Mathematical Modelling and Analysis
Volume 26, Issue 2, 304–317, 2021
https://doi.org/10.3846/mma.2021.13119

Higher Accuracy Order in Differentiation-by-Integration

Andrej Liptaj

Institute of Physics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 11 Bratislava, Slovak Republic
E-mail(*corresp.*): andrej.liptaj@savba.sk

Received July 8, 2020; revised March 19, 2021; accepted March 20, 2021

Abstract. In this text explicit forms of several higher precision order kernel functions (to be used in the differentiation-by-integration procedure) are given for several derivative orders. Also, a system of linear equations is formulated which allows to construct kernels with an arbitrary precision for an arbitrary derivative order. A computer study is realized and it is shown that numerical differentiation based on higher precision order kernels performs much better (w.r.t. errors) than the same procedure based on the usual Legendre-polynomial kernels. Presented results may have implications for numerical implementations of the differentiation-by-integration method.

Keywords: differentiation by integration, generalized Lanczos derivative, numerical differentiation, higher-order method, accuracy.

AMS Subject Classification: 26A24; 65D25.

1 Introduction

Derivative, if exists, can be expressed in an alternative way

$$f'(x)\,|_{x=x_0} = \lim_{h\to 0} \mathcal{D}^h_{d=1}(f)\,|_{x_0} \equiv \lim_{h\to 0} -\frac{1}{h}\int_{-1}^{1} k(t)\, f(x_0+ht)\,dt, \tag{1.1}$$

where k is an appropriate kernel function and d indicates the derivative order. The method is known as "*differentiation by integration*" (DbI) and an example was first published by Cioranescu [2]. Later, Lanczos [12] studied a version of (1.1) with kernel

$$k(t) = -\frac{3}{2}t \tag{1.2}$$

and his name became associated with this type of differentiation[1]. DbI has some interesting features because it generalizes the ordinary derivative in several aspects. It may be used in situations where the latter does not exist, or, when generalized to higher orders, it can be (in some cases) seen as a way to define a fractional-order derivative [3, 15]. The DbI also attracts attention for its potential applications in numerical analysis [9, 10, 11, 17, 22].

In published texts [1, 6, 8, 14, 18, 20, 21] two dominant ideas (closely related to each other [5]) can be seen:

- The least-squares property: the derivative as defined by the Lanczos' formula (1.1)–(1.2) behaves as the slope parameter of a line which is in the least-squares manner adjusted to the function f on the interval $[x_0 - h, x_0 + h]$. The idea was further generalized to higher derivatives in [1] where the kernel functions are based on the Legendre polynomials. This property is to be seen as very closely linked to the Savitzky–Golay filter [19] and the related way of numerically computing a derivative from discrete data sets.

- Orthogonality approach: if one assumes a function f is analytic[2] at x_0 with the radius of convergence $R_f > h$, one can expand it into the Taylor series
$$\int_{-1}^{1} k\left(t\right) f\left(x_0 + ht\right) dt = \sum_{n=0}^{\infty} \frac{1}{n!} h^n f^{(n)}\left(x_0\right) \int_{-1}^{1} k\left(t\right) t^n dt, \tag{1.3}$$
and search for a function k orthogonal to desired powers of t^n so as to make vanish all terms except those which we are interested in. For instance, to extract the dth derivative one can use a polynomial orthogonal to all t^n for $n < d$ (e.g. the Legendre polynomial L_d). If one then multiplies the resulting expression by an appropriate power of h, the limit $h \to 0$ makes disappear all higher terms $n > d$ and isolates the one with the dth derivative. This approach can provide large generalizations (as presented in [6]).

An universal procedure for constructing a kernel function which does not rely on a power expansion was for the first derivative presented in [10], a full generalization to all derivative orders was given in [13]. A function k is a valid kernel function for computing the dth derivative if and only if
$$k(t) = \frac{d^d}{dt^d} \omega\left(t\right), \tag{1.4}$$
where ω is a weight function satisfying
$$\int_{-1}^{1} \omega\left(t\right) dt = 1, \tag{1.5}$$
$$\frac{d^m}{dt^m} \omega\left(t\right) |_{t=-1} = \frac{d^m}{dt^m} \omega\left(t\right) |_{t=+1} = 0 \quad \text{for all } 0 \le m < d, \tag{1.6}$$

[1] "*Generalized Lanczos derivative*" is to be seen as an alternative label to "DbI".

[2] So f can be expressed as a power series, which are known to be uniformly convergent inside the radius of convergence.

assuming that all appearing integrals and derivatives exist. The dth derivative is then computed as

$$f^{(d)}(x)\,|_{x=x_0} = \lim_{h\to 0} \mathcal{D}_d^h(f)\,|_{x_0} \equiv \lim_{h\to 0}\left(\frac{-1}{h}\right)^d \int_{-1}^{1} k(t)\, f(x_0+ht)\, dt. \tag{1.7}$$

In the standard finite-difference approach is a large emphasis (in relation to numerical applications) given to the accuracy order in the discretization-parameter (h) expansion. It is known that the higher-order approaches perform much better than the lower-order ones[3]. Systematic study of these effects is somewhat missing with respect to the DbI methods, yet higher accuracy orders lead (as demonstrated later) to a largely improved overall numerical precision. In existing literature, formulas with higher accuracy order can be found (e.g. formula (4.1) in [6]), however explicit kernel function forms which could be directly used as a recipe for an immediate implementation are missing. One also lacks a numerical study of performance of such higher-order methods. The aim of this text is to

- provide explicit expressions for kernel functions for first few orders in precision and derivative (Section 2 and Appendix C). Further, we want to
- give an explicit formulation of a linear system which, when solved, leads to kernel functions for any derivative order and any order in precision (Section 2 and Appendix B). We also want to
- perform a numerical study of the higher accuracy order approach (Section 3).

We close the text by providing summary, conclusions and outlook (Section 4).

2 Higher order formulas

Different strategies can be adopted when constructing higher precision order kernels. Orthogonality approach is certainly a valid option and we briefly investigate it in Appendix A. Yet, we prefer to chose a different strategy which stems from basic principles (1.4)–(1.6). We adopt the most natural choice and search for kernels in form of least-degree polynomials. This minimalist approach is allowed by a "brute force" computation which provides us with the full control over coefficients and leads to explicit relations between them.

A general polynomial respecting (1.6) is written as

$$\begin{aligned}\omega(x) &= \mathcal{N}\left(1-x^2\right)^d \sum_{n=0}^{N} a_n x^n \qquad (2.1)\\ &= \mathcal{N}\left[\sum_{j=0}^{2d} \frac{1}{2}\left\{1+(-1)^j\right\} C_{d,j/2}\,(-1)^{j/2}\, x^j\right] \sum_{n=0}^{N} a_n x^n,\end{aligned}$$

[3] A simple forward difference is over-performed by the central difference, which is on its turn over-performed by, for example, the five-point rule.

where $\mathcal{N}$ is normalization, d is derivative order and $C_{a,b}$ represents binomial coefficient $\binom{a}{b}$. Formally appearing non-integer powers in the last line are canceled by the $1+(-1)^j$ term. The two sums can be multiplied

$$\omega\left(x\right) = \mathcal{N} \sum_{n=0}^{N+2d} \Omega_n x^n,$$

where

$$\Omega_n = \sum_{k=n-2d}^{n} \frac{1}{2} H_k H_{N-k} \left\{1 + \left(-1\right)^{n-k}\right\} C_{d,(n-k)/2} \left(-1\right)^{(n-k)/2} a_k, \tag{2.2}$$

with H indicating a step function $H_i = \begin{cases} 0, & \text{if } i < 0, \\ 1, & \text{else.} \end{cases}$ The condition (1.5) implies the normalization

$$\mathcal{N} = \left(\sum_{n=0}^{\lfloor (N+2d)/2 \rfloor} \frac{2\Omega_{2n}}{2n+1} \right)^{-1}, \tag{2.3}$$

where symbol $\lfloor \ldots \rfloor$ stands for the floor function. For the needs of formula (1.4) one has to compute higher-order derivatives

$$k(t) \equiv \omega^{(d)}(t) = \mathcal{N} \sum_{n=0}^{N+d} \frac{(n+d)!}{n!} \Omega_{n+d} t^n. \tag{2.4}$$

Expanding a function f into Taylor series and plugging (2.4) into (1.7) one gets[4]

$$\mathcal{D}_d^h\left(f\right) = \left(-\frac{1}{h}\right)^d \int_{-1}^{1} \mathcal{N} \sum_{n=0}^{N+d} \alpha_n t^n \sum_{m=0}^{\infty} \frac{1}{m!} h^m f^{(m)} t^m dt = \mathcal{N} \left(-1\right)^d \sum_{m=0}^{\infty} \frac{1}{m!} f^{(m)}$$
$$\times \left\{ \sum_{n=0}^{N+d} \frac{\alpha_n}{n+m+1} \left[1 + \left(-1\right)^{m+n}\right] \right\} h^{m-d}, \quad \alpha_n \equiv \frac{(n+d)!}{n!} \Omega_{n+d}, \tag{2.5}$$

where one controls the accuracy order by adjusting the coefficients in front of the powers of h. The series formally contains also negative powers (for $m < d$), formula (1.7) however implies that their coefficients are identically zero. Further more, formula (1.7) also implies[5] that the overall coefficient in front of the constant term (h^0) is $f^{(d)}$. The interesting terms are so those with $m > d$.

An important observation is that the odd powers of h are controlled only by odd-indexed coefficients $a_1, a_3, a_5, \ldots$. It can be verified in an explicit way,

[4] We assume the analyticity of f on an open interval containing $[x_0 - h, x_0 + h]$ and we omit its argument (which is implicitly assumed to be the point of differentiation x_0).

[5] Both statements can be verified in an explicit computation (but do not need to be).

that the index of Ω has the same parity as the power of h (symbol $\sim$ means conservation of index and power-related properties)

$$h^{m-d}\sum_{n=0}^{N+d}\frac{\alpha_n\left[1+(-1)^{m+n}\right]}{n+m+1}\sim\sum_{n=0}^{N+d}h^{m+n-(d+n)}\Omega_{n+d}\left[1+(-1)^{m+n}\right]$$
$$=\sum_{n=0}^{N+d}h^{m+n-Z_n}\Omega_{Z_n}\left[1+(-1)^{m+n}\right]\sim\sum_{n=0}^{N+d}h^{2i_n-Z_n}\Omega_{Z_n}=\sum_{n=0}^{N+d}h^{Z_n+2[i_n-Z_n]}\Omega_{Z_n},$$

where the factor $1+(-1)^{m+n}$ allows only for even-valued sums of the two numbers $m+n=2i$. The expression for Ω_Z is, on its turn, given only by terms whose coefficients a_i have an index of the same parity as Z:

$$\Omega_{2i+1}\sim\sum_{k=n-2d}^{n}\left\{1+(-1)^{2i+1-k}\right\}(-1)^{(2i+1-k)/2}a_k,\quad 0<k<N$$
$$\sim\sum_{k=n-2d}^{n}\left\{1+(-1)^{1+k}\right\}(-1)^{(2i+1-k)/2}a_k.$$

If k is even, then $1+k$ is odd and $1+(-1)^{1+k}$ is zero for all related a_k terms. Therefore Ω_{2i+1} depends only on the odd-index a_i coefficients. This being established, in what follows we will study only the even-index based weight functions[6], i.e. even functions of x on the interval $[-1,1]$. Even-indexed coefficients, controlling the even powers of h, cannot be set all to zero because one needs to fulfill the normalization condition (1.5).

We are now in the situation, where we need to adjust the even-indexed coefficients $a_0,a_2,a_4,\ldots$ appearing in

$$\mathcal{N}\sum_{m=0}^{\infty}\frac{(-1)^d}{m!}f^{(m)}\left\{\sum_{n=0}^{N+d}\frac{\alpha_n\left(a_0,\ldots,a_N\right)}{n+m+1}\left[1+(-1)^{m+n}\right]\right\}h^{m-d}\tag{2.6}$$
$$=\mathcal{D}_d^h(f)=f^{(d)}+c_2h^2+c_4h^4+c_6h^6+\ldots$$

so as to make chosen c_i coefficients vanish. Let us denote the highest even-power h term which is meant to be set to zero (together with all lower terms) by h^{2i}. For achieving that we need to match i (even-indexed) coefficients and formulate i equations. In a compact form the jth equation can be written $(1\le j\le i)$

$$\mathrm{E}_j:\qquad\sum_{n=0}^{2i+d}\frac{\alpha_n\left(a_0,\ldots,a_N\right)}{n+2j+d+1}\left[1+(-1)^{2j+d+n}\right]=0,\tag{2.7}$$

where one can fix the value of some coefficients

$$a_k=\begin{cases}1, & \text{if } k=0,\\ 0, & \text{if } 1\le k\le 2i \text{ and } k \text{ is odd},\\ 0, & \text{if } 2i<k,\end{cases}\tag{2.8}$$

[6] Aiming the highest accuracy order, we have no motivation to keep or study odd-h-power terms. We directly set them all to zero.

and equality

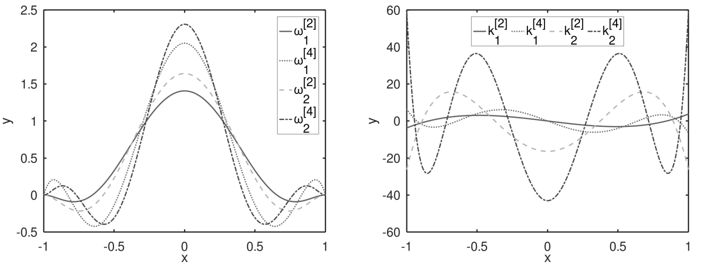
$$N = 2i \tag{2.9}$$

is assumed in all expressions. Here, without loss of generality, we set the first coefficient to one $a_0 = 1$. Initially a general number, N (as an unspecified number of polynomial terms) can be cut to $2i$ (it comprises all coefficients, odd-indexed included) and is so determined only by the accuracy order. Once the system is solved and a_i coefficients are found, one computes the normalization $\mathcal{N}$ from (2.3) and constructs the weight function (2.1). The kernel function is given as its dth derivative (1.4). An example solution of the system is presented in Appendix B.

We will note $\omega_d^{[2i]}$ and $k_d^{[2i]}$ those functions, which provide dth derivative with order of precision $\mathcal{O}\left(h^{2i+2}\right)$

$$\mathcal{D}_d^h\left(f, k_d^{[2i]}\right) = f^{(d)} + \mathcal{O}\left(h^{2i+2}\right).$$

This notations reflects the fact that the highest even-power term which is annihilated is h^{2i}. Since the next one is also zero (because has an odd power), the first nonzero term corresponds to the power $2i+2$. Solutions of the system (2.7) for all cases with $1 \leq d \leq 4$ and $2 \leq 2i \leq 10$ are listed in Appendix C. The cases with $2i = 0$ are omitted because they correspond to the known Legendre-polynomial based kernels[7] (e.g. formula (1.2)). Four selected weight functions and the corresponding kernel functions ($d \leq 2$ and $2i \leq 4$) are depicted in Figure 1.



(a) (b)

Figure 1. Selected weight functions (a) and corresponding kernel functions (b).

3 Numerical investigations

The topic under consideration is, from the perspective of a numerical study[8], rather large and multidimensional. One could by interested in the dependence

[7] Our approach represents a generalization which includes previously known results as special cases.

[8] Numerical analysis was done with the GNU Octave [7] software 5.1.0 configured for "x86_64-suse-linux-gnu". The *quadl* function was used for numerical integration. The full code used to reach the results presented in this section can be downloaded from `http://147.213.122.82/~andrej/LanczosHO/numExperiment.m`. Author also greatly profited from the WxMaxima tools [16].

Table 1. The absolute error E (formula (3.1)) for 4 derivatives orders (d), two kernel-function types (higher order HO vs. least-squares LS) and three test functions with fixed test points. The discretization parameter h is tuned to an optimal value for one of the methods (in bold), the result of other method is shown also.

d	Opt. for	$\sin(x)\,,\ x_0 = 1$			$\exp(x)\,,\ x_0 = \pi$			$\ln(x)\,,\ x_0 = \frac{1}{2}$		
		h^{opt}	E_{HO}	E_{LS}	h^{opt}	E_{HO}	E_{LS}	h^{opt}	E_{HO}	E_{LS}
1	HO	10^{-2}	$\mathbf{1.62 \times 10^{-14}}$	5.40×10^{-6}	10^{-2}	$\mathbf{6.64 \times 10^{-13}}$	2.31×10^{-4}	10^{-3}	$\mathbf{8.53 \times 10^{-14}}$	1.60×10^{-6}
	LS	10^{-5}	1.32×10^{-11}	$\mathbf{1.39 \times 10^{-11}}$	10^{-5}	7.73×10^{-10}	$\mathbf{4.26 \times 10^{-10}}$	10^{-5}	2.37×10^{-11}	$\mathbf{1.57 \times 10^{-10}}$
2	HO	10^{-1}	$\mathbf{7.82 \times 10^{-12}}$	6.00×10^{-4}	10^{-1}	$\mathbf{2.10 \times 10^{-10}}$	1.65×10^{-2}	10^{-2}	$\mathbf{2.60 \times 10^{-11}}$	6.86×10^{-4}
	LS	10^{-4}	5.74×10^{-7}	$\mathbf{1.41 \times 10^{-8}}$	10^{-4}	1.11×10^{-5}	$\mathbf{4.26 \times 10^{-7}}$	10^{-4}	5.13×10^{-7}	$\mathbf{4.23 \times 10^{-8}}$
3	HO	10^{-1}	$\mathbf{2.47 \times 10^{-11}}$	3.00×10^{-4}	10^{-1}	$\mathbf{4.26 \times 10^{-10}}$	1.29×10^{-2}	10^{-2}	$\mathbf{1.20 \times 10^{-8}}$	4.27×10^{-3}
	LS	10^{-3}	1.63×10^{-5}	$\mathbf{6.08 \times 10^{-7}}$	10^{-3}	3.58×10^{-5}	$\mathbf{6.82 \times 10^{-6}}$	10^{-3}	7.56×10^{-6}	$\mathbf{4.35 \times 10^{-5}}$
4	HO	10^{-1}	$\mathbf{4.08 \times 10^{-11}}$	3.82×10^{-4}	10^{-1}	$\mathbf{7.77 \times 10^{-8}}$	1.05×10^{-2}	10^{-2}	$\mathbf{1.39 \times 10^{-4}}$	3.49×10^{-2}
	LS	10^{-2}	1.67×10^{-4}	$\mathbf{1.87 \times 10^{-6}}$	10^{-2}	1.49×10^{-3}	$\mathbf{1.31 \times 10^{-5}}$	10^{-3}	1.32	$\mathbf{8.53 \times 10^{-3}}$

of results on the size of the discretization parameter h, or on the derivative order of the kernel function, or on the choice of the test functions.

To reduce the complexity, we opt for our purposes only for the $\mathcal{O}\left(h^6\right)$ precision kernels $k_d^{[2i=4]}$. Next, the optimal size of the discretization parameter is analyzed. For that we select three test functions with fixed points of differentiation x_0

$$\sin(x)\ \text{ with } x_0 = 1; \quad \exp(x)\ \text{ with } x_0 = \pi \text{ and } \ln(x)\ \text{ with } x_0 = 0.5.$$

Using these settings we scan 8 orders of magnitude in the discretization parameter $10^{-8} \leq h \leq 10^{-1}$, changing the step in a geometrical progression with factor 10. We determine the absolute errors

$$E = |f_{\text{anal.}}^{(d)}(x) - f_{\text{DbI}}^{(d)}(x)|_{x=x_0}, \tag{3.1}$$

where "anal." refers to the evaluation of the analytic derivative of f at x_0 and "DbI" represents the results of the DbI evaluation based on the step size h. For comparison purposes we evaluate, in addition to the results based on higher-order (HO) precision kernels, also the results from the least-squares (LS) approach (i.e. kernels constructed from a single Legendre polynomial.)

As seen from the error summary Table 1, the optimal h for higher precision order methods is $10^{-2} \lesssim h \lesssim 10^{-1}$, for the LS methods it is systematically smaller[9] $10^{-4} \lesssim h \lesssim 10^{-3}$. The important observation is that the HO kernels significantly over-perform the LS kernels (if each method uses its own optimal value of h). If the HO kernels are used with an h-size that is optimal for the LS methods, the HO results remain competitive.

Fixing the discretization parameter size to near-optimal values $h^{HO} = 10^{-2}$ and $h^{LS} = 10^{-3}$, we study the error behavior on a whole interval: $[-\pi, \pi]$ for $\sin(x)$, $[-2, 2]$ for $\exp(x)$ and $[0.1, 2]$ for $\ln(x)$. Results (i.e. pointwise errors defined by formula (3.1)) are shown in Figure 2 and provide a straightforward conclusion: when interested in absolute errors, the HO differentiation provides several orders of magnitude more precise results than the LS methods.

[9] This behavior is also observed in finite-difference methods when comparing formulas with different accuracy order.

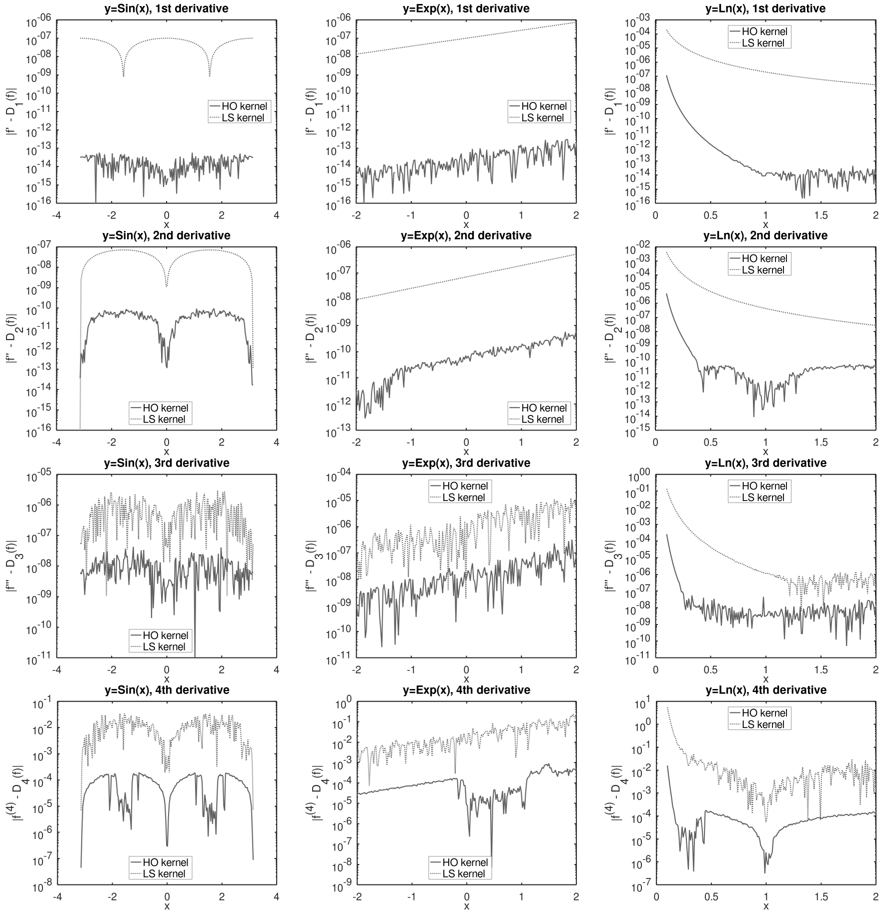


Figure 2. Absolute errors of the HO approach (full line) and LS-based methods (dotted line) for 3 functions (in columns) and 4 derivative orders (rows).

4 Summary, conclusions and outlook

In the finite difference methods, the higher precision approach is a well established standard. Despite clear advantages, such approach was not well studied for the DbI up to now. Using straightforward calculations we constructed a system of linear equations (2.7) which leads to arbitrary precision order kernel functions for arbitrary derivative in the DbI procedure. Numerical tests show that these kernels significantly over-perform (with respect to precision) standard least-squares kernels based on the Legendre polynomials, which were most widely used and discussed up to now.

The here-presented results are formulated in terms of continuous functions, a discrete formulation may represent an interesting topic. The appropriate tools and methods for the discretization of the DbI procedure have already been presented in several texts (see e.g. [4]).

Acknowledgements

The work was supported by VEGA grant No. 2/0153/17.

References

[1] N. Burch, P.E. Fishback and R. Gordon. The least-squares property of the Lanczos derivative. *Mathematics Magazine*, **78**(5):368–378, 2005. https://doi.org/10.2307/30044192.

[2] N. Cioranescu. La généralisation de la première formule de la moyenne. *Enseign. Math.*, **37**:292–302, 1938.

[3] E. Diekema. The fractional orthogonal derivative. *Mathematics*, **3**:273–298, 2015. https://doi.org/10.3390/math3020273.

[4] E. Diekema. The fractional orthogonal difference with applications. *Mathematics*, **3**(2):487–509, 2015. https://doi.org/10.3390/math3020487.

[5] E. Diekema. The fractional orthogonal derivative for functions of one and two variables. *PhD thesis*, 2018. Available from Internet: `https://pure.uva.nl/ws/files/29401750/Thesis.pdf`.

[6] E. Diekema and T.H. Koornwinder. Differentiation by integration using orthogonal polynomials, a survey. *Journal of Approximation Theory*, **164**(5):637–667, 2012. https://doi.org/10.1016/j.jat.2012.01.003.

[7] J.W. Eaton, D.Bateman, S. Hauberg and R. Wehbring. *GNU Octave version 5.1.0 manual: a high-level interactive language for numerical computations*, 2019. Available from Internet: `https://www.gnu.org/software/octave/doc/v5.1.0/`.

[8] A.H. Galeana, R.L. Vázquez, J.L.-Bonilla and L.-I. Pişcoran. On the Cioranescu - (Haslam - Jones) - Lanczos generalized derivative. *Global Journal of Advanced Researchon Classical and Modern Geometries*, **3**(1):44–99, 2014.

[9] C.W. Groetsch. Lanczos' generalized derivative. *The American Mathematical Monthly*, **105**(4):320–326, 1998. https://doi.org/10.1080/00029890.1998.12004888.

[10] D.L. Hicks and L.M. Liebrock. Lanczos' generalized derivative: Insights and applications. *Applied Mathematics and Computation*, **112**(1):63–73, 2000. https://doi.org/10.1016/S0096-3003(99)00048-X.

[11] X. Huang, Ch. Wu and J. Zhou. Numerical differentiation by integration. *Mathematics of Computation*, **83**(286):789–807, 2014. https://doi.org/10.1090/S0025-5718-2013-02722-6.

[12] C. Lanczos. *Applied analysis.* Prentice-Hall Mathematics, Englewood Cliffs, N.J., 1956.

[13] A. Liptaj. Maximal generalization of Lanczos' derivative using one-dimensional integrals. *arXiv e-prints*, p. arXiv:1906.04921, June 2019.

[14] D.-Y. Liu, O. Gibaru and W. Perruquetti. Differentiation by integration with Jacobi polynomials. *Journal of Computational and Applied Mathematics*, **235**(9):3015–3032, 2011. https://doi.org/10.1016/j.cam.2010.12.023.

[15] D.-Y. Liu, O. Gibaru, W. Perruquetti and T.-M. Laleg-Kirati. Fractional order differentiation by integration and error analysis in noisy environment. *IEEE Transactions on Automatic Control*, **60**(11):2945–2960, 2015. https://doi.org/10.1109/TAC.2015.2417852.

[16] Maxima. Maxima, a computer algebra system. version 5.34.1, 2014. Available from Internet: `http://maxima.sourceforge.net/`.

[17] T.J. McDevitt. Discrete Lanczos derivatives of noisy data. *International Journal of Computer Mathematics*, **89**(7):916–931, 2012. https://doi.org/10.1080/00207160.2012.666348.

[18] S.K. Rangarajan and S.P. Purushothaman. Lanczos generalized derivative for higher orders. *Journal of Computational and Applied Mathematics*, **177**(2):461–465, 2005. https://doi.org/10.1016/j.cam.2004.10.016.

[19] A. Savitzky and M.J.E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, **36**(8):1627–1639, 1964. https://doi.org/10.1021/ac60214a047.

[20] J. Shen. On the generalized "Lanczos' generalized derivative". *The American Mathematical Monthly*, **106**(8):766–768, 1999. https://doi.org/10.1080/00029890.1999.12005116.

[21] G.R.P. Teruel. A new class of generalized Lanczos derivatives. *Palestine Journal of Mathematics*, **7**(1):211–221, 2018.

[22] Z. Wang and R. Wen. Numerical differentiation for high orders by an integration method. *Journal of Computational and Applied Mathematics*, **234**(3):941–948, 2010. https://doi.org/10.1016/j.cam.2010.01.056.

Appendix A: Orthogonal polynomial approach

The approach based on the properties of orthogonal polynomials can be demonstrated on a kernel made up of two Legendre polynomials P_n

$$P\left(t\right)=\alpha P_{d}\left(t\right)+\beta P_{d+2}\left(t\right),$$

which is meant to extract the dth derivative (α and β are real coefficients). The $P_{d+1}(t)$ term is skipped for the same reasons as in the main text: this term controls the precision at the $\mathcal{O}\left(h^{1}\right)$ level and ignoring it is the easiest way for constructing a higher-order method (which further generalizes to all P_{d+2i+1} terms and odd powers of h). Assuming the analyticity of f on $(x_0-h-\epsilon,x_0+h+\epsilon)$, $\epsilon,h>0$, one can perform an expansion similar to (1.3)

$$\begin{aligned}\left(\frac{-1}{h}\right)^{d}\int_{-1}^{1}P\left(t\right)f\left(x_0+ht\right)dt=\alpha\left(\frac{-1}{h}\right)^{d}\sum_{n=0}^{\infty}\frac{1}{n!}h^{n}f^{(n)}\int_{-1}^{1}P_{d}\left(t\right)t^{n}dt\\+\beta\left(\frac{-1}{h}\right)^{d}\sum_{n=0}^{\infty}\frac{1}{n!}h^{n}f^{(n)}\int_{-1}^{1}P_{d+2}\left(t\right)t^{n}dt.\end{aligned}$$

Defining $a_n\equiv\int_{-1}^{1}P_d\left(t\right)t^{n}dt$, $b_n\equiv\int_{-1}^{1}P_{d+2}\left(t\right)t^{n}dt$ one notices that $a_{d+1+2i}=b_{d+3+2i}=0$ ($i=0,1,\dots$). This follows from the fact that the term containing a_d is (by construction of the DbI method) proportional to the derivative (which is nonzero in general) while in higher terms the t^n factor alternates between odd and even parity (with P_d being constant and strictly odd or even). Isolating few first terms in the sums and rearranging them leads to

$$\left(\frac{-1}{h}\right)^{d}\int_{-1}^{1}P\left(t\right)f\left(x_{0}+ht\right)dt=\alpha\frac{\left(-1\right)^{d}}{d!}a_{d}f^{(d)}$$
$$+\frac{\left(-1\right)^{d}}{(d+2)!}\left[\alpha a_{d+2}+\beta b_{d+2}\right]h^{2}f^{(d+2)}+\mathcal{O}\left(h^{4}\right).$$

By setting $\alpha=(-1)^{d}d!/a_{d}$ one gets the expected property of filtering the dth derivative and by defining $\beta=-\alpha a_{d+2}/b_{d+2}$ one arrives to a higher-order method where the h^{2} term is missing. Setting, for example, $d=3$ one has $a_{3}=4/35$, $a_{5}=8/63$ and $b_{5}=16/693$, from which follows $\alpha=-105/2$ and $\beta=1155/4$. The kernel function is then given by

$$-\frac{105}{2}P_{3}\left(t\right)+\frac{1155}{4}P_{5}\left(t\right)=\frac{945}{32}\left(77t^{5}-90t^{3}+21t\right),$$

which is equivalent to the expression of $k_{3}^{[2]}$ found in Appendix C.

The approach can be further generalized by writing

$$P\left(t\right)=\alpha P_{d}\left(t\right)+\beta P_{d+2}\left(t\right)+\gamma P_{d+4}\left(t\right)+\delta P_{d+6}\left(t\right)+\ldots.$$

Proceeding in a fully analogous way, it is trivial to see that by adjusting the coefficients α, β, γ, ... one can build derivative filters with higher and higher precision orders. In general a kernel function providing the precision order $\mathcal{O}\left(h^{2i}\right)$ is (independently on the derivative order) given by a linear combination of i Legendre polynomials (see formula (4.1) in [6]).

Appendix B: Example solution

In order to facilitate a possible implementation we present one explicit solution of the system (2.7). To avoid the simplest case, we choose an example with the order of precision $i=2$, i.e. the highest term to be annihilated is $h^{2i}=h^{4}$, see (2.6). For this order of precision let us find the expression for the first derivative, i.e. $d=1$. The number of equations is dictated by the inequality $1\leq j\leq i=2$, namely we have two equation with $j=1,2$. The general expression (2.7) thus takes form

$$E_{j=1}:\sum_{n=0}^{5}\frac{\alpha_{n}\left(a_{0},\ldots,a_{N}\right)}{n+4}\left[1+\left(-1\right)^{3+n}\right]=\frac{2}{5}\alpha_{1}+\frac{2}{7}\alpha_{3}+\frac{2}{9}\alpha_{5}=0,\qquad(4.1)$$

$$E_{j=2}:\sum_{n=0}^{5}\frac{\alpha_{n}\left(a_{0},\ldots,a_{N}\right)}{n+6}\left[1+\left(-1\right)^{5+n}\right]=\frac{2}{7}\alpha_{1}+\frac{2}{9}\alpha_{3}+\frac{2}{11}\alpha_{5}=0.\qquad(4.2)$$

Remembering that the numbering of coefficients stops at $N=2i=4$ (see formula (2.9)) the rules (2.8) dictate $a_{0}=1$, $a_{1}=0$ and $a_{3}=0$ while the values of a_{2} and a_{4} remain unknown, to be determined by solving the system.

From (2.5) and (2.2) one writes α_1

$$\begin{aligned}\alpha_1 = 2\,\Omega_{1+1} = 2 \sum_{k=0}^{2} \frac{1}{2} H_k H_{4-k} \left\{1+(-1)^{2-k}\right\} C_{1,\frac{2-k}{2}}\,(-1)^{(2-k)/2}\,a_k \\ = H_0H_4\left\{1+(-1)^2\right\}C_{1,\frac{2}{2}}\,(-1)^{2/2}\,a_0 + H_1H_3\left\{1+(-1)^1\right\}C_{1,\frac{1}{2}}\,(-1)^{1/2}\,a_1 \\ + H_2H_2\left\{1+(-1)^0\right\}C_{1,\frac{0}{2}}\,(-1)^{0/2}\,a_2 = -2C_{1,1}+2C_{1,0}a_2 = 2a_2-2.\end{aligned}$$

Proceeding in an analogous way one gets

$$\alpha_3 = 4\,\Omega_{3+1} = -4a_2 + 4a_4 \text{ and } \alpha_5 = 6\,\Omega_{5+1} = -6a_4.$$

Plugging these expressions into (4.1)–(4.2) one gets a system of two linear equations

$$\begin{aligned}\frac{2}{5}(2a_2-2) + \frac{2}{7}(-4a_2+4a_4) + \frac{2}{9}(-6a_4) = 0, \\ \frac{2}{7}(2a_2-2) + \frac{2}{9}(-4a_2+4a_4) + \frac{2}{11}(-6a_4) = 0,\end{aligned}$$

which leads to $a_2 = -6$ and $a_4 = 33/5$. The polynomial $\omega(x)$ from Equation (2.1) takes the form $\omega(x) = \mathcal{N}(1-x^2)(1-6x^2+\frac{33}{5}x^4)$. The computation of the normalization constant (2.3) is straightforward

$$\mathcal{N} = \left(\sum_{n=0}^{\lfloor 6/2 \rfloor} \frac{2\Omega_{2n}}{2n+1}\right)^{-1} = \left(2\Omega_0 + \frac{2}{3}\Omega_2 + \frac{2}{5}\Omega_4 + \frac{2}{7}\Omega_6\right)^{-1},$$

where we already know the values $\Omega_2 = a_2 - 1 = -7$, $\Omega_4 = a_4 - a_2 = 63/5$ and $\Omega_6 = -a_4 = -33/5$. The value of Ω_0 is

$$\Omega_0 = \sum_{k=-2}^{0} \frac{1}{2} H_k H_{4-k}\left\{1+(-1)^{-k}\right\} C_{1,-k/2}\,(-1)^{-k/2}\,a_k = H_0H_4C_{1,0}\,a_0 = 1.$$

Thus $\mathcal{N} = \left(2 - \frac{14}{3} + \frac{126}{25} - \frac{66}{35}\right)^{-1} = \frac{525}{256}$, so recovering the formula for $\omega_1^{[4]}(x)$. The kernel function $k(x)$ is obtained from $\omega(x)$ as its dth derivative (Equation (1.4)).

Appendix C: List of weight and kernel functions

List of higher accuracy order weight and kernel functions. Kernel functions k are to be used in formula (1.7), notation from the end of Section 2 is adopted.

- First derivative

$$\omega_1^{[2]}(x) = \frac{45}{32}\left(1-x^2\right)\left(1-\frac{7}{3}x^2\right), \quad k_1^{[2]}(x) = \frac{15}{8}\left(7x^3-5x\right),$$

$$\omega_1^{[4]}(x) = \frac{525}{256}\left(1-x^2\right)\left(1-6x^2+\frac{33}{5}x^4\right),$$

$$k_1^{[4]}(x) = -\frac{105}{128}\left(99x^5-126x^3+35x\right),$$

$$
\begin{aligned}
\omega_1^{[6]}(x) &= \frac{11025}{4096}\left(1-x^2\right)\left(1-11x^2+\frac{143}{5}x^4-\frac{143}{7}x^6\right),\\
k_1^{[6]}(x) &= \frac{315}{512}\left(715x^7-1287x^5+693x^3-105x\right),\\
\omega_1^{[8]}(x) &= \frac{218295}{65536}\left(1-x^2\right)\left(1-\frac{52}{3}x^2+78x^4-\frac{884}{7}x^6+\frac{4199}{63}x^8\right),\\
k_1^{[8]}(x) &= -\frac{3465}{32768}(20995x^9-48620x^7+38610x^5-12012x^3+1155),\\
\omega_1^{[10]}(x) &= \frac{2081079}{524288}\left(1-x^2\right)\\
&\quad\times\left(1-25x^2+170x^4-\frac{3230}{7}x^6+\frac{1615}{3}x^8-\frac{7429}{33}x^{10}\right),\\
k_1^{[10]}(x) &= \frac{9009}{131072}(156009x^{11}-440895x^9+461890x^7\\
&\quad-218790x^5+45045x^3-3003x).
\end{aligned}
$$

- Second derivative

$$
\begin{aligned}
\omega_2^{[2]}(x) &= \frac{105}{64}\left(1-x^2\right)^2\left(1-3x^2\right),\ k_2^{[2]}(x)=-\frac{105}{32}\left(45x^4-42x^2+5\right),\\
\omega_2^{[4]}(x) &= \frac{4725}{2048}\left(1-x^2\right)^2\left(1-\frac{22}{3}x^2+\frac{143}{15}x^4\right),\\
k_2^{[4]}(x) &= \frac{315}{256}\left(1001x^6-1485x^4+567x^2-35\right),\\
\omega_2^{[6]}(x) &= \frac{24255}{8192}\left(1-x^2\right)^2\left(1-13x^2+39x^4-\frac{221}{7}x^6\right),\\
k_2^{[6]}(x) &= -\frac{3465}{4096}(9945x^8-20020x^6+12870x^4-2772x^2+105),\\
\omega_2^{[8]}(x) &= \frac{945945}{262144}\left(1-x^2\right)^2\left(1-20x^2+102x^4-\frac{1292}{7}x^6+\frac{323}{3}x^8\right),\\
k_2^{[8]}(x) &= \frac{45045}{65536}(74613x^{10}-188955x^8+170170x^6\\
&\quad-64350x^4+9009x^2-231),\\
\omega_2^{[10]}(x) &= \frac{4459455}{1048576}\left(1-x^2\right)^2\\
&\quad\times\left(1-\frac{85}{3}x^2+\frac{646}{3}x^4-646x^6+\frac{7429}{9}x^8-\frac{37145}{99}x^{10}\right),\\
k_2^{[10]}(x) &= -\frac{45045}{524288}(3380195x^{12}-10296594x^{10}+11904165x^8\\
&\quad-6466460x^6+1640925x^4-162162x^2+3003).
\end{aligned}
$$

- Third derivative

$$
\begin{aligned}
\omega_3^{[2]}(x) &= \frac{945}{512}\left(1-x^2\right)^3\left(1-\frac{11}{3}x^2\right),\ k_3^{[2]}(x)=\frac{945}{32}\left(77x^5-90x^3+21x\right),\\
\omega_3^{[4]}(x) &= \frac{10395}{4096}\left(1-x^2\right)^3\left(1-\frac{26}{3}x^2+13x^4\right),\\
k_3^{[4]}(x) &= -\frac{10395}{256}\left(585x^7-1001x^5+495x^3-63x\right),
\end{aligned}
$$

$$\omega_3^{[6]}(x) = \frac{105105}{32768}\left(1-x^2\right)^3\left(1-15x^2+51x^4-\frac{323}{7}x^6\right),$$

$$k_3^{[6]}(x) = \frac{45045}{4096}(17765x^9-39780x^7+30030x^5-8580x^3+693x),$$

$$\omega_3^{[8]}(x) = \frac{2027025}{524288}\left(1-x^2\right)^3\left(1-\frac{68}{3}x^2+\frac{646}{5}x^4-\frac{1292}{5}x^6+\frac{7429}{45}x^8\right),$$

$$\begin{aligned}k_3^{[8]}(x) = -\frac{135135}{65536}&(676039x^{11}-1865325x^9+1889550x^7\\&-850850x^5+160875x^3-9009x),\end{aligned}$$

$$\begin{aligned}\omega_3^{[10]}(x) = \frac{75810735}{16777216}&\left(1-x^2\right)^3\\&\times\left(1-\frac{95}{3}x^2+266x^4-874x^6+\frac{10925}{9}x^8-\frac{6555}{11}x^{10}\right),\end{aligned}$$

$$\begin{aligned}k_3^{[10]}(x) = \frac{2297295}{524288}&(2064825x^{13}-6760390x^{11}+8580495x^9\\&-5290740x^7+1616615x^5-218790x^3+9009x).\end{aligned}$$

- Fourth derivative

$$\omega_4^{[2]}(x) = \frac{2079}{1024}\left(1-x^2\right)^4\left(1-\frac{13}{3}x^2\right),$$

$$k_4^{[2]}(x) = -\frac{10395}{64}\left(273x^6-385x^4+135x^2-7\right),$$

$$\omega_4^{[4]}(x) = \frac{45045}{16384}\left(1-x^2\right)^4\left(1-10x^2+17x^4\right),$$

$$k_4^{[4]}(x) = \frac{135135}{2048}(8415x^8-16380x^6+10010x^4-1980x^2+63),$$

$$\omega_4^{[6]}(x) = \frac{225225}{65536}\left(1-x^2\right)^4\left(1-17x^2+\frac{323}{5}x^4-\frac{323}{5}x^6\right),$$

$$\begin{aligned}k_4^{[6]}(x) = -\frac{135135}{8192}&(323323x^{10}-799425x^8+696150x^6\\&-250250x^4+32175x^2-693),\end{aligned}$$

$$\omega_4^{[8]}(x) = \frac{34459425}{8388608}\left(1-x^2\right)^4\left(1-\frac{76}{3}x^2+\frac{798}{5}x^4-\frac{1748}{5}x^6+\frac{2185}{9}x^8\right),$$

$$\begin{aligned}k_4^{[8]}(x) = \frac{328185}{613312}&(4970875x^{12}-14872858x^{10}+16787925x^8\\&-8698200x^6+1967525x^4-141750x^2+723),\end{aligned}$$

$$\begin{aligned}\omega_4^{[10]}(x) = \frac{160044885}{33554432}&\left(1-x^2\right)^4\\&\times\left(1-35x^2+322x^4-1150x^6+1725x^8-\frac{10005}{11}x^{10}\right),\end{aligned}$$

$$\begin{aligned}k_4^{[10]}(x) = -\frac{43648605}{1048576}&(7653825x^{14}-26842725x^{12}+37182145x^{10}\\&-25741485x^8+9258795x^6-1616615x^4+109395x^2-1287).\end{aligned}$$