SIAM J. Numer. Anal.
Vol. 50, No. 3, pp. 1581–1603

NUMERICAL DIFFERENTIATION OF APPROXIMATED FUNCTIONS WITH LIMITED ORDER-OF-ACCURACY DETERIORATION*

O. BRUNO† and D. HOCH†

Abstract. We consider the problem of numerical differentiation of a function f from approximate or noisy values of f on a discrete set of points; such discrete approximate data may result from a numerical calculation (such as a finite element or finite difference solution of a partial differential equation), from experimental measurements, or, generally, from an estimate of some sort. In some such cases it is useful to guarantee that orders of accuracy are not degraded: assuming the approximating values of the function are known with an accuracy of order $\mathcal{O}(h^r)$, where h is the mesh size, an accuracy of $\mathcal{O}(h^r)$ is desired in the value of the derivatives of f. Differentiation of interpolating polynomials does not achieve this goal since, as shown in this text, n-fold differentiation of an interpolating polynomial of any degree $\geq (r-1)$ obtained from function values containing errors of order $\mathcal{O}(h^r)$ generally gives rise to derivative errors of order $\mathcal{O}(h^{r-n})$; other existing differentiation algorithms suffer from similar degradations in the order of accuracy. In this paper we present a new algorithm, the LDC method (low degree Chebyshev), which, using noisy function values of a function f on a (possibly irregular) grid, produces approximate values of derivatives $f^{(n)}$ $(n = 1, 2\dots)$ with limited loss in the order of accuracy. For example, for (possibly nonsmooth) $\mathcal{O}(h^r)$ errors in the values of an underlying infinitely differentiable function, the LDC loss in the order of accuracy is "vanishingly small": derivatives of smooth functions are approximated by the LDC algorithm with an accuracy of order $\mathcal{O}(h^{r'})$ for all $r' < r$. The algorithm is very fast and simple; a variety of numerical results we present illustrate the theory and demonstrate the efficiency of the proposed methodology.

Key words. numerical differentiation, Chebyshev approximation, differentiation of approximated functions

AMS subject classifications. 65D25, 45D05, 35R25

DOI. 10.1137/100805807

1. Introduction.

The problem of evaluating numerical derivatives from approximate data arises in a wide range of areas of numerical analysis and scientific computing, including image processing, solution of integral and differential equations, and parameter identification [5, 7, 10, 11, 13, 14]. In some such cases it is necessary to guarantee that orders of accuracy are not degraded: assuming the approximating values of the function are known with an accuracy of order $\mathcal{O}(h^r)$, where h is the mesh size, an accuracy of $\mathcal{O}(h^r)$ is desired in the value of the derivatives of f. (The particular problem motivating this work, for example, concerns an algorithm for fast evaluation of exact nonreflecting boundary conditions for finite element and finite difference solutions of hyperbolic partial differential equations (PDEs) [5], which would suffer a degradation in the order of the solution accuracy unless certain normal derivatives are evaluated with an order of accuracy equal to that inherent in the undifferentiated solution itself.) Differentiation of interpolating polynomials does not achieve this goal since, as shown in Appendix A and demonstrated in Figure 2, n-fold differentiation of an interpolating polynomial of any degree $\geq (r-1)$ obtained from function val-

---

[*] Received by the editors August 18, 2010; accepted for publication (in revised form) February 21, 2012; published electronically June 19, 2012. This work was supported by the Air Force Office of Scientific Research and the National Science Foundation.

http://www.siam.org/journals/sinum/50-3/80580.html

[†] Applied and Computational Mathematics, Caltech, Pasadena, CA 91125 (obruno@caltech.edu, dhoch@acm.caltech.edu).

ues containing errors of order $\mathcal{O}(h^r)$ generally gives rise to derivative errors of order $\mathcal{O}(h^{r-n})$. Other existing differentiation algorithms suffer from similar degradation in the order of accuracy. In this paper we present a new low degree Chebyshev (LDC) differentiation algorithm. The name reflects the fact that the method relies mainly on use of Chebyshev polynomials of adequately low degree.) Using noisy function values of f on an arbitrary grid, the LDC differentiation algorithm produces approximate values of the derivatives $f^{(n)}$ $(n = 1, 2 \dots)$ with limited loss in the order of accuracy. For example, for (possibly nonsmooth) $\mathcal{O}(h^r)$ errors in the values of an underlying infinitely differentiable function, the LDC loss in the order of accuracy is "vanishingly small": derivatives of smooth functions are approximated by the LDC method with an accuracy of order $\mathcal{O}(h^{r'})$ for all $r' < r$.

Previous work on reconstruction of numerical derivatives from scattered noisy data [2, 9, 12, 13, 19, 20, 21, 22] has focused on two main approaches: use of finite differences on one hand and regularization on the other. Much of the work in this area has been concerned with stability, seeking mainly to eliminate large derivative errors that arise as two function values $f(x_1)$ and $f(x_2)$ with large errors occur at points x_1 and x_2 that lie very close to each other. In [21, 22], for example, an approach (with theoretical error bounds in [22]) is presented which seeks to minimize accuracy degradation by means of a strategy based on Tikhonov regularization. As it happens, however, previous contributions have not demonstrated a capability to produce derivatives, even of the first order, without order-of-accuracy deterioration. In the recent literature we find, for example, differentiation errors of order $h^{2/9}$ for mesh spacings of order h in the method [2], errors of the order of several percent even for machine accurate data in the contribution [9], and errors of order $\delta^{1/2}$ for data containing errors of order δ in the contributions [12, 13].

The LDC algorithm presented in this paper offers a number of advantages over those introduced earlier: (1) it is extremely simple—it only requires simple polynomial interpolations and Chebyshev approximations, both of low order; (2) its theoretical analysis leads to an effective strategy for selection of input parameters; (3) unlike the approaches [21, 22], it does not require the solution of a systems of linear equations; (4) it preserves the accuracy order even for random errors; and, (5) it is, in practice, significantly more accurate than previous methods.

This paper is organized as follows. In section 2 we introduce the LDC method and we present error bounds and convergence rates that characterize its performance. In section 3 we then present a variety of numerical examples, including cases in one and higher dimensions and involving derivatives of various orders as well as errors arising from numerical and random sources; this includes a case in which errors result from the finite element solution of a PDE for which the solution gradient is computed with accuracy of higher order than that implicit in classical estimates [4]. A brief discussion in section 4 summarizes and concludes the presentation.

2. Theory.

We consider a function f which is either s times differentiable ($f \in C^s$), infinitely differentiable ($f \in C^\infty$), or real analytic ($f \in C^\omega$). Without loss of generality we assume f is defined in the interval $[-1, 1]$. (We focus on a one-dimensional problem in this section, although, as illustrated in section 3.3, problems in arbitrary dimensions may be treated by this method.) Let a set $\{f_\ell\}_{\ell=1}^{L}$ of L approximate discrete values of f be given in such a way that for a fixed constant C we have

$$|f(s_\ell) - f_\ell| \le Ch^r, \quad \ell = 1, \dots, L, \tag{1}$$

where $-1 \le s_\ell \le 1$ for $\ell = 1, \dots, L$ with $s_1 = -1$ and $s_L = 1$, and where the mesh size h is given by

$$h \equiv \max_{1<\ell\leq L} |s_\ell - s_{\ell-1}|. \tag{2}$$

Roughly speaking, our method is based on using the given data to produce adequate Chebyshev interpolations: in view of the extremely fast convergence of Chebyshev approximations, Chebyshev expansions of low order can be used with high accuracy, thereby avoiding enhancement of sampling errors as a result of the differentiation process. In what follows we thus consider low order Chebyshev approximations in the interval $[-1,1]$; the corresponding Chebyshev points for a Chebyshev expansion of order N will be denoted by

$$t_k = \cos\left(\frac{\pi(k-1/2)}{N}\right), \quad k = 1,\dots,N. \tag{3}$$

Our strategy calls for interpolation of the values f_ℓ onto the Chebyshev grid $\{t_k\}$ by polynomials of adequate degrees so that the accuracy order in the approximate function values is preserved. To guarantee convergence of order r for derivatives of f in presence of errors as in (1), for every point t_k we select an r-tuple $\mathcal{N}_k = (s_j,\dots,s_{j+r-1})$ of r consecutive points among the original mesh such that $s_j \leq t_k \leq s_{j+r-1}$ (see Remark 14 below for details on selection strategies we have used in practice to produce such "t_k-encompassing" r-tuples), and we obtain an approximation $\bar{f}_k$ of $f(t_k)$ by evaluating at t_k the polynomial $p_{(r-1)}$ of degree $(r-1)$ that interpolates the approximate values $\{f_\ell\}_{\ell=j}^{j+r-1}$ on the mesh $\mathcal{N}_k$. Clearly, the values $\bar{f}_k$ thus obtained approximate the exact function f with an error of the order of h^r:

$$|f(t_k) - \bar{f}_k| \leq Ch^r, \quad k = 1,\dots,N. \tag{4}$$

Our method then proceeds by producing the Chebyshev interpolant $P_N(t)$ for $\{\bar{f}_k\}_{k=1}^N$, that is, the polynomial

$$P_N(t) = {\sum_{j=0}^{N-1}}' \bar{c}_j T_j(t) \quad \text{satisfying} \quad P_N(t_k) = \bar{f}_k \quad \text{for} \quad k = 1,\dots,N, \tag{5}$$

where $T_j(t)$ is the Chebyshev polynomial of degree j; see, e.g., [3, 17, 18].

Remark 1. Here and throughout the paper we use the usual convention according to which the prime in the summation symbol indicates that the contribution of the $j=0$ term to the sum is $1/2$ of its value: $\sum' a_j = \frac{1}{2}a_0 + \sum_{j\neq 0} a_j$.

LDC algorithm prescription. Using the notation and conventions set forth in the previous paragraphs, the nth LDC approximate derivative of the function $f \in C^s[-1,1]$ is defined as the n-fold derivative of the approximants P_N defined by (5). The LDC algorithm results as the LDC derivative is evaluated by means of Chebyshev differentiation, as indicated in Remark 3. LDC differentiation of a function $f \in C^s[a,b]$ is treated via rescaling to the interval $[-1,1]$, i.e., by applying the LDC algorithm just introduced to the rescaled function $f((b-a)(x+1)/2+a)$, $x \in [-1,1]$. This is the complete prescription of the proposed methodology. The error analysis we present in what follows establishes the excellent properties of the algorithm, and it provides an indication of the values of N that should be used for optimal performance.

Remark 2. As is well known, the evaluation of the Chebyshev interpolant is not subject to instabilities that arise in regular polynomial interpolation: the Chebyshev coefficients in (5) can be obtained in a numerically stable manner by taking advantage of the discrete-orthogonality relations

$$\sum_{k=1}^{N} T_i(t_k)T_j(t_k) = T_{ij}^N, \quad 0 \le i,j \le N-1, \tag{6}$$

where

$$T_{ij}^N = \begin{cases} 0 & \text{for} \quad i \neq j, \quad 0 \le i,j \le N-1, \\ \frac{N}{2} & \text{for} \quad i = j \neq 0, \quad 0 \le i,j \le N-1, \\ N & \text{for} \quad i = j = 0. \end{cases} \tag{7}$$

Indeed, it follows from these relations and (5) that

$$\bar{c}_j = \frac{2}{N}\sum_{k=1}^{N} \bar{f}_k T_j(t_k). \tag{8}$$

Similarly, the continuous orthogonality property

$$\int_{-1}^{1} \frac{T_i(t)T_j(t)}{\sqrt{1-t^2}}\mathrm{d}t = \ell_{i,j}\delta_{ij}, \qquad \text{where} \qquad \ell_{i,j} = \begin{cases} 0 & \text{for} \quad i \neq j, \\ \frac{\pi}{2} & \text{for} \quad i = j \neq 0, \\ \pi & \text{for} \quad i = j = 0 \end{cases} \tag{9}$$

can be used to obtain the alternative formula

$$\bar{c}_j = \frac{2}{\pi}\int_{-1}^{1} \frac{P_N(t)T_j(t)}{\sqrt{1-t^2}}\mathrm{d}t, \tag{10}$$

which, although not part of our numerical implementation, is useful in the analysis of our algorithm.

Remark 3. Our algorithm evaluates the needed derivatives of Chebyshev expansions (5) by means of the expression [18]

$$P'_N(t) \equiv {\sum_{k=0}^{N-1}}' \bar{c}'_k T_k(t), \tag{11}$$

where the coefficients $\bar{c}'_k$ satisfy the descending recurrence relation

$$\begin{aligned} \bar{c}'_{N-1} &= 0, \\ \bar{c}'_{N-2} &= 2(N-1)\bar{c}_{N-1}, \\ \bar{c}'_{k-1} &= 2k\bar{c}_k + \bar{c}'_{k+1}, \qquad k = N-2, N-3, \dots, 1. \end{aligned} \tag{12}$$

Needed derivatives of order m ($m = 2, 3, \dots$), in turn, are obtained by iterated application of the $m = 1$ procedure (11)–(12).

Error analysis. To evaluate the approximation error associated with our algorithm we consider the Chebyshev expansion of the function f

$$f(t) = {\sum_{j=0}^{\infty}}' c_j T_j(t) \qquad \text{with} \qquad c_j = \frac{2}{\pi}\int_{-1}^{1} \frac{f(t)T_j(t)}{\sqrt{1-t^2}}\mathrm{d}t, \tag{13}$$

(see Remark 1) and we denote by f_N the partial sum

$$f_N(t) = {\sum_{j=0}^{N-1}}' c_j T_j(t). \tag{14}$$

The triangle inequality then yields

$$|f(t) - P_N(t)| \le |f(t) - f_N(t)| + |f_N(t) - P_N(t)|. \tag{15}$$

Under adequate smoothness assumptions on the function f, the first term on the right hand side of (15) is small [3]. In order to express the differences of the coefficients $c_i - \bar{c}_i$ associated with the second term in (15),

$$f_N(t) - P_N(t) = {\sum_{i=0}^{N-1}}' (c_i - \bar{c}_i) T_i(t), \tag{16}$$

in terms of the functions f and $P_N(t)$ we notice that at the Chebyshev points t_k we have

$$f(t_k) - P_N(t_k) = {\sum_{j=0}^{N-1}}' (c_j - \bar{c}_j) T_j(t_k) + \sum_{j=N}^{\infty} c_j T_j(t_k). \tag{17}$$

Multiplying both sides of (17) by $T_i(t_k)$, summing over the Chebyshev points, and using the orthogonality relation (6)–(7), which is valid for $\{i, j\} \in \{0, \dots, N-1\}$, we obtain

$$c_i - \bar{c}_i = \frac{2}{N} \sum_{k=1}^{N} (f(t_k) - P_N(t_k)) T_i(t_k) - \frac{2}{N} \sum_{j=N}^{\infty} c_j \sum_{k=1}^{N} T_i(t_k) T_j(t_k). \tag{18}$$

To continue with our discussion we now note that the discrete orthogonality property relationship (6)–(7) can be extended to arbitrary indices $\{i, j\}$—as detailed in the following lemma.

LEMMA 2.1. *For $i, j \in \mathbb{N} \cup \{0\}$ we have*

$$\frac{2}{N} \sum_{k=1}^{N} T_i(t_k) T_j(t_k) = (-1)^{\ell_{ij}^{+}} \chi_{ij}^{+} + (-1)^{\ell_{ij}^{-}} \chi_{ij}^{-}, \tag{19}$$

where

$$\ell_{ij}^{+} = \frac{i+j}{2N}, \qquad \ell_{ij}^{-} = \frac{i-j}{2N}, \tag{20}$$

$$\chi_{ij}^{+} = \begin{cases} 1 & \textit{if} \quad \ell_{ij}^{+} \in \mathbb{Z}, \\ 0 & \textit{otherwise}, \end{cases} \qquad \textit{and} \quad \chi_{ij}^{-} = \begin{cases} 1 & \textit{if} \quad \ell_{ij}^{-} \in \mathbb{Z}, \\ 0 & \textit{otherwise}. \end{cases} \tag{21}$$

Proof. Simple trigonometric manipulations give rise to the relations [17, p. 86]

$$\sum_{k=1}^{N} T_i(t_k) T_j(t_k) = \frac{1}{2} \sum_{k=1}^{N} \left[\cos\left(\left(k - \frac{1}{2}\right) \frac{i+j}{N} \pi \right) + \cos\left(\left(k - \frac{1}{2}\right) \frac{i-j}{N} \pi \right) \right]. \tag{22}$$

But, using the well known formula [17, p. 85]

$$\sum_{k=1}^{N} \cos\left(k-\frac{1}{2}\right)\theta = \frac{\sin N\theta}{2\sin\frac{1}{2}\theta}$$

and letting $\ell_j = \frac{j}{2N}$ for $j \in \mathbb{Z}$ we easily obtain

$$\sum_{k=1}^{N} \cos\left(\left(k-\frac{1}{2}\right)\frac{j}{N}\pi\right) = \begin{cases} (-1)^{\ell_j} N & \text{if} \quad \ell_j \in \mathbb{Z}, \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

for all integer values of j. Clearly, equations (22) and (23) imply (19), and the lemma follows. ☐

We now establish an LDC approximation theorem for function values. Although this result does not address the differentiation problem, it does provide a useful reference point building up to our main result, Theorem 2.4 below, on LDC differentiation. In accordance with the notation set forth at the beginning of section 2, in what follows we use the following convention: we say that "$f \in C^s[-1,1]$ for $s = \omega$" if $f \in C^\omega[-1,1]$, that is, if f is a real analytic function defined in the interval $[-1,1]$.

THEOREM 2.2. *Let $f \in C^s[-1,1]$ with $s \in \mathbb{N}$ or $s = \omega$, and let P_N be defined by equations* (4) *and* (5)*. Then, the absolute error of the approximation of f by P_N satisfies*

$$|f(t) - P_N(t)| \le 2C_3 h^r N + C_1 e^{-\sigma N} \qquad \text{for} \quad s = \omega, \tag{24}$$

$$|f(t) - P_N(t)| \le 2C_3 h^r N + C_2 N^{1-s} \qquad \text{for} \quad s \in \mathbb{N} \tag{25}$$

for certain constants $C_j > 0$ and $\sigma > 0$. (C_3 equals the constant C in (4)*.) If $f \in C^s[-1,1]$ is in addition piecewise C^∞, then an estimate sharper than* (25) *holds:*

$$|f(t) - P_N(t)| \le 2C_3 h^r N + C_4 N^{-(s+1)} \qquad \text{for } s \in \mathbb{N} \text{ and } f \text{ piecewise } C^\infty. \tag{26}$$

The value of N that minimizes the error bound (24) *for $s = \omega$ is given by*

$$N^{\text{opt}} = \text{INT}_1\left[\frac{1}{\sigma}\log\left(\frac{C_1\sigma}{2Ch^r}\right)\right], \tag{27}$$

while the corresponding optimal values of N for the error bounds (25) *and* (26) *in the case that $s \in \mathbb{N}$ are*

$$N^{\text{opt}} = \text{INT}_2\left[\left(\frac{(s-1)C_2}{2C_3h^r}\right)^{\frac{1}{s}}\right] \tag{28}$$

and

$$N^{\text{opt}} = \text{INT}_3\left[\left(\frac{(s+1)C_4}{2C_3h^r}\right)^{\frac{1}{s+2}}\right], \tag{29}$$

respectively. (Here, for $j = 1$ (resp., $j = 2$, $j = 3$) $\text{INT}_j[x]$ denotes one of the two integers N closest to x, namely, the one for which the right-hand-side of the inequality (24) *(resp.,* (25)*,* (26)*) is smallest.) In particular, using $N = N^{\text{opt}}$ given by* (27)*–*(29) *as appropriate depending on the degree of smoothness of f, it follows that the accuracy in the optimal approximation of f by P_N is essentially as high as $\mathcal{O}(h^r)$, provided that the underlying noiseless function f is sufficiently smooth; see Remarks* 7 *and* 9.

Proof. We consider the case $f \in C^\omega[-1,1]$ first. Using (18) and (19) we obtain

$$c_i - \bar{c}_i = \frac{2}{N}\sum_{k=1}^{N}(f(t_k) - P_N(t_k))T_i(t_k) - \sum_{j=N}^{\infty}(-1)^{\ell_{ij}^+}\chi_{ij}^+ c_j - \sum_{j=N}^{\infty}(-1)^{\ell_{ij}^-}\chi_{ij}^- c_j, \tag{30}$$

and thus in view of (16) we conclude that

$$f_N(t) - P_N(t) = \frac{2}{N}{\sum_{i=0}^{N-1}}{}'\sum_{k=1}^{N}\left(f(t_k) - P_N(t_k)\right)T_i(t_k)T_i(t) - R^+(t) - R^-(t), \tag{31}$$

where

$$R^\pm(t) = {\sum_{i=0}^{N-1}}{}'\left(\sum_{j=N}^{\infty}(-1)^{\ell_{ij}^\pm}\chi_{ij}^\pm c_j\right)T_i(t). \tag{32}$$

In view of the assumed analyticity of f it follows that (1) $|f(t) - f_N(t)|$ is exponentially small; (2) the term multiplying $T_i(t)$ in (32), which amounts to a sum of a subset of the $j \geq N$ exponentially small Chebyshev coefficients of the analytic function f, is itself exponentially small; and thus, (3) the remainder functions $R^\pm(t)$ are themselves exponentially small. We conclude that for certain constants C_1 and σ we have

$$|f(t) - f_N(t)| + |R^+(t)| + |R^-(t)| \leq C_1 e^{-\sigma N}. \tag{33}$$

Under the assumption (4), in turn, the first term on the right-hand side of (31) is bounded by $2C_3 N h^r$ (where C_3 equals the constant C in (4)) and, thus, taking into account (15), the estimate (24) follows.

The case $f \in C^s[-1,1]$ can be treated similarly, relying on the well known relation

$$|f(t) - f_N(t)| \leq \mathcal{O}(N^{1-s}) \quad , \quad -1 \leq t \leq 1 \tag{34}$$

(that is, valid for $f \in C^s[-1,1]$; see, e.g., [17, Th. 5.14]) instead of the exponential convergence that we exploited in the case of analytic f. It is useful to establish here the error bound (34). To do this, consider the 2π-periodic function $F(\theta) = f(\cos\theta)$, $F \in C^s[0,2\pi]$. Integrating by parts s times the integral $c_j = (1/\pi)\int_0^{2\pi} F(\theta)\cos(j\theta)d\theta$ we find that $c_j = \mathcal{O}(j^{-s})$. It follows that the term multiplying $T_i(t)$ in (32) is itself of order $\mathcal{O}(N^{-s})$ (as is easily checked, noting that for each i the quantities $\chi_{ij}^\pm$ vanish except for values of j that differ by multiples of $2N$) so that the remainder functions $R^\pm(t)$ are of order $\mathcal{O}(N^{1-s})$ and the error bound proof (34) follows.

As mentioned above, (25) can now be established as in the case of analytic f but using the relation (34) instead of the exponential convergence of f_N to f. The proof of the relation (26) can be obtained similarly, noting that integrating by parts $s+2$ times the integral $\int_0^{2\pi} F(\theta)\cos(j\theta)d\theta$ (where the integration by parts of the two highest orders are performed on subintervals, where $F \in C^{s+2}$) leads, in this case, to the estimates $c_j = \mathcal{O}(j^{-(s+2)})$ and thus $|f(t) - f_N(t)| = \mathcal{O}(N^{-(s+1)})$.

The relations (27) through (29), finally, result directly by differentiation with respect to N of the right-hand sides of equations (24) through (26), respectively. The proof is now complete. □

Remark 4. Equations (27) through (29) suggest that a small value of N *should* be used to produce the optimal Chebyshev interpolant P_N: use of values of N larger than those given by (27) and (28) generally gives rise to suboptimal approximations.

Remark 5. In the case of a nonanalytic C^∞ function f, taking into account that the parameter C_2 is a function of s, $C_2 = C_2(s)$, the optimal value N^{opt} can be determined in terms of the limit $\lim_{s\to\infty} C_2^{\frac{1}{s}}$, provided this limit exists, since $\lim_{s\to\infty}(\frac{s-1}{2Ch^r})^{\frac{1}{s}} = 1$.

Before turning to the main result of this section, Theorem 2.4, concerning the properties of the LDC differentiation algorithm we present an estimate we need in the proof of that theorem (that we were not able to find explicitly stated in the literature) on the rate at which the derivatives of the truncated Chebyshev expansion f_N approximate the corresponding derivatives of f.

LEMMA 2.3. *Let $f \in C^s[-1,1]$ with $s \in \mathbb{N}$ or $s = \omega$ be a function whose Chebyshev expansion is given by* (13), *and let f_N denote the truncated Nth order expansion* (14). *Then, if $n, s \in \mathbb{N}$, $2n \le s \pm 1$, we have*

$$|f^{(n)}(t) - f_N^{(n)}(t)| \le \frac{\tilde{C}_2}{N^{s-2n\pm 1}}, \tag{35}$$

where the + sign applies under the additional assumption that f is, say, piecewise C^∞, and where the constant $\tilde{C}_2 = \tilde{C}_2(n)$ is independent of N. For $s = \omega$ and for any $n \in \mathbb{N}$, in turn, $f_N^{(n)}$ approaches $f^{(n)}$ exponentially as fast as $N \to \infty$.

Proof. The result follows from the triangle inequality

$$|f^{(n)}(t) - f_N^{(n)}(t)| = \left|\sum_{j=N}^{\infty} c_j T_j^{(n)}(t)\right| \le \sum_{j=N}^{\infty} |c_j| \left|T_j^{(n)}(t)\right| \tag{36}$$

together with the well known estimate

$$\left|\frac{d^n T_j}{dt^n}\right| \le C_n j^{2n}, \tag{37}$$

where C_n is a constant independent of j, since, as noted in the proof of Theorem 2.2, we have $c_j = \mathcal{O}(j^{-s})$ for $f \in C^s[-1,1]$ with $s \in \mathbb{N}$ and $c_j = \mathcal{O}(j^{-(s+2)})$ if f is, in addition, piecewise smooth, whereas, as is well known, c_j is exponentially small for $f \in C^\omega[-1,1]$. For the sake of completeness, we note that the estimate (37) follows directly from the known relations

$$|T_j^{(n)}(t)| \le T_j^{(n)}(1) \quad \text{and} \quad T_j^{(n)}(1) = \prod_{m=0}^{n-1} \frac{j^2 - m^2}{2m+1}. \tag{38}$$

The first of these relations, in turn, results inductively from the identity [17, Eq. 2.49]

$$\frac{dT_j}{dt}(t) = 2j \sideset{}{'}\sum_{\substack{r=0 \\ j-r \text{ odd}}}^{j-1} T_r(t).$$

The second relation in (38), finally, follows from the expression

$$(1-t^2)\, T_j^{(n+1)}(t) - (2n-1)\, t\, T_j^{(n)}(t) + (j^2 - (n-1)^2)\, T_j^{(n-1)}(t) = 0 \tag{39}$$

that results as the classical ODE satisfied by T_j is differentiated $(n-1)$ times. ☐

The approximation properties of our differentiation algorithm are detailed in the following theorem.

Theorem 2.4. *Let $f \in C^s[-1,1]$ with $s \in \mathbb{N}$ or $s = \omega$, let $n \in \mathbb{N}$, and let P_N be defined by equations* (4) *and* (5)*. Then, the absolute error of the approximation of $f^{(n)}$ by $P_N^{(n)}$ satisfies*

$$|f^{(n)}(t) - P_N^{(n)}(t)| \le C_3(n)h^r N^{2n+1} + C_1(n)e^{-\sigma N} \qquad \textit{if} \quad s = \omega, \tag{40}$$

$$|f^{(n)}(t) - P_N^{(n)}(t)| \le C_3(n)h^r N^{2n+1} + C_2(n)N^{2n+1-s} \quad \textit{if} \quad s \in \mathbb{N} \tag{41}$$

for certain constants $C_j > 0$ and $\sigma > 0$. If $f \in C^s[-1,1]$ is in addition piecewise C^∞, then an estimate sharper than (41) *holds:*

$$|f^{(n)}(t) - P_N^{(n)}(t)| \le C_3(n)h^r N^{2n+1} + C_4(n)N^{2n-1-s} \qquad \textit{for } s \in \mathbb{N} \textit{ and } f \textit{ piecewise } C^\infty. \tag{42}$$

The value of N that minimizes the $s = \omega$ error bound (40) *is given by*

$$N^{\mathrm{opt}} = \begin{cases} \mathrm{INT}_1\left[\frac{1}{\sigma}\log\left(\frac{\sigma C_1(0)}{C_3(0)h^r}\right)\right] & \textit{for} \quad n = 0, \\ \mathrm{INT}_1\left[\frac{2n}{\sigma}W\left(\frac{\sigma}{2n}\left(\frac{\sigma C_1(n)}{(2n+1)C_3(n)h^r}\right)^{\frac{1}{2n}}\right)\right] & \textit{for} \quad n \ge 1, \end{cases} \tag{43}$$

where $W = W(z)$ is the principal branch of the Lambert function (which for $z > 0$ is determined as the unique real-valued solution of the equation $W(z)e^{W(z)} = z$; see [6] *and Remark* 6*). The corresponding optimal values of N for the error bounds* (41) *and* (42) *are given by*

$$N^{\mathrm{opt}} = \mathrm{INT}_2\left[\left(\frac{C_2(n)(s-1-2n)}{(2n+1)C_3(n)h^r}\right)^{\frac{1}{s}}\right] \tag{44}$$

and

$$N^{\mathrm{opt}} = \mathrm{INT}_3\left[\left(\frac{C_4(n)(s+1-2n)}{(2n+1)C_3(n)h^r}\right)^{\frac{1}{s+2}}\right], \tag{45}$$

respectively. (Here, for $j = 1$ (resp., $j = 2$, $j = 3$) $\mathrm{INT}_j[x]$ denotes one of the two integers N closest to x, namely, the one for which the right-hand side of the inequality (40) *(resp.,* (41), (42)*) is smallest.)*

Proof. The proof of this theorem is similar to that of Theorem 2.2, and we thus only provide a brief sketch of some needed additions. In view of Lemma 2.3, the first term on the right-hand side of the inequality

$$|f^{(n)}(t) - P_N^{(n)}(t)| \le |f^{(n)}(t) - f_N^{(n)}(t)| + |f_N^{(n)}(t) - P_N^{(n)}(t)| \tag{46}$$

is bounded by $\tilde{C}_1(n)\exp(-\sigma N)$ if $s = \omega$ and by $\tilde{C}(n)/N^{s-2n\pm 1}$ if $s \in \mathbb{N}$ (for certain constants $\tilde{C}_1(n)$, σ, and $\tilde{C}(n)$), where the + sign applies under the additional assumption that f is piecewise C^∞. Using (31) and (37), in turn, we see that the second term in (46) is bounded by the sum of $2C_nCh^r\sum_{i=0}^{N-1} i^{2n}$ and corresponding bounds on $|\frac{d^n}{dt^n}R^\pm|$. The needed bounds for the latter quantities can be obtained easily from the well known estimates

$$|c_j| \le Ke^{-\sigma j} \quad \text{for} \quad s = \omega \quad \text{and} \quad |c_j| \le Kj^{-s} \quad \text{for} \quad s \in \mathbb{N}$$

and

$$|c_j| \le Kj^{-(s+2)} \quad \text{for} \quad s \in \mathbb{N} \quad \text{and } f \text{ piecewise } C^\infty$$

(that hold for adequately selected constants K) together with (37), taking into account that χ^+_{ij} (resp., χ^-_{ij}) vanishes whenever $i + j$ (resp., $i - j$) is not a multiple of $2N$. ☐

Remark 6. By considering the graph of We^W as a function of W, it is easily checked that the equation $W(z)e^{W(z)} = z$ does indeed determine a unique real-valued function $W = W(z)$ (the Lambert function used in (43)) for all positive values of z. In view of the relations $e^W < We^W < e^{(1+\varepsilon)W}$, which, for a given $\varepsilon > 0$ are valid for W large enough, we see that W satisfies $W(z) \sim \log(z)$ as $z \to +\infty$.

Remark 7. In view of Theorem 2.4 and Remark 6 it is easy to check that the accuracy in the optimal LDC approximation of $f^{(n)}$ (that is, the LDC approximation of optimal degrees (43), (44), or (45), as appropriate) is essentially as high as $\mathcal{O}(h^r)$. More precisely, the accuracy provided by the optimal LDC algorithm is better than $\mathcal{O}(h^{r-\eta})$ either for all $\eta > 0$ or for small $\eta > 0$, depending on the degree of smoothness of the underlying noiseless function f; see also Remark 9.

Remark 8. According to our "algorithm prescription" in section 2, LDC differentiation of a function $f \in C^s[a, b]$ is treated via rescaling to the interval $[-1, 1]$. For simplicity, Theorem 2.4, which presents the orders N^{opt} of the Chebyshev approximations that lead to optimally small error bounds, is stated for a function $f \in C^s[-1, 1]$. But, as is easily checked by rescaling, error bounds identical in form to those on the right-hand sides of (40) through (45) hold for LDC differentiation of an arbitrary function $f \in C^s[a, b]$ for any $a < b$ ($s \in \mathbb{N}$ or $s = \omega$).

Remark 9. Often the underlying noiseless functions whose approximate derivatives are sought are either analytic or s-times differentiable for a large value of s. In such cases, in view of the fast convergence to zero of either the exponential term in (40), or, as relevant, the large powers $(s - 2n \pm 1)$ of $1/N$ in (41) and (42) typically found in practice, one can expediently choose, through consideration of a few relevant numerical examples, a relatively small value of N leading to a machine-precision error contribution from the second terms on the right-hand sides of (40)–(42). This leads to the desired h^r convergence in the derivatives—at least until the machine-precision base error associated with the second error terms in these equations are reached. According to this strategy one fixes a convenient value of the parameter N as h is refined, which results in an error in the numerical derivative that is essentially of the order $\mathcal{O}(h^r)$. More rigorously, it is easy to check that use of the parameter N^{opt}, which varies very slowly with h, results in a derivative error of order $\mathcal{O}(h^r \log^{2n+1}(\frac{1}{h}))$ in the analytic case and $\mathcal{O}(h^{r-\eta})$, with arbitrarily small η, in the infinitely smooth case; cf. Remark 7.

Remark 10. The optimal Chebyshev orders N^{opt} put forth in Theorem 2.4 depend on the exponential parameter σ, the differentiability order s, and quotients of the error proportionality constants C_1, C_2, and C_3, which are generally difficult to obtain from direct analysis of the data at hand. Explicit estimates for these optimal parameters can nevertheless be obtained for each case, $s \in \mathbb{N}$ or $s = \omega$, provided the errors resulting for two fixed values of the mesh size h can be evaluated as functions of N—as is often the case, e.g., by obtaining trusted function values on the basis of fine meshes and preliminary h-convergence analysis. Indeed, if such an error estimation method is in place for a given problem and a given fixed mesh size h, the optimal number N^{opt} for the mesh size h can be obtained by "exhaustive evaluation" of errors, that is, quite simply, by evaluating errors for a range of values of N, starting with a low value (e.g., $N = 2$) and subsequently increasing it by one until no additional improvements are obtained. In light of the relation (43) (or (44), (45) for nonanalytic differentiable functions), this procedure needs to be performed only for two mesh sizes (say, the coarsest and finest mesh sizes $h_c = (b-a)/(L_c-1))$ and $h_f = (b-a)/(L_f-1)$, where L_c and L_f denote the number of mesh-points for the coarsest and finest data-meshes in the interval $[a,b]$ at which noisy data values f_ℓ are provided) in the range of mesh-sizes of interest. The (two) constants needed in each case can thereby be determined, and the optimal values N^{opt} for all other mesh sizes can thus be evaluated by means of the expressions provided in Theorem 2.4. Considering an analytic function f, for example, and denoting by $N_c = N_c^{\mathrm{opt}}$ and $N_f = N_f^{\mathrm{opt}}$ the optimal numbers of Chebyshev points for the mesh sizes h_f and h_c, respectively, the optimal value N^{opt} as a function of h is approximately given by

$$N^{\mathrm{opt}}(h) = \begin{cases} k_1 \log\left(k_2 h^{-r}\right) & \text{for} \quad n = 0, \\ \frac{n}{k_1} W\left(\frac{k_1}{n}\left(\frac{k_2}{h^r}\right)^{\frac{1}{2n}}\right) & \text{for} \quad n \geq 1, \end{cases} \tag{47}$$

where the constants k_1 and k_2 are given by

$$k_1 = \begin{cases} -\frac{1}{r}\frac{N_c - N_f}{\log(h_c) - \log(h_f)} & \text{for} \quad n = 0, \\ \frac{n}{N_c - N_f} \log\left(\frac{N_f}{N_c}\left(\frac{h_f}{h_c}\right)^{\frac{r}{2n}}\right) & \text{for} \quad n \geq 1, \end{cases} \tag{48}$$

and

$$k_2 = \begin{cases} h_c^r e^{\frac{N_c}{k_1}} & \text{for} \quad n = 0, \\ h_c^r \left(N_c e^{\frac{k_1 N_c}{n}}\right)^{2n} & \text{for} \quad n \geq 1. \end{cases} \tag{49}$$

3. Numerical examples.

The numerical results presented in this section demonstrate the properties of the LDC algorithm in a range of contexts, including evaluation of derivatives of various orders for cases in which the input data contains (i) errors that arise from interpolation by polynomials (sections 3.1 and 3.2), (ii) random errors (section 3.2), and (iii) errors resulting from a numerical solution of a PDE (section 3.3). In particular, in section 3.1.2 we demonstrate the dependence of the optimal differentiation parameter values on the size of the domain of definition of the given function; see also Remark 11.

Throughout this section, the performance of the LDC method is compared with that resulting from other approaches, including methods based on polynomial interpolation [16, pp. 288–294], algorithms based on use of radial basis functions (RBFs)—such as those introduced in [22] and references therein—as well as the finite element differentiation methods implicit in finite element PDE solvers [4]. The numerical results presented in this section demonstrate, in a varied set of examples, the practical realization of the theoretical accuracy predictions presented in section 2. As mentioned in the introduction, the error bounds for the LDC method improve significantly over the corresponding estimates for other algorithms (including, e.g., [2, 9, 12, 13]); accordingly, the numerical results in this section are considerably more accurate than those resulting from previous approaches.

We conclude the preliminaries of this section with two remarks concerning parameter selection and implementation details.

Remark 11. Most of the examples presented in sections 3.1 through 3.3, all of which concern (or are reduced to) differentiations of functions defined in a finite interval $[a,b]$ of the real line, are treated by means of Chebyshev expansions in the full interval of definition through rescaling to the interval $[-1,1]$, as prescribed in section 2. An alternative approach may be based on decomposition of the interval $[a,b]$ into a number of subintervals, in each one of which the LDC algorithm is applied; naturally, such domain decomposition strategies may be highly advantageous for functions containing multiscale features. The use of the LDC algorithm in a domain decomposition context is considered briefly in section 3.1.2.

Remark 12. Various fast algorithms associated with Chebyshev expansions are used as part of our implementations of the differentiation methodology introduced in section 2. In particular, in our implementations the fast cosine transform is used to evaluate the needed Chebyshev coefficients (8) efficiently, the derivative coefficients $\bar{c}'_k$ are obtained from the corresponding function coefficients by means of (12) and the evaluation of Chebyshev expansions such as (5) and (11) for functions and their derivatives is produced by means of Clenshaw's recurrence formula; see, e.g., [3] or [18] for details.

3.1. First order differentiation: Interpolation errors. In this section we demonstrate the properties of the LDC algorithm by means of test cases involving differentiation of the Gaussian function

$$f(x) = \exp\left(-(x-x_0)^2/\gamma^2\right) \tag{50}$$

from noisy values $\{\bar{f}_\ell\}$ of f on various grids τ_h in an interval $[a,b]$. For simplicity we use equispaced meshes $\tau_h = \{x_\ell = a + (\ell-1)h : 1 \le \ell \le L\}$ of mesh size $h = (b-a)/(L-1)$; the LDC algorithm exhibits similar qualities, of course, for non-equispaced grids. For the test cases considered in this section, the errors introduced in function values arise from interpolation, as detailed in Remark 13. As illustrated in sections 3.2 and 3.3, however, the LDC method is suitable for use in cases in which function values include other types of errors, such as, e.g., finite element errors or random errors.

Remark 13. To produce interpolatory noisy function values $\bar{f}_\ell$ of a given function f on a mesh τ_h in an interval $[a,b]$ we consider for integer values K ($1 < K < L$) equispaced grids $\tilde{\tau}_k = \{s_j = a + (j-1/2)k : j \in \mathbb{Z}\}$ of mesh size $k = (b-a)/(K-1)$ together with certain polynomials that interpolate f at points in $\tilde{\tau}_k$; noisy values of f on the grid τ_h are then produced through evaluation of the polynomial interpolants. In detail, the noisy values at a node $x_\ell \in \tau_h$ are obtained as follows: the r consecutive x_ℓ-encompassing nodes $\{s_j, \dots, s_{j+r-1}\}$ of the mesh $\tilde{\tau}_k$ are selected, $s_j \le x_\ell \le s_{j+r-1}$ (see Remark 14), and the standard interpolating polynomial $p_{(r-1)}$ of degree $(r-1)$ that interpolates f at the points $\{s_j, \dots, s_{j+r-1}\}$ is evaluated at the point x_ℓ (see Remark 15). The data thus obtained on the grid τ_h approximates the corresponding values f with an error of the order $\mathcal{O}(h^r)$ provided values K proportional to L (and, thus, k approximately proportional to h) are used.

Remark 14. Given a mesh $\tau = \{s_1, s_1 \dots, s_q\}$, an integer $r \le q$, and a real number x satisfying $s_1 \le x \le s_q$, we consider the r-tuples R_ℓ ($r = 1, \dots, m$) defined by $R_1 = (s_1, \dots, s_r)$, $R_2 = (s_r, \dots, s_{2r-1}), \dots$, $R_\ell = (s_{\ell r-(\ell-1)}, \dots, s_{(\ell+1)r-\ell})$, where ℓ is the largest positive integer for which $(\ell+1)r - \ell \le q$. Further, if $(\ell+1)r - \ell < q$ we also consider the set $R_{\ell+1} = (s_{q-r+1}, \dots, s_q)$. With these conventions, the x-encompassing r-tuple of points from the given q-point mesh τ is defined as the r-tuple R_j with minimum value of j for which $\min R_j \le x \le \max R_j$.

Remark 15. In all the examples considered in this text, the evaluation of point values of Chebyshev interpolants (expansions) and their derivatives was performed as indicated in Remark 12. For all other polynomial interpolants p_d used, point values $p_d(x)$ of p_d were obtained by means of the classical Neville algorithm [18], while, when needed for comparison purposes, point values $p_d^{(n)}(x)$ of the derivatives of p_d of various orders n were obtained by the corresponding generalization of the Neville algorithm presented in [15].

In the present section we present first order differentiation results produced by the LDC algorithm on the basis of $\mathcal{O}(h^2)$-approximate values $\{\bar{f}_\ell\}$ (obtained as indicated in Remark 13 with $r = 2$) on the mesh τ_h. In detail, we display the LDC differentiation errors $\varepsilon^{2,(n)}_{\mathrm{LDC}}$ and $\varepsilon^{\infty,(n)}_{\mathrm{LDC}}$ in the L^2 and L^∞ norms, which are given by

$$(51)\quad \varepsilon^{2,(n)}_{\mathrm{LDC}} \equiv \sqrt{h_s \sum_{x\in\tau^s} \left(P^{(n)}_N(x) - f^{(n)}(x)\right)^2}, \qquad \varepsilon^{\infty,(n)}_{\mathrm{LDC}} \equiv \max_{x\in\tau^s} |P^{(n)}_N(x) - f^{(n)}(x)|,$$

on a sampling mesh τ^s in the interval $[a, b]$ of sufficiently fine mesh size h_s. For comparison purposes, some of our figures also display the corresponding L^2 and L^∞ errors $\varepsilon^{2,(n)}_{\mathrm{LPI}(d)}$ and $\varepsilon^{\infty,(n)}_{\mathrm{LPI}(d)}$ that result as nth order derivatives of the ordinary interpolating polynomials p_d of degree d (local polynomial interpolants of degree d, LPI(d)) are used to approximate, as indicated in Remark 15, the corresponding nth derivatives of f:

$$(52)\quad \varepsilon^{2,(n)}_{\mathrm{LPI}(d)} \equiv \sqrt{h_s \sum_{x\in\tau^s} \left(p^{(n)}_d(x) - f^{(n)}(x)\right)^2}, \qquad \varepsilon^{\infty,(n)}_{\mathrm{LPI}(d)} \equiv \max_{x\in\tau^s} |p^{(n)}_d(x) - f^{(n)}(x)|.$$

Remark 16. As discussed in Appendix A and demonstrated in Figure 2, differentiation of LPIs in the presence of noise of order $\mathcal{O}(h^r)$ produces optimal results for LPIs of degree $d = (r - 1)$. Thus, in the presence of underlying function values with errors of the order of $\mathcal{O}(h^r)$ we write $\varepsilon^{2,(n)}_{\mathrm{LPI}}$ instead of $\varepsilon^{2,(n)}_{\mathrm{LPI}(r-1)}$ and $\varepsilon^{\infty,(n)}_{\mathrm{LPI}}$ instead of $\varepsilon^{\infty,(n)}_{\mathrm{LPI}(r-1)}$.

3.1.1. First order differentiation. For our first set of examples we consider the Gaussian function (50) with $\gamma = 0.5$ and $x_0 = 0.1$ in the interval $[0, 1]$, and we use $n = 1$ (first order differentiation), $r = 2$, and meshes τ_h containing $L = 3K$ points for various values of K. From the prescriptions above we clearly have $s_{j+1} - s_j = k = \frac{1}{K-1}$, $x_{\ell+1} - x_\ell = h = \frac{1}{L-1}$, and the approximate value $\bar{f}_\ell \sim f(x_\ell)$ is obtained by evaluating at x_ℓ the linear polynomial that interpolates f at the points s_j and s_{j+1} for which $x_\ell \in [s_j, s_{j+1}]$. Clearly, the values $\{\bar{f}_\ell\}^L_{\ell=1}$ approximate the corresponding values $f(x_\ell)$ of the analytic function f with errors of the order $\mathcal{O}(h^2)$ at all points $x_\ell \in \tau_h$. From the data $\{\bar{f}_\ell\}^L_{\ell=1}$ we construct an LDC approximation to the derivative of f, which, according to the theory presented in section 2, should converge to $f'(x)$ with an error of the order of $h^2 \log^3(\frac{1}{h})$—essentially, with order $\mathcal{O}(h^2)$.

The expressions (47) through (49) for the optimal parameters N^{opt} for a range $[h_f, h_c]$ of mesh sizes h between a coarse mesh size h_c and a fine mesh size h_f (or, equivalently, for a range $[L_c, L_f]$ of values of L, $h_c = (b-a)/(L_c-1)$, $h_f = (b-a)/(L_f-1)$) can be evaluated numerically if the corresponding optimal values $N = N^{\mathrm{opt}}_c$ for $L = L_c$ and $N = N^{\mathrm{opt}}_f$ for $L = L_f$ are known; according to Remark 10, in turn, such estimates for the optimal values N^{opt}_c and N^{opt}_f can be obtained by exhaustive evaluation of errors at $h = h_c$ and $h = h_f$. In the present context exhaustive evaluation leads to the values $N^{\mathrm{opt}}_c = 3$ for $L = L_c = 6$ and $N^{\mathrm{opt}}_f = 17$ for $L = L_f = 30,000$ in the case $n = 0$ (derivative of order 0) and $N^{\mathrm{opt}}_c = 4$ for $L = L_c = 6$ and $N^{\mathrm{opt}}_f = 18$ for $L = L_f = 30,000$ in the case $n = 1$ (derivative of order 1). Figure 1 displays the resulting relations (47) rounded to the nearest integer for $n = 0$ and $n = 1$. Table 1 displays, for selected values of L ($h = (b - a)/(L - 1)$), the optimal values of N as determined from Figure 1 (in parenthesis), as well as the values obtained by direct

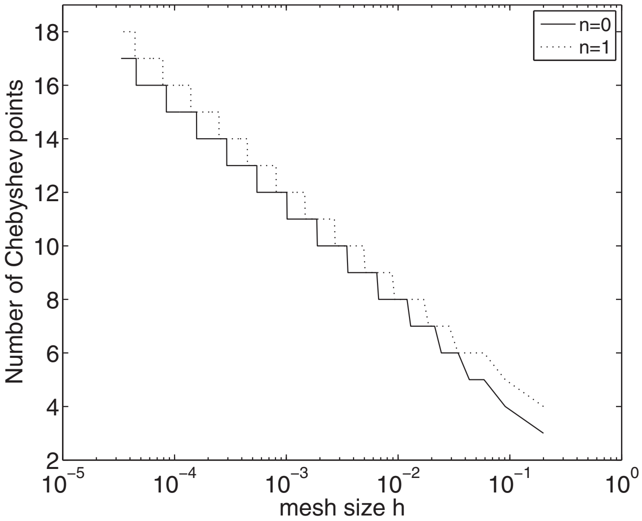

FIG. 1. *Optimal-order curves* $N^{\mathrm{opt}} = N^{\mathrm{opt}}(h)$ *((47) to (49)) using values of* $N_c = N^{\mathrm{opt}}(h_c)$ *and* $N_f = N^{\mathrm{opt}}(h_f)$ *estimated on the basis of exhaustive evaluation for mesh sizes* h_c *and* h_f*, as discussed in Remark* 10*. The specific values thus obtained and used for* $n = 0$ *are* $N_c = 3$ *(for* $L_c = 6$*) and* $N_f = 17$ *(for* $L_f = 30{,}000$*). The corresponding values for* $n = 1$ *are* $N_c = 4$ *(for* $L_c = 6$*) and* $N_f = 18$ *(for* $L_f = 30{,}000$*).*

TABLE 1

Optimal orders N^{opt} *obtained on the basis of exhaustive evaluation, as discussed in Remark* 10*, for meshes containing L points. The values in parenthesis are the corresponding point values on the curves displayed in Figure* 1*.*

	N^{opt}	
L	$n = 0$	$n = 1$
6	3(3)	4(4)
9	3(4)	4(5)
600	10(11)	12(11)
900	12(11)	12(12)
3,000	14(13)	14(14)
9,000	16(15)	15(16)
18,000	17(16)	16(17)
24,000	17(17)	18(17)
27,000	18(17)	17(18)
30,000	17(17)	18(18)

inspection of the errors arising from the algorithm for a series of values of N (without parenthesis). As can be seen, the optimal values of N obtained by direct inspection match closely those resulting from (47).

The results in Figure 2, in turn, confirm in practice the theoretical superiority of the LDC method over the standard approach which, in the present context, amounts to differentiation of LPI p_1 of degree one. In all cases the upper envelope of the LDC nth derivative error is parallel to the second order slope, while the LPI differentiation curve is parallel to the first order slope. The top-row graphs also display errors arising from differentiation of polynomial interpolants of higher order. These graphs show that, as discussed in Appendix A, differentiation of polynomials of degree $d = (r - 1)$ ($d = 1$ in the present $r = 2$ case) gives rise to lower differentiation errors than does differentiation of polynomial interpolants of higher order; see also Remark 16.

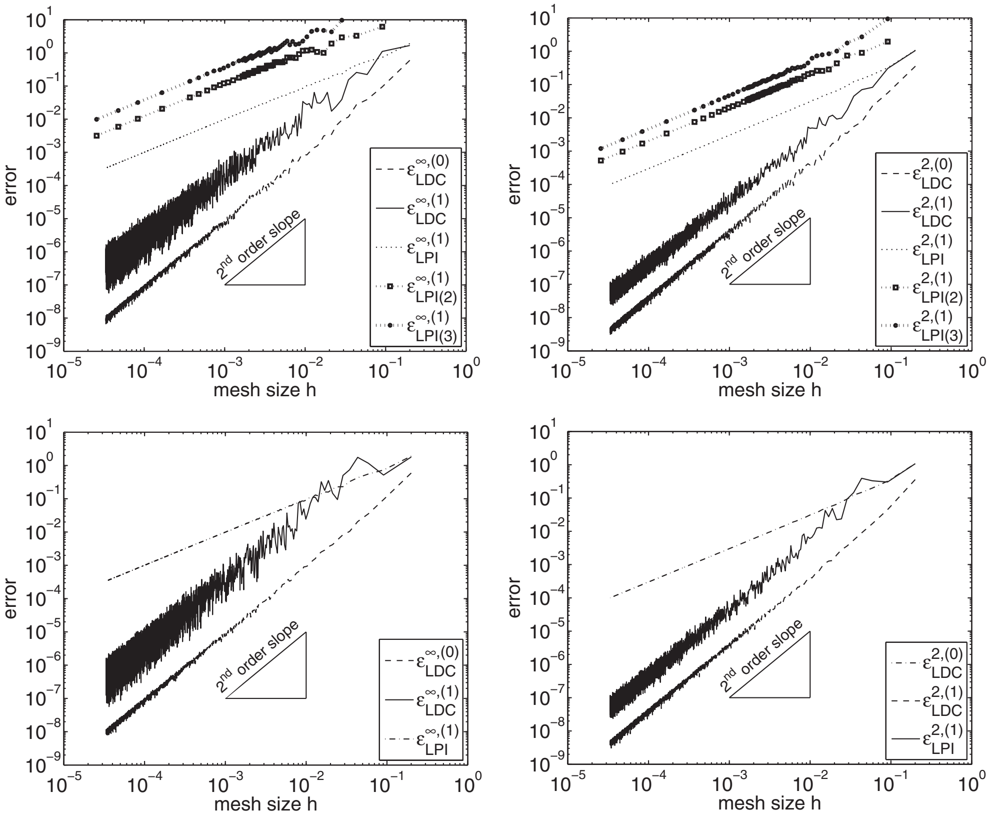

FIG. 2. *Second order convergence of the LDC methods for $n = 0$ and $n = 1$ in the L^∞-norm (left column) and L^2-norm (right column) and first order convergence resulting from direct differentiation of LPIs (= LPI(1); see Remark 16), LPI(2) and LPI(3) of degrees one, two, and three, respectively. Top row: the optimum value $N^{\text{opt}}(h)$ (see Figure 1) is used for all LDC differentiations. Bottom row: the suboptimal value $N = 18$ is used for all LDC differentiations.*

The LDC curves in the top row of the graphs in Figure 2 display LDC errors obtained as the optimal values of N is used for each value of h, while the LDC errors displayed in the bottom two graphs show the corresponding errors that arise when the suboptimal value $N = 18$ is used for all values of h; as expected, the optimal values of N result in somewhat better accuracies, especially in the region of the h axis for which the value $N = 18$ is furthest away from optimality. In any case, these examples demonstrate clearly that the LDC method overcomes the classical deterioration of orders of accuracy inherent in previous methodologies.

3.1.2. Domain decomposition. As indicated in Remark 11, application of the LDC approach in conjunction with a domain decomposition strategy may be advantageous for functions containing multiscale features. While a full discussion in these regards is beyond the scope of this paper, in this section we present an example that provides some insights in these regards. We thus consider the function (50) in the interval $[0, 1]$ (with parameters as in the previous section); we decompose this interval into 2^m subdomains and we evaluate, for various values of m, for coarse and fine mesh sizes and for the case $n = 1$ (first order derivatives), the maximum of the optimal parameters N^{opt} over the 2^m subdomains. The tabulated values of these

Table 2

Maxima of the optimal orders $N_{c,j}^{\mathrm{opt}}$ *and* $N_{f,j}^{\mathrm{opt}}$ *over the* 2^m *subintervals* $S_j^{2^m}$ $(j = 1, \ldots, 2^m)$, $[0,1] = \cup_j S_j^{2^m}$, *for an* $n = 1$ *example (first order derivatives), and for coarse and fine meshes containing* $L_c/2^m$ *and* $L_f/2^m$ *discretization points in each subdomain* $S_j^{2^m}$.

m	$\max_j N_{c,j}^{\mathrm{opt}}$	$\max_j N_{f,j}^{\mathrm{opt}}$
0	12	18
1	10	12
2	9	11
3	6	8
4	6	8

maxima provide an indication of the approximation properties of the LDC algorithm under a domain decomposition framework.

In more detail, our experiment was conducted as follows. Beginning with the domain $[0, 1]$, we used exhaustive evaluations as discussed in Remark 10 to obtain the values N_c^{opt} and N_f^{opt} corresponding, respectively, to a total of $L_c = 600$ and $L_f = 30,000$ discretization points over the complete interval $[0, 1]$ ($h_c = 1/(L_c - 1)$, $h_f = 1/(L_f - 1)$); the resulting values are $N_c^{\mathrm{opt}} = 12$ and $N_f^{\mathrm{opt}} = 18$. We then performed a first domain decomposition by splitting the interval $[0, 1]$ into the subdomains $S_1^2 = [0, 1/2]$ and $S_2^2 = [1/2, 1]$, and, for each one of the subdomains, we evaluated N_c^{opt} and N_f^{opt} using $L_c = 300$ and $L_f = 15,000$, respectively; the maximum of the two corresponding optimal orders $N_{c,1}^{\mathrm{opt}}$ and $N_{c,2}^{\mathrm{opt}}$ thus obtained was 10, whereas the maximum of the two optimal orders $N_{f,1}^{\mathrm{opt}}$ and $N_{f,2}^{\mathrm{opt}}$ was 12. Table 2 displays the corresponding maxima $\max_{j=1,\ldots,2^m} N_{f,j}^{\mathrm{opt}}$ of optimal parameters for various values of m as the domain $[0, 1]$ is decomposed into 2^m subdomains $S_1^{2^m}, \ldots, S_{2^m}^{2^m}$. The overall errors undergo only slight increases: not more than one digit of accuracy is lost from the case $m = 0$ to the case $m = 4$ for a given value of h, with smaller accuracy losses for intermediate values of m.

3.2. Higher-order derivatives: Random errors.

Data containing random errors can be particularly challenging for differentiation algorithms. In section 3.2.1 we demonstrate the stability of our numerical scheme for derivatives of various orders in the presence of a Gaussian distribution of noise. The results are in excellent agreement with the theoretical optimal convergence rates: no degradation occurs in the order of accuracy as a result of the differentiation process. In section 3.2.2 we compare our approach with the method put forth in reference [22] by discussing, in our context, a problem considered in that contribution.

3.2.1. Derivatives of order $n \geq 1$: Random errors of order $\mathcal{O}(h^r)$ ($r \geq 1$).

The LDC method can produce nth order derivatives ($n = 1, 2, 3 \ldots$) without order-of-accuracy deterioration even in cases in which data errors (of order $\mathcal{O}(h^r)$, $r = 1, 2, 3 \ldots$) are highly nonsmooth. For the examples of this section, which are designed to illustrate this point, we consider once again the function (50) in the interval $[0, 1]$ and we use *random* noisy data values $\{\bar{f}_\ell\}_{\ell=1}^L$ of f, $\bar{f}_\ell \sim f(x_\ell)$ with various orders of approximation.

The random noisy values $\bar{f}_\ell$ are produced as follows: we first use the method described in Remark 13 to produce a preliminary set of nonrandom noisy values $\hat{f}_\ell \sim f(x_\ell)$, with errors $|\hat{f}_\ell - f(x_\ell)| \sim \mathcal{O}(h^r)$; for the examples presented below in this section we use the values $r = 2, 3, 4$, and 5. We then obtain the needed noisy values $\bar{f}_\ell$ by means of a normally-distributed random number ϵ_ℓ with zero mean and

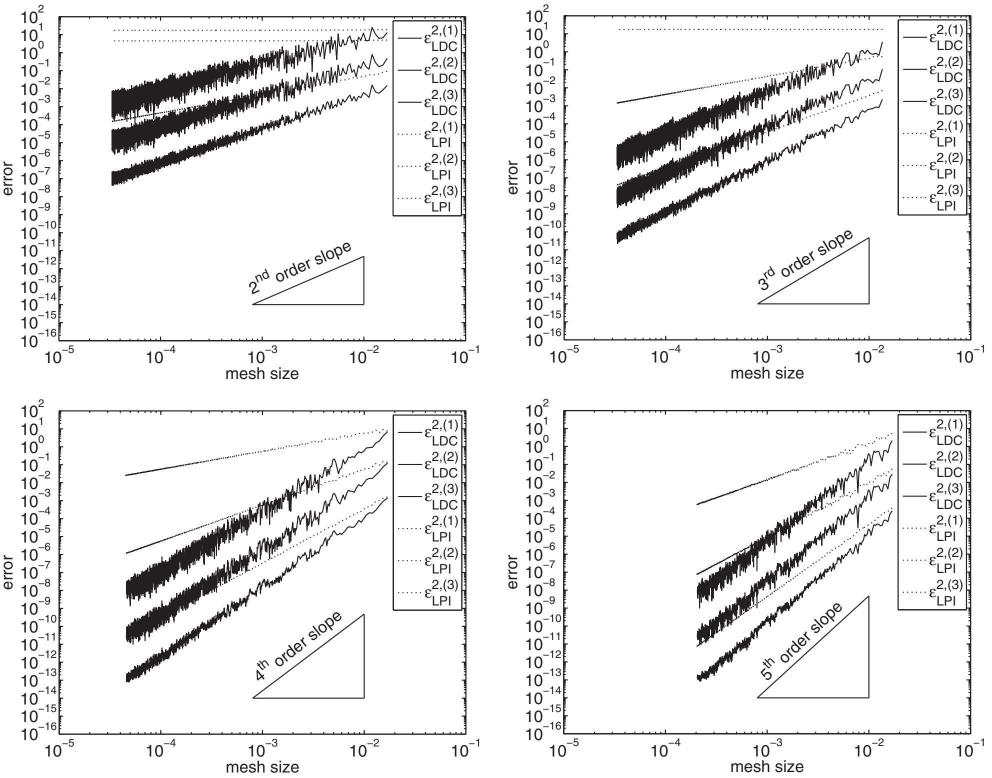

FIG. 3. *Convergence of order r with $r = 2$ (upper left), $r = 3$ (upper right), $r = 4$ (lower left), and $r = 5$ (lower right) produced by the LDC differentiation method for differentiation of order n with $n = 1$, 2, and 3 and corresponding convergence of orders $r-1$, $r-2$, and $r-3$, respectively, that result from direct differentiation of LPIs of degree $r - 1$ (see Remark* 16*). The data used contains Gaussian random errors of order $r = 2$, 3, 4, and 5 (from upper left to lower right) which were produced and added as indicated in the text. The coarse and fine parameter values were determined in all cases by exhaustive evaluation, as discussed in Remark* 10*, using $L_c = 60$ and $L_f = 30{,}000$. The resulting coarse and fine optimal parameters obtained are $N_c^{\mathrm{opt}} = 10$ and $N_f^{\mathrm{opt}} = 20$ for $r = 2$; $N_c = 10$ and $N_f = 25$ for $r = 3$; $N_c = 13$ and $N_f = 30$ for $r = 4$; and $N_c = 15$ and $N_f = 35$ for $r = 5$.*

variance 1: we let $\bar{f}_\ell = f(x_\ell) + (1 + \epsilon_\ell)(\hat{f}_\ell - f(x_\ell))$, which clearly gives noisy values with random errors of order $\mathcal{O}(h^r)$.

Suitable Chebyshev expansion orders N are determined as indicated in Remark 10 and (47). The specific values of N_c^{opt} and N_f^{opt} used in these examples for $L_c = 60$ and $L_f = 30,000$ are reported in the caption of Figure 3. (Note that this value of L_c is larger than the one used in section 3.1.2, which seems appropriate in the present context in which higher orders r of accuracy are considered. But the choice of the parameters L_c and L_f is otherwise completely unessential and arbitrary: any other pair of values of these parameters, even greatly different from those used here, would be equally adequate.)

The graphs in Figure 3 demonstrate excellent agreement with the convergence rates predicted by our theory, and, through comparison with the errors arising from differentiation of interpolating polynomials, they show clearly the significant improvements that result from the LDC algorithm over the performance of previous approaches.

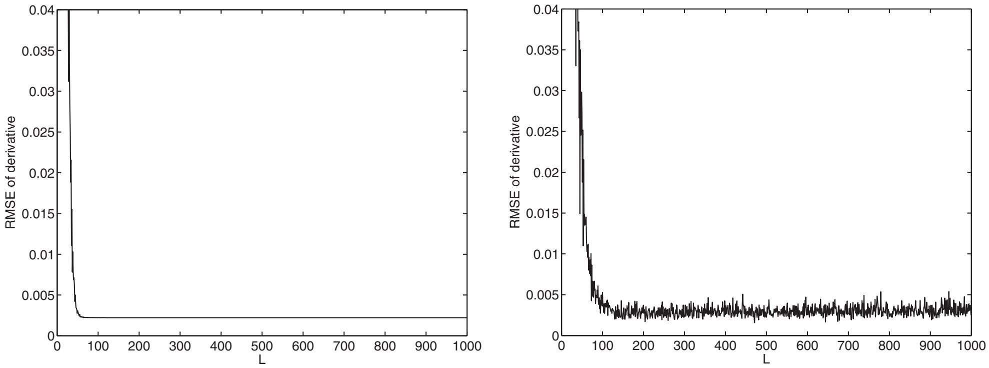

FIG. 4. *Root mean square error versus discretization size L. Left: $\varepsilon = 1$, $r = 6$. Right: ε is a uniformly distributed random number in* $[0, 1]$*,* $r = 3$.

3.2.2. Comparison to the RBF approximation method. In this section we compare the LDC methodology to the RBFs differentiation approach presented in [22]. For definiteness we focus on one example from that reference, which concerns the function

$$f(x) = \sin(2\pi x)\exp(-x^2). \tag{53}$$

Reference [22] applies the RBF differentiation algorithm to noisy data of the form $\bar{f}_i = f(x_i) + 0.001\varepsilon\sin(\pi x_i)$ in the domain $[-2, 2]$, $i = 1, \ldots, L$, and it presents a plot of differentiation errors as a function of L for the case $\varepsilon = 1$. Reference [22] quantifies the error resulting from the radial-basis-function approach in terms of the root mean square error (RMSE) in the first derivative; denoting by $g(x) = f'(x)$ and by $\tilde{g}(x)$ a corresponding approximate derivative, the RMSE is given by

$$E(\tilde{g}) = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(f'(x_i) - \tilde{g}(x_i))^2}, \tag{54}$$

where N_s denotes the number of points used to test the accuracy of the resulting approximation.

The RBF method [22] produces the derivative for the case $\varepsilon = 1$ with RMSE errors of the order of 0.008 on the basis of large numbers L of data points. On the left of Figure 4 we present the results produced by our method using $r = 6$, $N_c^{\text{opt}} = 25$ for $L_c = 24$ and $N_f^{\text{opt}} = 40$ for $L_f = 1000$ with values of N^{opt} otherwise determined by means of (47)–(49). In particular, the LDC approach, which unlike the RBF method does not require the solution of a system of linear equations, results in an RMSE error of ≈ 0.002 for large L—much closer to the ≈ 0.001 error in the original function values than the 0.008 RMSE errors produced by the RBF approach.

We now turn our attention to a more challenging problem. We use once again the function (53), but this time we let ε be a uniformly distributed random number in $(0, 1)$ so that the function values contain random errors, rather than the smooth errors considered in the previous case. In the present context we used the value $r = 3$, which we found to give rise to optimal accuracies for this case. The results are displayed on the right of Figure 4. The RMSE is more oscillatory, yet an excellent convergence, with RMSE errors below 0.005 (and, mostly, close to 0.003), results.

3.3. Differentiation of functions from data resulting from the numerical solution of a PDE. As is well known, evaluation of the gradients of the solution of a PDE from a finite element numerical solution by differentiation of the shape functions leads to a reduction (by one) in the order of accuracy. In this section with present an example demonstrating that the LDC approach can indeed be used to retrieve the full accuracy order in the gradient. This approach is particularly important for cases in which, for example, the accuracy order of the numerical gradient can somehow affect, through the solution process, the accuracy of the solution itself; an example of such a situation is presented in [5].

For our example we consider the problem of evaluating the gradient of a finite element numerical solution of the Poisson equation on a certain domain $\Omega \subseteq \mathbb{R}^3$. The domain Ω in our example results as a sphere of radius $R = 0.25$ centered at the origin is removed from a cube with vertices $[\pm 0.45, \pm 0.45, \pm 0.45]$. Clearly, the boundary Γ of the sphere and the boundary $\mathcal{B}$ of the cube form the inner and outer boundaries of the three dimensional domain Ω.

Denoting by $\boldsymbol{\nu}$ the outer normal to the domain Ω we thus consider the elliptic boundary value problem

$$\Delta u = f, \tag{55}$$

$$\boldsymbol{\nu} \cdot \nabla u|_\Gamma = g_\Gamma, \tag{56}$$

$$\left(u + \boldsymbol{\nu} \cdot \nabla u\right)|_{\mathcal{B}} = g_{\mathcal{B}} \tag{57}$$

and we seek to evaluate the solution gradient from a numerical finite element approximation u_h to the solution u. The basis of finite element solvers for the problem (55)–(57) lies on the corresponding variational formulation: the solution u is the unique element of the Sobolev space $H^1(\Omega)$ such that for all $v \in H^1(\Omega)$ we have

$$(u_h, v)_{\mathcal{B}} + (\nabla u_h, \nabla v)_\Omega = (g_{\mathcal{B}}, v)_{\mathcal{B}} + (g_\Gamma, v)_\Gamma - (f, v)_\Omega, \tag{58}$$

where $(\cdot,\cdot)_{\mathcal{B}}$, $(\cdot,\cdot)_\Gamma$, and $(\cdot,\cdot)_\Omega$ denote the L^2 scalar products in $\mathcal{B}$, Γ, and Ω, respectively.

In order to produce our finite element solution we discretize the computational domain Ω by means of a tetrahedral grid τ with maximum mesh size H and we use linear basis functions ϕ_i. Equation (58) then leads to the linear system

$$\left(\boldsymbol{C} + \boldsymbol{K}\right) \boldsymbol{u} = \boldsymbol{l}, \tag{59}$$

where

$$\boldsymbol{C}_{i,j} = (\phi_i, \phi_j)_{\mathcal{B}}, \quad \boldsymbol{K}_{i,j} = (\nabla\phi_i, \nabla\phi_j)_\Omega, \quad \boldsymbol{l}_i = (g_\Gamma, \phi_i)_\Gamma + (g_{\mathcal{B}}, \phi_i)_{\mathcal{B}} - (f, \phi_i)_\Omega. \tag{60}$$

We solve (59) by means of an iterative CG-solver [8] and thus produce our finite element approximate solution

$$u(\boldsymbol{x}) = \sum_{i \in T_x} u_i \phi_i(\boldsymbol{x}). \tag{61}$$

To proceed with our demonstration we now evaluate the gradient of the numerical solution (61) throughout Ω by two different methods: finite element differentiation and LDC differentiation.

The finite element gradient can be produced directly: from (62) we obtain

$$\nabla_\phi u(\boldsymbol{x}) \equiv \sum_{i \in T_x} u_i \nabla\phi_i(\boldsymbol{x}), \tag{62}$$

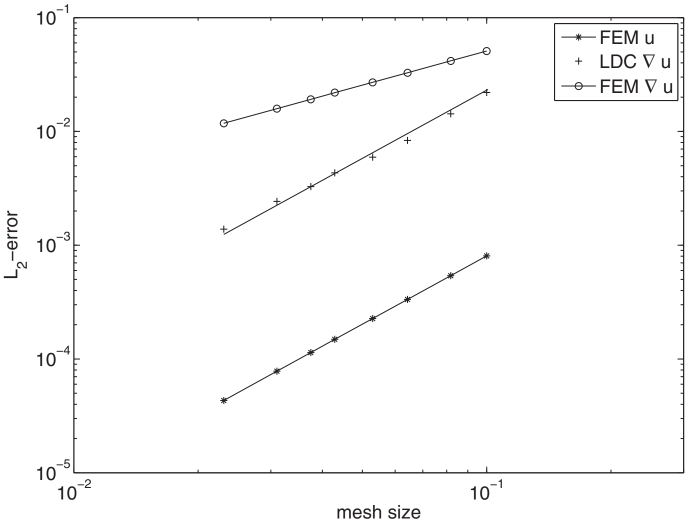

FIG. 5. *L_2-error vs. mesh size h for function and gradient values. The expected second and first order errors for the finite element method function and gradient are clearly observed, as is the second order convergence of the LDC gradient.*

where, for a given point $\boldsymbol{x} \in \Omega$, $T_x \in \tau$ denotes the element corresponding to the evaluation point $\boldsymbol{x}$. This expression can be evaluated easily for any point $\boldsymbol{x} \in \Omega$.

In order to compute the LDC approximation of the solution gradient, in turn, we use lines parallel to the coordinate axis passing through the point $\boldsymbol{x}$. Each one of these coordinate lines intersect the boundary of Ω twice—either on $\mathcal{B}$ both times or once on $\mathcal{B}$ and once on Γ. On each one of the $\boldsymbol{x}$-dependent segments given by intersection of these axes with the domain Ω we select an appropriate number N of Chebyshev points; in accordance with the prescriptions of the LDC algorithm, we determine the corresponding finite element T_c for every Chebyshev point $\boldsymbol{x_c}$ and we obtain the value of the finite element solution at the point $\boldsymbol{x_c}$ by evaluating (61) at $\boldsymbol{x} = \boldsymbol{x_c}$. Once the values at the Chebyshev points are known, we compute the Chebyshev coefficients for every dimension and we use (11)–(12) to obtain the LDC gradient ∇u.

For our numerical example we use for the right-hand sides of (55)–(57) the functions

$$f(\boldsymbol{x}) = 6e^{|\boldsymbol{x}|^2}\left(\cos(|\boldsymbol{x}|^2) - \sin(|\boldsymbol{x}|^2)\right) - 8|\boldsymbol{x}|^2 \sin(|\boldsymbol{x}|^2)e^{|\boldsymbol{x}|^2} \quad \text{for} \quad \boldsymbol{x} \in \Omega, \tag{63}$$

$$g_\Gamma(\boldsymbol{x}) = 2\left(\cos(|\boldsymbol{x}|^2) - \sin(|\boldsymbol{x}|^2)\right) e^{|\boldsymbol{x}|^2} \boldsymbol{\nu} \cdot \boldsymbol{x} \quad \text{for} \quad \boldsymbol{x} \in \Gamma, \tag{64}$$

$$g_{\mathcal{B}}(\boldsymbol{x}) = 2\left(\cos(|\boldsymbol{x}|^2) - \sin(|\boldsymbol{x}|^2)\right) e^{|\boldsymbol{x}|^2} \boldsymbol{\nu} \cdot \boldsymbol{x} + \cos(|\boldsymbol{x}|^2)e^{|\boldsymbol{x}|^2} \quad \text{for} \quad \boldsymbol{x} \in \mathcal{B} \tag{65}$$

for which the exact solution of the system is given by

$$u(\boldsymbol{x}) = \cos(|\boldsymbol{x}|^2)e^{|\boldsymbol{x}|^2} \quad \text{for} \quad \boldsymbol{x} \in \Omega. \tag{66}$$

Figure 5 displays the L^2-error on the solution as well as the finite element gradient and LDC gradient for various mesh sizes of the finite element grid. A clear second order convergence pattern can be observed for the error of the function and its LDC gradient. In contrast, the convergence rate of the finite element gradient is only accurate to first order, as expected.

4. Conclusions. We have proposed a simple and highly accurate approach for evaluation of derivatives from approximate discrete function values. Our text contains a full theoretical analysis of the method as well as a range of numerical examples, including differentiation of various orders and errors of various types: interpolatory errors, finite element solution errors, and random errors. The numerical results, which in all cases are in clear agreement with our theory, demonstrate the significant performance improvements that the new approach provides over the corresponding performance of previously existing methods.

Appendix A. Differentiation of noisy interpolants. As pointed out in section 1 and demonstrated in Figures 2 and 3 (cf. also Remark 16), nth order differentiation of an LPI of degree $d = (r-1)$ for a function f in presence of noise of order $\mathcal{O}(h^r)$ yields approximations of the nth derivative $f^{(n)}$ with the reduced order of accuracy $\mathcal{O}(h^{r-n})$; significant order-of-accuracy deterioration occurs. Since as is known [16, Th. 1, p. 289], the nth order derivatives of *noiseless* polynomial interpolants of degree d approximates the nth order derivative of f with an error of order $\mathcal{O}(h^{d+1-n})$ (the order of accuracy of the derivative approximation increases with d in the noiseless case), the question arises as to whether in presence of noise of order $\mathcal{O}(h^r)$, differentiation of polynomial interpolants of degree d higher than $(r-1)$ might give rise to approximations of the derivative $f^{(n)}$ with accuracy of order higher than $\mathcal{O}(h^{r-n})$.

The examples presented in Figure 2 in these regards indicate that this is not the case: the LPI(d) derivative approximations produced by differentiation of polynomial interpolants of degrees $d > (r-1)$ do not give rise in increased orders of accuracy and, in fact, they give rise to poorer accuracies, by significant additive constants, than that resulting from LPI($r-1$). In this appendix we provide a theoretical rationale for this observation on the basis of the classical theories of polynomial interpolation and differentiation.

To do this, recall the Newton form of the interpolating polynomial Q_d of degree d of a function f [16, pp. 245–248]:

$$Q_d(x) = \sum_{j=0}^{d} f[x_0,\dots,x_j] \prod_{k=0}^{j-1}(x-x_k), \tag{67}$$

where the divided differences $f[x_0,\dots,x_j]$ are given by

$$f[x_0,x_1,\dots,x_j] = \sum_{\ell=0}^{j} \frac{f(x_\ell)}{\displaystyle\prod_{\substack{k=0\\(k\neq\ell)}}^{j}(x_\ell-x_k)} . \tag{68}$$

For notational simplicity, for a fixed polynomial degree d we consider a uniform mesh size h, and we assume the interpolation region is the interval $[0,dh]$ so that $x_j = jh$, $j = 0,\dots d$.

To see that errors of order $\mathcal{O}(h^{r-n})$ result from nth order differentiation of polynomial interpolants of degree $d \geq (r-1)$ (with constants in the $\mathcal{O}(h^{r-n})$ error bounds that grow with d), we first note from (68) that, since $(x_\ell - x_k) = (\ell-k)h$, errors of order $\varepsilon = \mathcal{O}(h^r)$ in the approximate values of the function f give rise to errors of order ε/h^j in the divided differences $f[x_0,x_1,\dots,x_j]$. Use of such noisy divided-difference values in the expression (67) gives errors of order ε in the polynomial interpolant, since for $x \in [0,dh]$ the factor $\prod_{k=0}^{j-1}(x-x_k)$ in (67) is itself a quantity of order h^j.

But the derivative of $Q_d(x)$ is a sum of products of divided differences by terms of the form

$$\prod_{\substack{k=0\\(k\neq\ell)}}^{j-1}(x-x_k). \tag{69}$$

Each one of the polynomials (69) is a quantity of order h^{j-1}, which, when multiplied by the ε/h^j errors in the divided differences, gives rise to overall errors of order $\varepsilon/h = \mathcal{O}(h^{r-1})$ for the derivative of the interpolating polynomial. Since, as mentioned above, the nth derivative of the noiseless polynomial interpolant approximates the nth derivative of f with an error of order $\mathcal{O}(h^{d+1-n})$, it follows from the triangle inequality that for an error $\varepsilon = \mathcal{O}(h^r)$ in the given values of the function f the first derivative of the polynomial interpolating such noisy values approximates the exact first derivative of f with an error of order h^{r-1}, independently of the degree $d \geq (r-1)$ of the interpolating polynomial used. In fact the multiplicative constant in the $\mathcal{O}\left(h^{r-1}\right)$ error bound grows *quadratically* with the polynomial degree d since the number of terms of order ε/h in the derivative of the Newton polynomial grows quadratically with d and, thus, the optimal selection for the degree of the interpolating polynomial is $d = r-1$. This remark is consistent with the results of our experiments displayed in Figure 2: derivatives using polynomial interpolants LPI(2) and LPI(3) of degrees 2 and 3 give rise to order-of-magnitude higher errors than those resulting from the optimal interpolant, which in this case is the first degree polynomial LPI = LPI(1). A similar analysis can clearly be performed for differentiation of any order n: in this case, $\mathcal{O}\left(h^{r-n}\right)$ errors result for any interpolation degree $d \geq (r-1)$, with multiplicative constants in the $\mathcal{O}\left(h^{r-n}\right)$ error bound that grow with the polynomial degree d.

REFERENCES

[1] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* U.S. Government Printing Office, Washington, DC, 1964.

[2] R. S. Anderssen and M. Hegland, *For numerical differentiation, dimensionality can be a blessing!*, Math. Comp., 68 (1999), pp. 1121–1141.

[3] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*, Dover, New York, 2001.

[4] D. Braess, *Finite Elements. Theory, Fast Solvers, and Applications in Solid Mechanics,* 3rd ed., Cambridge University Press, Cambridge, UK, 2007.

[5] D. Hoch, *Nonreflecting Boundary Conditions Obtained From Equivalent Sources For Time-Dependent Scattering Problems*, Ph.D. Thesis, California Institute of Technology, Pasadena, CA, 2008.

[6] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, *On the Lambert W function*, Adv. Comput. Math., 5 (1996), pp. 329–359.

[7] J. Cheng, Y. C. Hon, and Y. B. Wang, *A numerical method for the discontinuous solutions of Abel integral equations*, in Inverse Problems and Spectral Theory, Contemp. Math. 348, American Mathematical Society, Providence, RI, 2004, pp. 233–243.

[8] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 409–436.

[9] J. Cullum, *Numerical differentiation and regularization*, SIAM J. Numer. Anal., 8 (1971), pp. 254–265.

[10] S. R. Deans, *The Radon Transform and Some of Its Applications*, Wiley, New York, 1983.

[11] R. Gorenflo and S. Vessella, *Abel integral equations*, in Analysis and Applications, Lecture Notes in Math. 1461, Springer-Verlag, New York, 1991.

[12] C. W. Groetsch, *Differentiation of approximately specified functions*, Amer. Math. Monthly, 98 (1991), pp. 847–850.

[13] M. Hanke and O. Scherzer, *Inverse problem light: Numerical differentiation*, Amer. Math. Monthly, 108 (2001), pp, 512–521.

[14] M. Hanke and O. Scherzer, *Error analysis of an equation error method for the identification of the diffusion coefficient in a quasi-linear parabolic differential equation*, SIAM J. Appl. Math., 59 (1999), pp, 1012–1027.

[15] D. B. Hunter, *An iterative method of numerical differentiation*, Comput. J., 3 (1960/1961), pp. 270–271.

[16] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Dover, New York, 1994.

[17] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, Chapman & Hall, London, 2003.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 1988.

[19] A. G. Ramm and A. B. Smirnova, *On stable numerical differentiation*, Math. Comp., 70 (2001), pp. 1131–1153.

[20] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*, Scripta Series in Mathematics, Wiley, New York, 1977.

[21] Y. B. Wang, X. Z Jia, and J. Cheng, *A numerical differentiation method and its application to reconstruction of discontinuity*, Inverse Problems, 18 (2002), pp, 1461–1476.

[22] T. Wei and Y. C. Hon, *Numerical differentiation by radial basis functions approximation*, Adv. Comput. Math., 27 (2007), pp, 247–272.