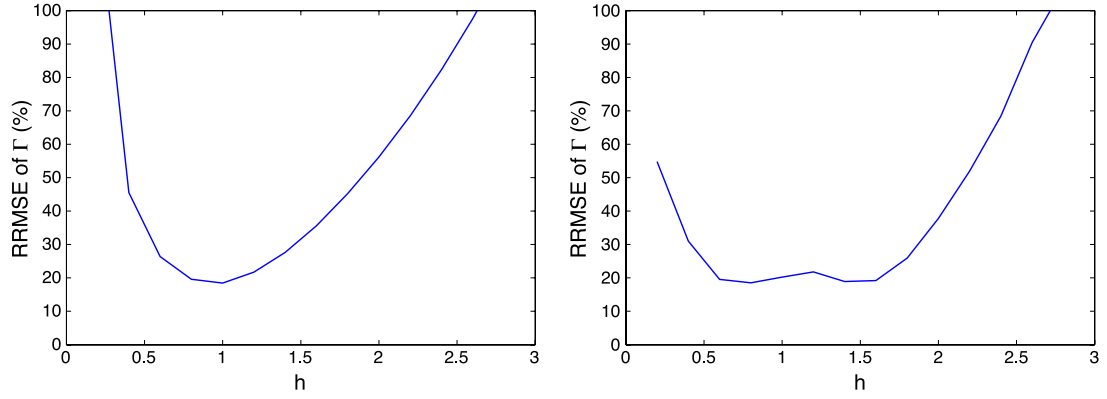

Fig. 6. RRMSE of digital option gamma with $K = 108$, $T = 0.05$, $\sigma = 0.1$, and (left) $n = 100K$ with FDS (right) $n = 20K$ with NIS.

Acknowledgments

Kang's work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (Ministry of Education, Science and Technology, MEST) (No. 2009-0068043, 2010-0015465). Kim's work was supported by Basic Science Research Program through the NRF funded by the MEST (No. 2011-0007651). Shin's work was supported by DAPA and ADD (No. UD110006MD).

Appendix

See Figs. 5 and 6.

References

[1] L.B.G. Andersen, V.V. Piterbarg, Interest Rate Modeling, vol. III, Atlantic Financial Press, New York, 2010.
[2] B. Ankenman, B.L. Nelson, J. Staum, Stochastic kriging for simulation metamodeling, Operations Research 58 (2010) 371–382.
[3] S. Asmussen, P. Glynn, Stochastic Simulation: Algorithms and Analysis, Springer-Verlag, New York, 2007.
[4] R.R. Barton, M. Meckesheimer, in: S.G. Henderson, B.L. Nelson (Eds.), Metamodel-Based Simulation Optimization, in: Handbooks in Operations Research and Management Science, vol. 13, Elsevier, 2006, (Chapter 18).
[5] D.G. Berghaus, J.P. Cannon, Obtaining derivatives from experimental data using smoothed-spline functions, Experimental Mechanics (1973) 38–42.
[6] T. Chan, J. Chen, Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods, SIAM, Philadelphia, 2005.

[7] M.C. Fu, in: S.G. Henderson, B.L. Nelson (Eds.), Gradient Estimation, in: Handbooks in Operations Research and Management Science, vol. 13, Elsevier, 2006, (Chapter 19).
[8] P. Glasserman, Monte Carlo Methods in Financial Engineering, Springer-Verlag, New York, 2004.
[9] P. Glynn, Optimization of stochastic systems via simulation, in: E. MacNair, K. Musselman, P. Heidelberger, (Eds.), Proceedings of the 1989 Winter Simulation Conference, pp. 90–105, 1989.
[10] M.F. Hutchinson, F. de Hoog, Smoothing noisy data with spline functions, Numerische Mathematik 47 (1985) 99–106.
[11] W. Kang, K. Kim, H. Shin, Fairing the gamma: an engineering approach to sensitivity estimation, Technical report, KAIST, 2010.
[12] S.N. Kersey, Best near-interpolation by curves: existence, SIAM Journal on Numerical Analysis 38 (2000) 1666–1675.
[13] S.N. Kersey, Near-interpolation, Numerische Mathematik 94 (2003) 523–540.
[14] G. Liu, L.J. Hong, Kernel estimation of the greeks for options with discontinuous payoffs, Operations Research 59 (2011) 96–108.
[15] C.H. Reinsch, Smoothing by spline functions II, Numerische Mathematik 16 (1971) 451–454.
[16] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Phisica D 60 (1992) 259–268.
[17] N. Sapidis, G. Farin, Automatic fairng algorithm for *b*-spline curves, Computer-Aided Design 22 (1990) 121–129.
[18] W.C.M. van Beers, J.P.C. Kelijnen, Kriging for interpolation in random simulation, Journal of Operational Research Society 54 (2003) 255–262.
[19] H.J. Woltring, On optimal smoothing and derivative estimation from noisy displacement data in biomechanics, Human Movement Science 4 (1985) 229–245.
[20] C. Zhang, P. Zhang, F. Cheng, Fairing spline curves and surface by minimizing energy, Computer-Aided Design 33 (2001) 913–923.