View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Research Papers in Economics

J Optim Theory Appl (2008) 136: 341–357
DOI 10.1007/s10957-007-9315-9

OpenAccess

Gradient Estimation Using Lagrange Interpolation Polynomials

R.C.M. Brekelmans · L.T. Driessen · H.J.M. Hamers · D. den Hertog

Published online: 31 January 2008
© The Author(s) 2008

Abstract We use Lagrange interpolation polynomials to obtain good gradient estimations. This is e.g. important for nonlinear programming solvers. As an error criterion, we take the mean squared error, which can be split up into a deterministic error and a stochastic error. We analyze these errors using N-times replicated Lagrange interpolation polynomials. We show that the mean squared error is of order $N^{-1+\frac{1}{2d}}$ if we replicate the Lagrange estimation procedure N times and use $2d$ evaluations in each replicate. As a result, the order of the mean squared error converges to N^{-1} if the number of evaluation points increases to infinity. Moreover, we show that our approach is also useful for deterministic functions in which numerical errors are involved. We provide also an optimal division between the number of gridpoints and replicates in case the number of evaluations is fixed. Further, it is shown that the estimation of the derivatives is more robust when the number of evaluation points is increased. Finally, test results show the practical use of the proposed method.

Keywords Gradient estimation · Lagrange interpolation · Mean squared error

Communicated by L.C.W. Dixon

We thank Jack Kleijnen, Gül Gürkan, and Peter Glynn for useful remarks on an earlier version of this paper. We thank Henk Norde for the proof of Lemma 2.2.

R.C.M. Brekelmans
Tilburg University, Tilburg, The Netherlands

L.T. Driessen
Philips Lightening, Eindhoven, The Netherlands

H.J.M. Hamers (✉) · D. den Hertog
Faculty of Economics and Business Administration, Department of Econometrics and OR, Tilburg University, Tilburg, The Netherlands
e-mail: h.j.m.hamers@uvt.nl

1 Introduction

In this paper we estimate the gradient $\nabla f(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$. The function f is not explicitly known and we cannot observe it exactly. All observations are the result of an evaluation of the function, which is subject to certain perturbations. These perturbations can be of stochastic nature (e.g. in discrete-event simulation) or numerical nature (e.g. deterministic simulation models are often noisy due to numerical errors).

Obviously, gradients play an important role in all kind of optimisation techniques. In most non-linear programming (NLP) codes first-order and even second-order derivatives are used. Sometimes these derivatives can be calculated symbolically: becoming more and more popular is automatic differentiation; (see Ref. [1]). Although this is becoming more and more popular, there is still much research going on in estimating and approximating gradients, especially for stochastic environments, since the stochastic gradient is often difficult or impossible to obtain in practice; (see Refs. [2, 3]). In the relatively recent book of (Ref. [4], pp. 153–156), many reasons and examples are given for which it is difficult or even inherently impossible to obtain the gradient via symbolic or automatic differentiation. Example classes given by Spall are generic parameter estimation for complex loss function, model-free feedback control system, and simulation-based optimization. Moreover, there are still many optimisation methods and solvers that use e.g. finite differencing to obtain a good approximation of the gradient; (see Refs. [4] or [5]).

Finite differences schemes have also been applied and analyzed for problems with stochastic functions. (Ref. [6]) were the first to describe the so-called stochastic (quasi)gradients; (see Ref. [7]). Methods based on stochastic quasi gradients are still subject of much research; for an overview see Ermoliev (Ref. [8]). It was shown that the estimation error by using optimal stepsizes is $O(N^{-\frac{1}{2}})$ for forward finite differencing and $O(N^{-\frac{2}{3}})$ for central finite differencing, in which N is the number of replicates; (see Refs. [9–12]). Moreover, Glynn (Ref. [9]) developed a gradient estimator based on m evaluations instead of 2. He showed that for $m \to \infty$ the convergence rate is $\mathcal{L}_m N^{-1}$. However, this scheme appeared to be impractical since the constant $\mathcal{L}_m$ is highly increasing in m.

In this paper we will extend the finite difference method. As in Glynn (Ref. [9]), instead of using two evaluations for each dimension, we use more $(2d)$ evaluations. We use Lagrange interpolation polynomials to obtain a good point estimate of the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$. More precisely, each partial derivative is estimated using an interpolating function $h(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{2d-1} x^{2d-1}$ that equals f in $2d$ evaluated points in one coordinate direction of f, with d a positive integer. Then $h'(0) = a_1$ is an estimate for this partial derivative. We consider the errors in the gradient estimation both due the deterministic approximation error ('lack of fit') and the presence of noise. We provide bounds for both the deterministic and the stochastic error. We show that the convergence rate is $N^{-1+\frac{1}{2d}}$, where N is the number of replicates of the Lagrange interpolation. This improves the above mentioned convergence rates for finite differencing when $d \geq 2$. Note that for $d = 1$, the corresponding interpolation function $h(x)$ boils down into a linear Lagrange interpolation function, corresponds to the central finite difference method. Compared

with Glynn (Ref. [9]), we observe that for $d \to \infty$ the convergence rate approaches $\mathcal{K}_d N^{-1}$, however, contrary to Glynn's method, our constants $\mathcal{K}_d$ are relatively small and bounded from above. Moreover, we provide some results in case we have a deterministic function in which numerical errors are involved. Given a fixed budget of evaluations, we provide an optimal division between the number of replicates (N) and the number of evaluations in such a replicate ($2d$). We also show that the estimation of the derivative is more robust against errors in the estimation of the parameters (variance, upper bound for the $(2d+1)$-th derivative), when the number of evaluation points is increased. The practical use of our method is shown by results on certain test problems.

This paper is organized as follows. Section 2 discusses the estimate of the gradient using Lagrange polynomials. The replicated Lagrange polynomials and the behavior of the mean squared error are considered in Sect. 3. In Sect. 4 we consider the error of the gradient estimation if the function is deterministic. The optimal division between the number of replicates and the number of evaluations in such a replicate, if there is a fixed budget of evaluations, is discussed in Sect. 5. In Sect. 6 we show that the estimation is more robust when more evaluation points are used. Section 7 reports on the results of several test problems.

2 Gradient Estimation of Stochastic Noisy Functions Using Lagrange Polynomials

In this section we estimate the gradient of a $2d$ times continuously differentiable function $f : I\mathbb{R}^n \to \mathbb{R}$ that is subject to stochastic noise using Lagrange interpolation polynomials. We provide an upper bound for the mean squared error.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function subjected to stochastic noise. Hence, for a fixed $y \in \mathbb{R}^n$, we observe

$$g(y) = f(y) + \epsilon(y). \tag{1}$$

The error term $\epsilon(y)$ represents a random component. In this paper we assume that the error terms in (1) are i.i.d. random errors with $E[\epsilon(y)] = 0$ and $V[\epsilon(y)] = \sigma^2$. This assumption implies that the error terms do not depend on y. Note that g can also be a computer simulation model.

We will approximate $\frac{\partial f(y)}{\partial y_i}$, $(i = 1, \ldots, n)$ in a point $y \in \mathbb{R}^n$ using the approximation function g, defined in (1). Without loss of generality we take $y = (0, \ldots, 0)^T$. For convenience, let $I = \{-d, \ldots, -1, 1, \ldots, d\}$. Next, the function g is evaluated in the gridpoints $y_v^i = vhe_i$ for all $v \in I$, where $h > 0$ and e_i is the i-th unit vector of dimension n. Observe that the gridpoints are equidistant on each side of zero and that this distance is given by h (see Fig. 1).

Now, take the interpolating polynomial $h_i : \mathbb{R} \to \mathbb{R}$ defined as

$$h_i(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{2d-1} x^{2d-1}, \tag{2}$$

that is exact in the evaluated points, i.e., according to (1) it holds that

$$h_i(x_v^i) = g(y_v^i), \quad \text{for all } v \in I, \tag{3}$$

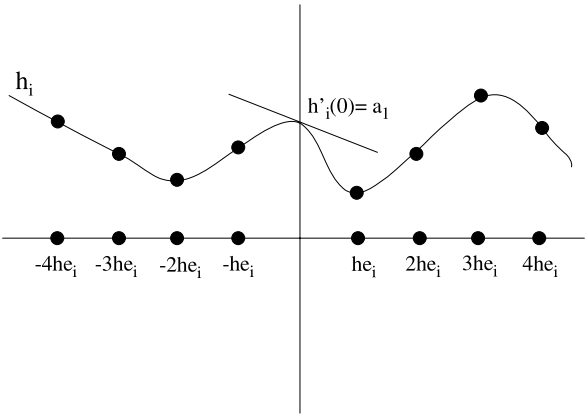

Fig. 1 Estimate of gradient using interpolating polynomial

where $x_v^i = e_i^T y_v^i$. Obviously, $h_i'(0) = a_1$ is an estimate of $\frac{\partial f(0)}{\partial y_i}$ (see Fig. 1).

Using the Lagrange functions $l_{v,i} : \mathbb{R} \to \mathbb{R}$ defined as

$$l_{v,i}(x) = \prod_{u \in I \setminus \{v\}} \frac{x - x_u^i}{x_v^i - x_u^i},$$

for any $v \in I$, (2) can be rewritten into

$$h_i(x) = \sum_{v \in I} l_{v,i}(x) g(y_v^i). \tag{4}$$

Hence, the derivative of $h_i(x)$ equals

$$h_i'(x) = \sum_{v \in I} \left[l_{v,i}(x) g(y_v^i) \sum_{u \in I \setminus \{v\}} \frac{1}{x - x_u^i} \right]. \tag{5}$$

From (5) it follows that the estimate of the partial derivative is a linear combination of the evaluations. Observe that the corresponding coefficients only depend on the $2d$ evaluation points. Table 1 provides the coefficients for $2d = 2, 4, 6, 8, 10$, respectively. The example in Sect. 6 will illustrate the use of the coefficients in Table 1.

Obviously, we are interested in the quality of $h_i'(0)$ as estimate of the partial derivative $\frac{\partial f(0)}{\partial y_i}$. Therefore, we define

$$h_{i,1}'(x) = \sum_{v \in I} \left[l_{v,i}(x) f(y_v^i) \sum_{u \in I \setminus \{v\}} \frac{1}{x - x_u^i} \right] \tag{6}$$

and

$$h_{i,2}'(x) = \sum_{v \in I} \left[l_{v,i}(x) \epsilon(y_v^i) \sum_{u \in I \setminus \{v\}} \frac{1}{x - x_u^i} \right]. \tag{7}$$

Table 1 Coefficients to generate estimate partial derivative

$2d=2$		$2d=4$		$2d=6$		$2d=8$		$2d=10$	
$v=ih$	coeff $g(y_v^i)$	$v=ih$	coeff $g(y_v^i)$	$v=ih$	coeff $g(y_v^i)$	$v=ih$	coeff $g(y_v^i)$	$v=ih$	coeff $g(y_v^i)$
−1	−0.5	−2	0.0833	−3	−0.0167	−4	0.0036	−5	−0.0008
1	0.5	−1	−0.6667	−2	0.1500	−3	−0.0381	−4	0.0099
		1	0.6667	−1	−0.7500	−2	0.2000	−3	−0.0595
		2	−0.0833	1	0.7500	−1	−0.8000	−2	0.2381
				2	−0.1500	1	0.8000	−1	−0.8333
				3	0.0167	2	−0.2000	1	0.8333
						3	0.0381	2	−0.2381
						4	−0.0036	3	0.0595
								4	−0.0099
								5	0.0008

It follows that

$$h_i'(x) = h_{i,1}'(x) + h_{i,2}'(x). \tag{8}$$

A well-known measure for the quality of the estimate of the partial derivative $\frac{\partial f(0)}{\partial y_i}$ by $h_i'(0)$ is the mean squared error:

$$E\left(h_i'(0) - \frac{\partial f(0)}{\partial y_i}\right)^2.$$

By defining the deterministic error

$$\left(error_d^{h_i'}\right)^2 = \left(h_{i,1}'(0) - \frac{\partial f(0)}{\partial y_i}\right)^2$$

and the stochastic error

$$\left(error_s^{h_i'}\right)^2 = E(h_{i,2}'(0))^2,$$

we get, because $E[\epsilon(x)] = 0$, that

$$E\left(h_i'(0) - \frac{\partial f(0)}{\partial y_i}\right)^2 = \left(error_d^{h_i'}\right)^2 + \left(error_s^{h_i'}\right)^2. \tag{9}$$

From (9), we learn that the mean squared error is the sum of the deterministic and the stochastic error. The following lemma provides an upper bound for the deterministic error.

Lemma 2.1 *For the Lagrange estimate, we have*

$$\left(error_d^{h_i'}\right)^2 \le M_{2d}^2 C_1^2(d) h^{4d-2},$$

where $C_1(d) = \frac{2}{(2d)!}\sum_{q=1}^{d}\left[q^{2d-1}\prod_{r\in I\setminus\{q\}}\frac{|r|}{|r-q|}\right]$ *and* M_{2d} *is an upper bound for the* $2d$ *order derivative of* f.

Proof For an upper bound of the deterministic error, we use the Kowalewski exact remainder for polynomial interpolation (cf. Ref. [13]):

$$f_i(x) - h_{i,1}(x) = \frac{1}{(2d-1)!}\sum_{v\in I} l_{v,i}(x)\int_{x_v^i}^{x}(x_v^i - t)^{2d-1} f^{2d}(t)dt, \tag{10}$$

where f_i is the slice function of f taking the ith component as variable. Taking the derivative to x on both sides of (10), substituting $x = 0$ and using $|f^{2d}(y)| \le M_{2d}$, we obtain

$$error_d^{h_i'} \le \frac{M_{2d}}{(2d)!}\sum_{v\in I}\left[\mid l'_{v,i}(0)\mid (x_v^i - 0)^{2d}\right],$$

where

$$l'_{v,i}(0) = \prod_{u\in I\setminus\{v\}}\left[\frac{0-x_u^i}{x_v^i - x_u^i}\right]\cdot\left[\sum_{u\in I\setminus\{v\}}\frac{1}{0-x_u^i}\right].$$

Because

$$\mid l'_{v,i}(0)\mid = \left|\prod_{u\in I\setminus\{v\}}\frac{0-x_u^i}{x_v^i - x_u^i}\right|\frac{1}{\mid x_v^i\mid}$$

and because $x_u^i = hu$, for all $u \in I$, we have

$$error_d^{h_i'} \le \frac{M_{2d}}{(2d)!}\,2\sum_{q=1}^{d}\left[(qh)^{2d}\prod_{r\in I\setminus\{q\}}\frac{\mid r\mid h}{\mid r-q\mid h}\cdot\frac{1}{qh}\right] = M_{2d}C_1(d)h^{2d-1},$$

which completes the proof. □

The next lemma shows, that $C_1(d)$ converges to zero. Hence, $error_d^{h_i'}$ will also converge to zero if M_{2d} is bounded

Lemma 2.2 *Let* $C_1(d) = \frac{2}{(2d)!}\sum_{q=1}^{d}\left[q^{2d-1}\prod_{r\in I\setminus\{q\}}\frac{|r|}{|r-q|}\right]$. *Then, the following two statements hold*:

(i) $C_1(d) \le 2d\left(\frac{3}{4-\epsilon}\right)^d$, *with* $\epsilon > 0$ *small*;

(ii) $C_1(d) \to 0$ *if* $d \to \infty$.

Proof It is sufficient to prove (i). First, observe that $C_1(d)$ can be rewritten into

$$C_1(d) = \frac{2(d!)^2}{(2d)!}\sum_{q=1}^{2d-1}\frac{q^{2d-1}}{(d+q)!(d-q)!}.$$

Let $a_d = \frac{(2d)!}{2(d!)^2}$. Then, $a_{d+1} = \frac{(d+1)^2}{(2d+2)(2d+1)} a_d$. Hence, there exists a small $\epsilon > 0$ such that $a_d \geq (4-\epsilon)a_{d+1}$ for large d. This implies that there is a constant c such that, for large d, we have

$$a_d \geq c(4-\epsilon)^d. \tag{11}$$

Let $b_d = \sum_{q=1}^{d} \frac{q^{2d-1}}{(d+q)!(d-q)!}$. Then, for each $q = 1, \ldots, d$, we have

$$\begin{aligned}
\frac{q^{2d-1}}{(d+q)!(d-q)!} &\leq q^{-1} \frac{q^{2d-1}}{\left(\frac{d+q}{3}\right)^{d+q}\left(\frac{d-q}{3}\right)^{d-q}} \\
&= 3^{2d}\left[\left(\frac{1+x}{x}\right)^{-1-x}\left(\frac{1-x}{x}\right)^{-1+x}\right]^d \\
&\leq 3^{2d} \cdot \left(\frac{1}{3}\right)^d = 3^d
\end{aligned}$$

where the first inequality follows from the Stirlings formula and $q^{-1} \leq 1$. In the second inequality, we use the fact that the continuous and concave function $z : (0, 1] \to \mathbb{R}$, defined by $z(x) = \left(\frac{1+x}{x}\right)^{-1-x}\left(\frac{1-x}{x}\right)^{-1+x}$, is upper bounded by $\frac{1}{3}$. Hence, we can conclude that

$$b_d \leq d3^d. \tag{12}$$

From (11) and (12), it follows that, for large d, we have

$$C_1(d) \leq 2d\left(\frac{3}{4-\epsilon}\right)^d. \qquad \square$$

The following lemma provides an expression for the stochastic error.

Lemma 2.3 *For the Lagrange estimate, we have*

$$\left(error_s^{h_i'}\right)^2 = C_2(d)\frac{\sigma^2}{h^2},$$

with $C_2(d) = 4\sum_{q=1}^{d}\left(\prod_{r\in I\setminus\{q\}} \frac{|r|}{|r-q|}\frac{1}{q}\right)^2$.

Proof We obtain

$$\begin{aligned}
\left(error_s^{h_i'}\right)^2 &= E(h_{i,2}'(0))^2 \\
&= E\left(\sum_{v\in I}\prod_{u\in I\setminus\{v\}} \frac{0-x_u^i}{x_v^i - x_u^i}\epsilon(x_v^i)\sum_{u\in I\setminus\{v\}}\frac{1}{0-x_u^i}\right)^2
\end{aligned}$$

$$= 4\frac{\sigma^2}{h^2}\sum_{q=1}^{d}\left(\prod_{r\in I\setminus\{q\}}\frac{|r|}{|r-q|}\frac{1}{q}\right)^2. \qquad \square$$

The next lemma shows that $C_2(d)$ is upper bounded.

Lemma 2.4 *Let* $C_2(d) = 4\sum_{q=1}^{d}\left(\prod_{r\in I\setminus\{q\}}\frac{|r|}{|r-q|}\frac{1}{q}\right)^2$. *Then,* $C_2(d) \le \frac{2}{3}\pi^2$ *for all* d.

Proof Observe that $C_2(d) = 4\sum_{q=1}^{2d-1}\left(\frac{(d!)^2}{(d+q)!(d-q)!}\frac{1}{q}\right)^2$. Because $\frac{(d!)^2}{(d+q)!(d-q)!} \le 1$ for all q, we have that $C_2(d) \le 4\sum_{q=1}^{2d-1}\frac{1}{q^2} \le 4\cdot\frac{1}{6}\pi^2 = \frac{2}{3}\pi^2$, which completes the proof. $\square$

3 Stochastic Noisy Functions and Replicates

In this section we estimate the gradient of a $2d$ continuous differentiable function $f:\mathbb{R}^n\to\mathbb{R}$ that is subject to stochastic noise by replicating the Lagrange estimation of the previous sections. We investigate the mean squared error.

The following lemmata with respect to the deterministic and stochastic error follow straightforward from Lemma 2.1 and Lemma 2.3, respectively. Obviously, the upper bound for the deterministic error will not change in case of replicates.

Lemma 3.1 *For the Lagrange estimation with* N *replicates, we have*

$$\left(error_d^{h_i'}\right)^2 \le M_{2d}^2 C_1^2(d) h^{4d-2}.$$

Evidently, the stochastic error in case of replicates is decreased by a factor N, the number of replicates.

Lemma 3.2 *For the Lagrange estimation with* N *replicates, we have*

$$\left(error_s^{h_i'}\right)^2 = C_2(d)\frac{\sigma^2}{Nh^2}.$$

In the final part of this section we determine the step size h that minimizes the mean squared error. From Lemma 3.1 and 3.2, it follows that the mean squared error, as a function of h, is upper bounded by

$$UMSE(h) = M_{2d}^2 C_1^2(d) h^{4d-2} + C_2(d)\frac{\sigma^2}{Nh^2}. \tag{13}$$

The following theorem states the optimal step size for the upper bound and shows that the minimum mean squared error converges to N^{-1} if d goes to infinity.

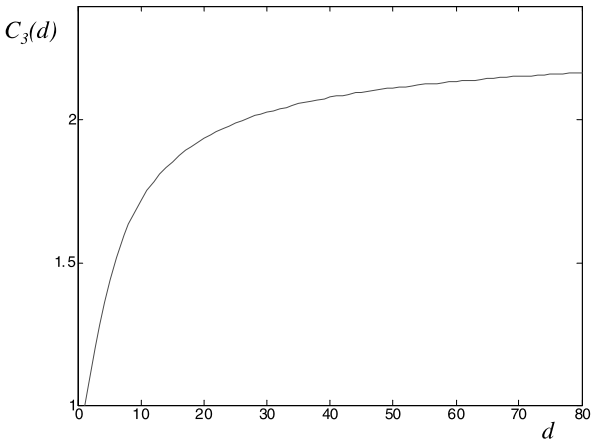

Fig. 2 The behavior of $C_3(d)$

Theorem 3.1 *Let UMSE*(h) *be defined as in* (13). *Then*:

(i) *The optimal stepsize* h^* *is* $h^* = (PN)^{\frac{-1}{4d}}$ *with* $P = \left(\frac{C_2(d)\sigma^2}{M_{2d}^2 C_1^2(d)(2d-1)}\right)^{-1}$.

(ii) *The minimum of UMSE is* $UMSE(h^*)^{\frac{-1}{4d}} = M_{2d}^{\frac{1}{d}}\sigma^{2-\frac{1}{d}}C_3(d)N^{-1+\frac{1}{2d}}$ *with* $C_3(d) = (C_1(d))^{\frac{1}{d}}(C_2(d))^{1-\frac{1}{2d}}(2d-1)^{\frac{1}{2d}}\left(\frac{2d}{2d-1}\right)$.

(iii) $C_3(d) \leq 0.9\pi^2$ *for large* d.

(iv) $UMSE(h^*) \to \mathcal{O}(N^{-1})$ *if* $d \to \infty$.

Proof The proof of (i) and (ii) is straightforward and (iv) results from (ii) and (iii). We will prove (iii). From Lemma 2.2 (i), it follows that $C_1(d)^{\frac{1}{d}} = (2d)^{\frac{1}{d}}\left(\frac{3}{4-\epsilon}\right)$. Because $(2d)^{\frac{1}{d}}$ converges to 1, we have that $(2d)^{\frac{1}{d}} \leq 1.1$ for large d and $\left(\frac{3}{4-\epsilon}\right) \leq 1$. Hence,

$$C_1(d)^{\frac{1}{d}} \leq 1.1 \tag{14}$$

if obviously it holds that

$$C_2(d)^{1-\frac{1}{2d}} \leq \frac{2}{3}\pi^2. \tag{15}$$

Because both $(2d-1)^{\frac{1}{2d}}$ and $\frac{2d}{2d-1}$ converge to 1 we have that both terms are upper bounded by 1.1 if d is large. Combining this observation with (14) and (15), we obtain

$$C_3(d) \leq 1.1 \cdot \frac{2}{3}\pi^2 \cdot 1.1 \cdot 1.1 < 0.9\pi^2. \qquad \square$$

In Fig. 2, the behavior of $C_3(d)$ is illustrated.

Table 2 provides the *UMSE* for some specific values of d. Observe that, already for small d, the best results in forward finite differencing ($\mathcal{O}(N^{-\frac{1}{2}})$) and central finite

Table 2 The *UMSE* for some values of d

d	*UMSE*
1	$1 \cdot M_2 \sigma N^{-\frac{1}{2}}$
2	$1.10(M_4)^{\frac{1}{2}} \sigma^{\frac{3}{2}} N^{-\frac{3}{4}}$
10	$1.68(M_{20})^{\frac{1}{10}} \sigma^{\frac{19}{10}} N^{-\frac{19}{20}}$
20	$1.94(M_{40})^{\frac{1}{20}} \sigma^{\frac{39}{20}} N^{-\frac{39}{40}}$
50	$2.12(M_{100})^{\frac{1}{50}} \sigma^{\frac{99}{50}} N^{-\frac{99}{100}}$

differencing ($\mathcal{O}(N^{-\frac{2}{3}})$) are improved. In fact, for $d = 1$, our result is identical to forward finite differencing.

It is interesting to compare the results in this table with the Glynn results (Ref. [9]). The order of convergence is the same, however, the constants explode for increasing d for his method. The corresponding constants for his method are e.g.: 31 for $d = 2$, 1.5×10^9 for $d = 10$ and 3.8×10^{20} for $d = 20$.

4 Numerically Noisy Functions

In this section, we estimate the gradient of a $2d$ times continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ that is subjected to numerical noise using Lagrange polynomials.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function that is subjected to numerical noise. Hence, for a fixed $y \in \mathbb{R}^n$, we observe that

$$g(y) = f(y) + \epsilon(y),$$

where $\epsilon(y)$ is the fixed, unknown numerical error. To estimate the gradient of f, we take the same approach as in Section 1.2. Let the functions h, $h'_{i,1}$ and $h'_{i,2}$ be defined as in (4), (6) and (7), respectively.

Then, the total error of the estimate of the partial derivative is equal to

$$\left| \frac{\partial f(0)}{\partial y_i} - h'_i(0) \right|. \tag{16}$$

We define the deterministic model error by

$$\left| \frac{\partial f(0)}{\partial y_i} - h'_{i,1}(0) \right|$$

and the numerical error by

$$\left| h'_{i,2}(0) \right|.$$

We get, by using (8), the following upper bound for the total error:

$$\left| \frac{\partial f(0)}{\partial y_i} - h'_i(0) \right| \leq \left| \frac{\partial f(0)}{\partial y_i} - h'_{i,1}(0) \right| + \left| h'_{i,2}(0) \right|. \tag{17}$$

Similarly to Sect. 2.1, we can provide upper bounds for the deterministic model and the numerical error. The proofs of the following two lemmas are omitted because they are almost identical to the proofs of Lemma 2.1 and 2.3, respectively.

Lemma 4.1 *For the Lagrange estimate, we have*

$$\left|\frac{\partial f(0)}{\partial y_i} - h'_{i,1}(0)\right| \leq M_{2d} C_1(d) h^{2d-1}.$$

Lemma 4.2 *For the Lagrange estimate, we have*

$$\left|h'_{i,2}(0)\right| \leq C_2(d)^{\frac{1}{2}} \frac{K}{h},$$

where K is an upper bound of ϵ.

In the final part of this section we determine the step size h that minimizes the total error. From Lemmas 4.1 and 4.2, it follows that the total error TE, as a function of h, is upper bounded by

$$UTE(h) = M_{2d} C_1(d) h^{2d-1} + C_2(d)^{\frac{1}{2}} \frac{K}{h}. \tag{18}$$

The next theorem provides the stepsize that minimizes the total error.

Theorem 4.1

(i) *The optimal step size h^** is $h^* = \left(\frac{C_2(d)^{\frac{1}{2}} K}{(2d-1)^{2d-1} M_{2d} C_1(d)}\right)^{-\frac{1}{2d}}$,
(ii) *The minimum of UTE is*

$$UTE(h^*) = M_{2d}^{1-\frac{1}{2d}} C_1(d)^{\frac{1}{2d}} C_2(d)^{\frac{1}{2}-\frac{1}{4d}} K^{1-\frac{1}{2d}} (2d-1)^{\frac{1}{2d}} \left(\frac{2d}{2d-1}\right).$$

The proof is straightforward and is therefore omitted.

Observe that, for the special case $d = 1$, that the result in Theorem 4.1 is similar to the result obtained in Gill et al. (Ref. [14], p. 340) for the forward finite-difference approximation.

5 Gridpoints versus Replicates

In this section we provide an optimal division between the number of gridpoints and replicates in case the number of evaluations is fixed.

Let B be the total number of evaluations available, N the number of replicates and $2d$ the number of evaluations per replicate. The problem to solve is the following:

$$\min \; UMSE(K N^{\frac{-1}{4d}}) = C_3(d) \left(\frac{M_{2d}}{\sigma}\right)^{\frac{1}{d}} \sigma^2 N^{-1+\frac{1}{2d}}, \tag{19}$$

Table 3 The optimal division between d and N at a fixed number of evaluations B

$\sigma = 1$, $M_{2d} = 1$				$\sigma = 0.1$, $M_{2d} = 1$			
B	d	error	N	B	d	error	N
$B \leq 23$	1	0.2879	6	$B \leq 3$	1	0.0349	1
$B \leq 803$	2	0.0207	134	$B \leq 12$	2	0.0148	2
$B \leq 21984$	3	0.0013	2748	$B \leq 240$	3	0.0012	30
$B \leq 386720$	4	0.0001	38672	$B \leq 3880$	4	0.0001	388
$B \leq 5461476$	5	9.85×10^{-6}	455123	$B \leq 54660$	5	9.85×10^{-6}	4555
$\sigma = 0.01$, $M_{2d} = 1$				$\sigma = 0.001$, $M_{2d} = 1$			
B	d	error	N	B	d	error	N
$B \leq 3$	1	0.0011	1	$B \leq 2376$	1	0.2901	594
$B \leq 12$	2	0.0003	2	$B \leq 79932$	2	0.0208	13322
$B \leq 24$	3	0.0001	3				
$B \leq 40$	4	0.0001	4				
$B \leq 600$	5	9.03×10^{-6}	50				

$$\text{s.t.} \quad B = 2dN,$$
$$d, N \text{ positive integers.}$$

In Table 3 we provide the optimal division between d and N for some values of B and a specific ratio of $\frac{M_{2d}}{\sigma}$.

In the upper left cell of Table 3, we have chosen $\sigma = 1$ and $M_{2d} = 1$. This cell illustrates that for a fixed budget $B = 24$ till $B = 803$ it is optimal to evaluate 4 ($d = 2$), points in each replicate. Obviously, in this case the number of replicates is determined by the quotient of the budget and 4. From $B = 804$ till $B = 21983$ it turns out that it is optimal to evaluate 6 points in each replicate. For example, if $B = 6000$ then we take $d = 3$, which equals 6 evaluations, and 1000 replicates. The other three cells of Table 3 present the results for different ratios of σ and M_{2d}. Observe that the error decreases if σ decreases. Moreover, the turning points to increase the number of gridpoints also decreases if σ decreases. For example, if $\sigma = 1$, then we turn to 6 gridpoints if $B = 804$, whereas if $\sigma = 0.01$ we already increase to 6 gridpoints if $B = 12$.

Observe that Table 3 suggests that $d = \mathcal{O}(\log(B))$ if $B \to \infty$. Indeed, this observation can be made plausible using the following arguments. There are two possible scenarios for the behavior of d if B becomes large. The first one is that $d \to \infty$. Because in this situation $C_3(d) \to c_3$. Then, a good approximation for (19) is obtained if we solve the following relaxation:

$$\min \left(\frac{M_{2d}}{\sigma}\right)^{\frac{1}{d}} N^{-1+\frac{1}{2d}},$$
$$\text{s.t.} \quad B = 2dN.$$

The minimum is found by determining the stationary points of

$$\log\left[\left(\frac{B}{2d}\right)^{-1+\frac{1}{2d}}\left(\frac{M_{2d}}{\sigma}\right)^{\frac{1}{d}}\right],$$

which boils down to

$$\frac{1}{2d^2}\left[-\log\left(\frac{B}{2d}\right)-2\log\left(\frac{M_{2d}}{\sigma}\right)+2d-1\right]=0,$$

which shows that $d \approx k\log(B)$ for some constant k. Substituting this result in (19), we obtain

$$C_3(d)\left(\frac{M_{2d}}{\sigma}\right)^{\frac{1}{d}}\sigma^2\left(\frac{B}{2k\log(B)}\right)^{-1+\frac{1}{2k\log(B)}}. \tag{20}$$

Let $\beta(d)=C_3(d)\left(\frac{M_{2d}}{\sigma}\right)^{\frac{1}{d}}\sigma^2$. Then, (20) can be rewritten into

$$\beta(d)\left(\frac{B}{2k\log(B)}\right)^{-1+\frac{1}{2k\log(B)}}. \tag{21}$$

This would end the argumentation. However, it can be the case that $d\to d^*$. Then, there exists a $\hat{d}\le d^*$ such that the minimum of (19) is attained in $\hat{d}$ and equals

$$C_3(\hat{d})\left(\frac{M_{2\hat{d}}}{\sigma}\right)^{\frac{1}{\hat{d}}}\sigma^2\left(\frac{B}{2\hat{d}}\right)^{-1+\frac{1}{2\hat{d}}}. \tag{22}$$

Let $\alpha(\hat{d})=C_3(\hat{d})\left(\frac{M_{2\hat{d}}}{\sigma}\right)^{\frac{1}{\hat{d}}}\sigma^2$. Then, (22) is equal to

$$\alpha(\hat{d})\left(\frac{B}{2\hat{d}}\right)^{-1+\frac{1}{2\hat{d}}}. \tag{23}$$

Now, we can conclude that $d=\mathcal{O}(\log B)$ if the minimum of (19) is attained in the situation where $d\to\infty$. Hence, we are finished if we can show that the expression in (23) is larger than (21).

Observe that $\alpha(\hat{d})$ is a constant and $\beta(d)$ is bounded under the assumption that $M_{2d}\le a^{2d}$ for some constant $a>1$ and Theorem 3.3(iii). Hence, it is sufficient to prove that

$$\frac{\left(\frac{B}{2k\log(B)}\right)^{-1+\frac{1}{2k\log(B)}}}{\left(\frac{B}{2\hat{d}}\right)^{-1+\frac{1}{2\hat{d}}}}\to 0,\quad \text{if } B\to\infty. \tag{24}$$

Straightforward calculations yield that (24) can be rewritten in $v(B)w(B)$, where

$$v(B)=\left(\frac{B}{2k\log(B)}\right)^{\frac{1}{2k\log(B)}}$$

and

$$w(B) = \left(\frac{2k\log(B)}{2\hat{d}}\right)\left(\frac{2\hat{d}}{B}\right)^{\frac{1}{2\hat{d}}}.$$

Because $v(B) \to$ constant if $B \to \infty$ and $w(B) \to 0$ if $B \to \infty$, we have that $v(B)w(B) \to 0$ if $B \to \infty$. Hence, we can support the observation that if $B \to \infty$ then $d = \mathcal{O}(\log(B))$.

6 Practical Aspects

To obtain the values for the optimal step size, one has to estimate the unknown constants σ and M_{2d}. In this section we first show that the estimated gradient is not very sensitive with respect to these constants. This means that even poor estimates of these quantities do not affect the quality of the gradient too much. We even show that the estimation is less sensitive when more evaluation points are used.

To analyze the sensitivity of the estimated gradient with respect to the unknown constants, let us assume that our estimates for σ (and M_{2d}) are $\hat{\sigma} = \kappa\sigma$ (and $\widehat{M}_{2d} = \kappa M_{2d}$), with $\kappa > 0$, respectively. Moreover, let us define the relative error *rUMSE* as the quotient of the *UMSE* when the above mentioned estimates are used for one of the constants, and the *UMSE* when the optimal step size of h is used. For example,

$$rUMSE_{\widehat{\sigma}}(h^*) = \frac{UMSE_{\widehat{\sigma}(h^*)}}{UMSE_{\sigma}(h^*)},$$

where the subindex $\widehat{\sigma}$ indicates that the estimated value $\widehat{\sigma}$ is used instead of σ. A similar definition holds for the other estimated constant M_{2d}. The following theorem gives expressions for these relative errors.

Theorem 6.1 *For the relative errors, we have*

$$rUMSE_{\widehat{\sigma}} = \frac{\kappa^{2-\frac{1}{d}} + (2d-1)\kappa^{-\frac{1}{d}}}{2d} \quad \text{and} \quad rUMSE_{\widehat{M}_{2d}} = \frac{\kappa^{-2+\frac{1}{d}} + (2d-1)\kappa^{\frac{1}{d}}}{2d}.$$

Proof We first show the results when $\widehat{\sigma}$ is used. $UMSE_{\widehat{\sigma}}(h^*)$ is given in Theorem 3.3. Moreover, to calculate $UMSE(h^*)_{\sigma}$ we substitute the estimated optimal step size, i.e., h^* in which $\widehat{\sigma}$ is used instead of σ into the expression for the UMSE. After some tedious calculations we obtain the first part of the theorem. The second part can be obtained in a similar way. □

In Fig. 3 the relative errors with respect to M_{2d} and σ, respectively, are shown for several values of d.

From Fig. 3 it is clear that the estimated gradients are not very sensitive with respect to the constants. For example an error of 20% for σ results into a 1.5% increase of the UMSE for $d = 2$ and an error of 50% for σ results into an 7% increase. Another important observation is that for $0 \leq \kappa \leq 2.5$ the relative errors are even decreasing

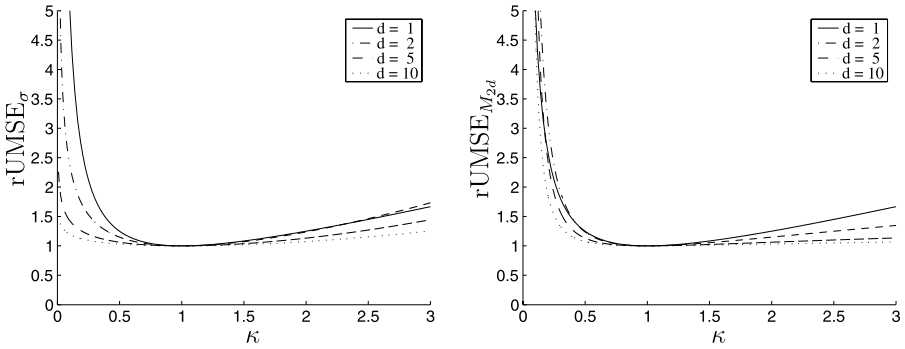

Fig. 3 The relative errors with respect to M_{2d} and σ

in d. This means that the estimate for the derivative is more robust if we use more points for the interpolation.

To estimate the constants M_{2d} we can use the same techniques as proposed for the classical finite difference schemes. For example in Gill et al. (Ref. [14], pp. 341–345), an algorithm is described in which M_2 is estimated by a second order finite difference scheme. Of course, as described in Gill et al. (Ref. [14]), such an estimate is not made in each iteration, since this will cost too many function evaluations, but only a few times. To estimate σ we can carry out replicates in a single point and use standard statistical methods. Again, such an estimate needs not to be made in each iteration.

7 Preliminary Test Results

For a first comparison between our method (Lagrange) and traditional central finite differencing (CFD), we defined a set of 7 one-dimensional test functions. We are interested in estimating the derivative $f_i'(0)$. For each test function f_i we observe the function g_i, $g_i(y) = f_i(y) + \epsilon(y)$, where $\epsilon(y)$ is normally distributed with expectation $\mu = 0$ and standard deviation σ which is varied from 0.0001 to 0.1, increasing with steps of a factor 10. All results are based on 1000 replications. The test functions are listed in Table 4 below, together with their derivative in 0.

Both for Lagrange and for CFD expressions for the optimal step size h exist. To calculate those optimal step sizes we would need to estimate an upper bound on the higher order derivatives (order 3 for CFD and order $2d$ for Lagrange). In these tests we deduced the optimal step size for both methods by experimental grid search: we took the step size for which the average estimation error over 1000 experiments was smallest. For Lagrange we considered combinations of simulation budget $M = \{4, 16, 32, 128, 1024\}$ and $d = \{1, 2, 4, 8, 16\}$.

Table 5[1] shows the results for test function 1. All calculations have been carried out in double precision. The shown errors are absolute deviations from the real derivative. When errors become small, machine precision starts to play a role. Those cases

[1] Tables 5–11 can be found at: http://center.uvt.nl/staff/hamers/publications.html/tables5-11.pdf.

Table 4 Test functions

i	f_i	$f_i'(0)$
1	$-1+e^{y}$	1
2	$-1+e^{3y}$	3
3	$\frac{e^{y}-e^{-y}}{2}$	1
4	$\cos(4(y-\frac{\pi}{8}))$	4
5	$y^4-y^3+100(1-y)^2$	-200
6	$(e^{y+1}-1)^2+(\frac{1}{\sqrt{1+(y+1)^2}}-1)^2$	$2e^2-2e-\frac{1}{2}+\frac{1}{\sqrt{2}}$
7	$\frac{\sin(24y-\frac{\pi}{8})}{12}+y$	$2\cos(\frac{-\pi}{8})+1$

have been marked with '$<$ mp'. For $\sigma=0$ we only included the results for $B=32$, and set the number of replications for both methods equal to 1. As there is no noise, results for other budgets and replications are exactly the same. The results for Lagrange with $d=1$ are exactly the same as for CFD, as the seed of the random generator has been set such that the same noise realizations occur for Lagrange and CFD. Differences for CFD between lines with the same values for σ and B are the result of different noise realizations. The error quotient for the best choice of d for the same σ and B is printed in italics. These error are most frequent larger than 1, indicating that Lagrange outperforms CFD. For higher noise levels the difference between Lagrange and CFD is smaller and the best results for Lagrange occur for low values of d. This is explained by the fact that for high noise levels the need for replication increases and the Lagrange method moves towards the CFD method by choosing a low value of d.

For the other 6 test functions, we only included the results for the best choice of d for a given B. Tables 6 to 11 summarize our findings. Test function 5 draws attention, as Lagrange outperforms CFD to a much greater extend here than in the other test functions. This is not surprising because the fifth test function is polynomial. For test function 7 for $\sigma=0.1$ Lagrange almost always chooses $d=1$ and boils down to CFD.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Griewank, A.: On automatic differentiation. In: Iri, M., Tanabe, K. (eds.) Mathematical Programming, pp. 83–107. KTK, Tokyo (1989)
2. He, Y., Fu, M.C., Marcus, S.I.: Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization. IEEE Trans. Autom. Control **48**, 1459–1463 (2003)
3. Kim, J., Bates, D.G., Postlethwaite, I.: Nonlinear robust performance analysis using complex-step gradient approximation. Automatica **42**, 177–182 (2006)
4. Spall, J.C.: Introduction to Stochastic Search and Optimization—Estimation, Simulation and Control. Wiley-Interscience Series in Discrete Mathematics and Optimization. Willey-Interscience, Hoboken (2003)
5. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. MPS-SIAM Series on Optimization, Philadelphia (2000)

6. Kiefer, J., Wolfowitz, J.: Stochastic estimation of a regression function. Ann. Math. Stat. **23**, 462–466 (1952)
7. Blum, J.R.: Multidimensional stochastic approximation methods. Ann. Math. Stat. **25**, 737–744 (1954)
8. Ermoliev, Y.: Stochastic quasigradients methods. In: Ermoliev, Y., Wets, R.J.-B. (eds.) Numerical Techniques for Stochastic Optimization. Springer (1980), Chap. 6
9. Glynn, P.W.: Optimization of stochastic systems via simulation. In: MacNair, E.A., et al. (eds.) Proceedings of the 1989 Winter Simulation Conference, pp. 90–105 (1989)
10. Zazanis, M.A., Suri, R.: Comparison of perturbation analysis with conventional sensitivity estimates for stochastic systems. Oper. Res. **41**, 694–703 (1993)
11. L'Ecuyer, P., Perron, G.: On the convergence rates of IPA and FDC derivative estimators for finite-horizon stochastic systems. Oper. Res. **42**, 643–656 (1994)
12. L'Ecuyer, P.: An overview of derivative estimation. In: Nelson, B.L., et al.: (eds.) Proceedings of the 1991 Winter Simulation Conference, pp. 207–217 (1991)
13. Davis, P.J.: Interpolation and Approximation. Dover, New York (1975)
14. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)