# Higher accuracy order in differentiation-by-integration

Andrej Liptaj[*]

December 18, 2019

**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 particular derivative orders. A system of linear equations is formulated which allows to construct kernels with arbitrary precision for arbitrary derivative order. A study on a real computer is performed and it is shown that numerical differentiation based on these kernels performs much better (w.r.t errors) than the same procedure based on usual Legendre-polynomial kernels. Presented results may have implications for numerical implementations of the differentiation-by-integration method.

## 1 Introduction

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

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

where $k$ is an appropriate kernel function and $n$ indicates the derivative order. The method is know as "*differentiation by integration*" (DbI) and an example was first published by Cioranescu [1]. Later, Lanczos [2] studied a version of (1) with kernel

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

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]. The DbI also attracts attention for its potential applications in numerical analysis [4, 5, 6, 7, 8].

In published texts [9, 10, 11, 12, 13, 14] two dominant ideas can be seen:

[*] Institute of Physics, Bratislava, Slovak Academy of Sciences, andrej.liptaj@savba.sk

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

- The least-squares property: the derivative as defined by the Lanczos' formula (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 $[-h, h]$. The idea was further generalized to higher derivatives in [10] 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 [15] and related way of numerically computing a derivative from discrete data sets.

- Orthogonality approach: if one expands a function $f$ into Taylor series

$$\int_{-1}^{1} k\left(t\right) f\left(x_0 + ht\right) dt = \sum_{n=0}^{\infty} \frac{1}{n!} h^n f^{(n)}\left(x_0\right) \int_{-1}^{1} k\left(t\right) t^n dt, \tag{3}$$

  one can 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 $D$th derivative one can use a polynomial orthogonal to all $t^n$ for $n < D$ (e.g. 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 $D$th derivative. This approach can provide large generalizations (as presented in [13]) but remains limited to analytic functions $f$.

A universal procedure for constructing a kernel function for the first derivative was presented in [5], a full generalization to all derivative orders was given in [16]. A function $k$ is a valid kernel function for computing $D$th derivative if and only if

$$k = \frac{d^D}{dt^D} \omega\left(t\right), \tag{4}$$

where $\omega$ is a weight function satisfying

$$\int\limits_{-1}^{1} \omega\left(t\right) dt = 1, \tag{5}$$

$$\frac{d^m}{dt^m} \omega\left(t\right)|_{t=-1} = \frac{d^m}{dt^m} \omega\left(t\right)|_{t=+1} = 0 \text{ for all } m < D, \tag{6}$$

assuming that all appearing integrals and derivatives exist. The $D$th derivative is then computed as

$$\frac{d^D}{dt^D} f\left(x\right)|_{x=x_0} = \lim_{h\to 0} \mathcal{D}_D^h\left(f\right)|_{x_0} \equiv \lim_{h\to 0} \left(\frac{-1}{h}\right)^D \int_{-1}^{1} k\left(t\right) f\left(x_0 + ht\right) dt. \tag{7}$$

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 know that the higher-order approaches perform much better than the lower-order ones[2]. Systematic study of these effects is somewhat missing with respect to the DbI methods. Higher accuracy orders might break the “least-square” behavior, but lead (as demonstrated later) to largely improved overall numerical precision. In existing literature, formulas with higher accuracy order can be found (e.g. formula 4.1 in [13]), 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 (Sec. 2). 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 (Sec. 2). We also want to
- perform a numerical study of the higher accuracy order approach (Sec. 3).

We will briefly discuss the presented results in Sec. 4 and close the text by providing summary and conclusion (Sec. 5).

## 2 Higher order formulas

Different strategies can be adopted when constructing higher precision order kernels. Orthogonality approach is certainly an option: by an appropriate choice of a kernel $k$ one can set to zero all $h^n$ terms in (3) up to a desired value of $n < N$, with exception of the term containing the derivative one is interested in. Yet, we prefer to chose a different strategy which stems from basic principles (4), (5) and (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 full control over coefficients and leads to explicit relations between them.

A general polynomial respecting (6) is written as

$$\begin{aligned}\omega\left(x\right) &= \mathcal{N}\left(1-x\right)^{D}\left(1+x\right)^{D}\sum_{n=0}^{N}a_n x^n, \qquad (8)\\ &= \mathcal{N}\left[\sum_{j=0}^{2D}\frac{1}{2}\left\{1+\left(-1\right)^{j}\right\}C_{D,j/2}\left(-1\right)^{j/2}x^{j}\right]\sum_{n=0}^{N}a_n x^n,\end{aligned}$$

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

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

by the $1+(-1)^j$ term. The two sums can be multiplied

$$\omega\left(x\right)=\mathcal{N}\sum_{n=0}^{N+2D}\varOmega_n x^n, \tag{9}$$

where

$$\varOmega_n=\sum_{k=n-2D}^{n}\frac{1}{2}H_k H_{N-k}\left\{1+\left(-1\right)^{n-k}\right\}C_{D,(n-k)/2}\left(-1\right)^{(n-k)/2}a_k, \tag{10}$$

with $H$ indicating a step function

$$H_i=\begin{cases}0 & \text{if } i<0\\ 1 & \text{else}\end{cases}.$$

The condition (5) implies the normalization

$$\mathcal{N}=\left(\sum_{n=0}^{\lfloor(N+2D)/2\rfloor}\frac{2\varOmega_{2n}}{2n+1}\right)^{-1}, \tag{11}$$

where symbol $\lfloor\ldots\rfloor$ stands for the floor function. For the needs of formula (4) one has to compute higher-order derivatives

$$\omega^{(m)}=\mathcal{N}\sum_{n=0}^{N+2D-m}\frac{(n+m)!}{n!}\varOmega_{n+m}x^n\Rightarrow\omega^{(D)}=\mathcal{N}\sum_{n=0}^{N+D}\frac{(n+D)!}{n!}\varOmega_{n+D}x^n. \tag{12}$$

Expanding a function $f$ into Taylor series and plugging (12) into (7) using (4) one gets[3]

$$\begin{aligned}f^{(D)}&=\left(-\frac{1}{h}\right)^D\int_{-1}^{1}\mathcal{N}\sum_{n=0}^{N+D}\alpha_n t^n\sum_{m=0}^{\infty}\frac{1}{m!}h^m f^{(m)}t^m dt,\\ &=\mathcal{N}\left(-1\right)^D\sum_{m=0}^{\infty}\frac{1}{m!}f^{(m)}\left\{\sum_{n=0}^{N+D}\frac{\alpha_n}{n+m+1}\left[1+\left(-1\right)^{m+n}\right]\right\}h^{m-D},\\ &\text{with } \alpha_n\equiv\frac{(n+D)!}{n!}\varOmega_{n+D},\end{aligned}$$

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 (7) however implies that their coefficients are identically zero. Further more, formula (7) also implies[4] that the overall coefficient in front of the constant term ($h^0$) is $f^{(D)}$. The interesting terms are so those with $m>D$.

[3] Argument of $f$ is omitted, one implicitly assumes it takes value $x_0$ (point of differentiation).

[4] Both of these statements can be verified by in explicit computation (but do not need to be).

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

$$\begin{aligned} h^{m-D} \sum_{n=0}^{N+D} \frac{\alpha_n \left[1+(-1)^{m+n}\right]}{n+m+1} &\sim \sum_{n=0}^{N+D} h^{m+n-(D+n)} \varOmega_{n+D} \left[1+(-1)^{m+n}\right] \\ &= \sum_{n=0}^{N+D} h^{m+n-Z_n} \varOmega_{Z_n} \left[1+(-1)^{m+n}\right] \\ &\sim \sum_{n=0}^{N+D} h^{2i_n - Z_n} \varOmega_{Z_n} = \sum_{n=0}^{N+D} h^{Z_n + 2i_n} \varOmega_{Z_n}, \end{aligned}$$

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

$$\begin{aligned} \varOmega_{2i+1} &\sim \sum_{k=n-2D}^{n} \left\{1+(-1)^{2i+1-k}\right\} (-1)^{(2i+1-k)/2} a_k, \quad 0<k<N \\ &\sim \sum_{k=n-2D}^{n} \left\{1+(-1)^{1+k}\right\} (-1)^{(2i+1-k)/2} a_k. \end{aligned}$$

If $k$ is even, then $1+k$ is odd and $1+(-1)^{1+k}$ is zero for all related $a_k$ terms. Therefore $\varOmega_{2i+1}$ depends only on the odd-index $a_i$ coefficients. This being established, in what follows we will study only even-index based weight functions[5], i.e. even function 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 (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

$$\begin{aligned} \mathcal{N} \frac{(-1)^D}{m!} f^{(m)} \left\{ \sum_{n=0}^{N+D} \frac{\alpha_n(a_k)}{n+m+1} \left[1+(-1)^{m+n}\right] \right\} h^{m-D} \\ = \mathcal{D}^h_{n=D}(f) = f^{(D)} + c_2 h^2 + c_4 h^4 + c_6 h^6 + \ldots \end{aligned}$$

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 $j$th equation can be written $(1 \le j \le i)$

$$\mathrm{E}_j : \qquad \sum_{n=0}^{2i+D} \frac{\alpha_n(\varDelta[a_k])}{n+2j+D+1} \left[1+(-1)^{2j+D+n}\right] = 0, \tag{13}$$

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

where the dependence of $\alpha_n$ on coefficients is

$$\Delta\left[a_k\right]=\begin{cases}1 & \text{if } k=0\\ 0 & \text{if } 1\le k\le 2i \text{ and } k \text{ is odd}\\ a_k & \text{if } 1\le k\le 2i \text{ and } k \text{ is even}\\ 0 & \text{else}\end{cases}$$

and equality

$$N=2i$$

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 (11) and constructs the weight function (8). The kernel function is given as its $D$th derivative (4).

We will note $\omega_D^{[2i]}$ and $k_D^{[2i]}$ those functions, which provide $D$th 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 $2i+2$. Solving system (13) one gets as example results

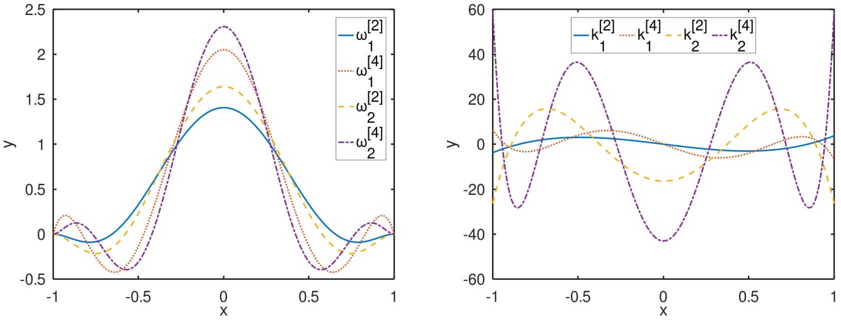
$$\omega_1^{[2]}(x)=\frac{45}{32}(1-x)(1+x)\left(1-\frac{7}{3}x^2\right),$$
$$k_1^{[2]}(x)=\frac{1}{8}\left(105x^3-75x\right),$$

$$\omega_1^{[4]}(x)=\frac{525}{256}(1-x)(1+x)\left(1-6x^2+\frac{33}{5}x^4\right),$$
$$k_1^{[4]}(x)=-\frac{1}{128}\left(10395x^5-13230x^3+3675x\right),$$

$$\omega_2^{[2]}(x)=\frac{105}{64}(1-x)^2(1+x)^2\left(1-3x^2\right),$$
$$k_2^{[2]}(x)=-\frac{1}{32}\left(4725x^4-4410x^2+525\right),$$

$$\omega_2^{[4]}(x)=\frac{4725}{2048}(1-x)^2(1+x)^2\left(1-\frac{22}{3}x^2+\frac{143}{15}x^4\right),$$
$$k_2^{[4]}(x)=\frac{1}{256}\left(315315x^6-467775x^4+178605x^2-11025\right).$$

The functions are depicted in Fig. (1). All cases with $1\le D\le 4$ and $2\le 2i\le 10$ are listed in Appendix.



(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 numerical study[6], rather large and multidimensional. One could by interested in the dependence of results on the

- size of the discretization parameter $h$, or on the
- precision order of the kernel function, 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 3 test functions with fixed point of differentiation $x_0$

$$\sin(x) \ \text{ with } x_0 = 1; \quad \exp(x) \ \text{ with } x_0 = \pi \text{ and } \ln(x) \ \text{ with } x_0 = \frac{1}{2}.$$

Using these settings we scan 8 orders of magnitude in discretization parameter $10^{-8} \leq h \leq 10^{-1}$, changing the step in geometrical progression with factor 10. For comparison purposes we add results from the least-squares (LS) approach, which is based on the Legendre-polynomial kernels.

As seen from the summary Table (1), the optimal $h$ for higher precision order (HO) methods is $h \sim 10^{-1}, 10^{-2}$, for the LS methods it is systematically smaller[7] $h \sim 10^{-3}, 10^{-4}$. The important observation is that the HO kernels significantly over-perform the LS kernels (if each method uses its own optimal

[6] Numerical analysis was done with the GNU Octave [17] software 5.1.0 configured for "x86_64-suse-linux-gnu". The *quadl* function was used for numerical integration. Author also greatly profited from the WxMaxima tools [18].

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

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 are shown in Fig. (2). Analyzing them, one can draw a straightforward conclusion: when interested in absolute errors, the HO differentiation provide several orders of magnitude more precise results than the LS methods and, when it comes to implementing a numerical differentiation-by-integration method on a computer, the approach based on the HO kernels is a clearly the preferred option.

# 4 Discussion

An interesting lesson can be learnt from previous observations: if a function-related quantity $Q$ is to be extracted in a numerical way from the function behavior on the interval $[-h, h]$, where $h$ is small enough and can play a role of a power-expansion parameter, then the "least-squares" optimum does not need to be the "$Q$-extraction" optimum. The results suggest, that a more effective way might consist in expanding $Q$ into powers of $h$: $Q = q_0 + q_1 h + q_2 h^2 + \ldots$ and finding a method which reaches a higher precision with respect to this expansion.

One should also address the question of numerical (i.e. round-off) errors. This text deals with the so-called discretization error which can be made arbitrary small and can be handled with full mathematical rigorosity. However, on a real computer the use of a (very) high-accuracy approach would fail: as seen from Fig. (1) higher-order weight functions become more sharply peaked near zero which can, in some sense, be interpreted as effective shrinking of the interval $[-h, h]$. If integral becomes small, numerical errors grow. In addition, higher-order weight functions (and thus kernel functions too) become more and more oscillatory, which further worsens numerical uncertainties. One can also notice, that with increasing precision order kernel function become numerically large.

Numerical errors are difficult to handle: a reliable mathematical treatment of the float-point arithmetic in computer registers is presumably a very hard task. In principle one cannot exclude a different behavior of numerical errors for HO and LS approaches, but such differences (if existing and significant) are hard to estimate in an exact way. Rather than trying to quantify this issue, a trustful (we believe) conclusion can be deduced from the numerical tests in Sec. 3. Whatever the behavior of numerical errors is, the HO differentiation provides in studied examples a significantly smaller overall (total) error. It is reasonable to assume that this is a standard behavior[8] of the HO method when compared to the LS approach.

[8] HO method having smaller discretization error, one judges improbable for the HO approach to have such big numerical error (in copmarison to LS outputs) so as to have a larger total error.

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

Table 1: Absolute error for 4 derivatives orders ($D$), two kernel-function types (higher order $HO$ vs. least-squares $LS$) and three test functions with fixed test points. Optimal value of the discretization parameter $h$ is tuned for one of the methods (in bold), the the result of other method is shown also.

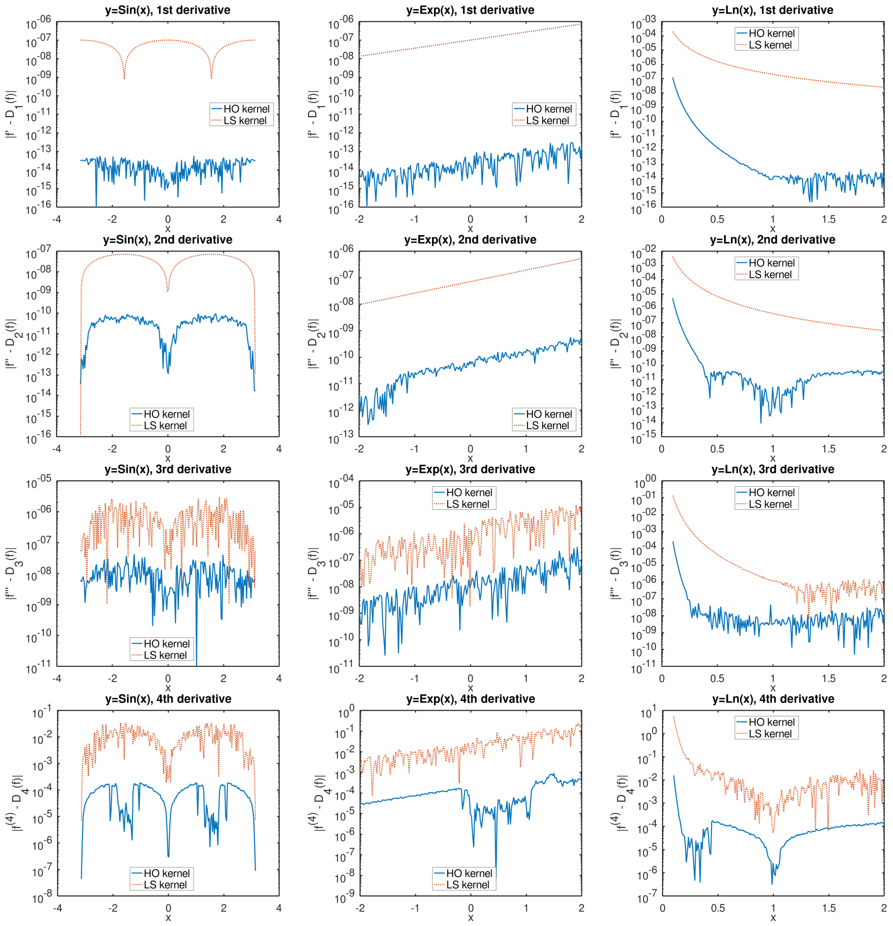


Figure 2: Absolute errors of the HO approach (full blue line) and LS-based methods (dotted red line) for 3 functions (in columns) and 4 derivative orders (rows).

# 5 Summary and conclusion

By using straightforward calculations we constructed a system of linear equations (13) 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 based kernels, which were most widely used and discussed up to now. If a DbI method is to be implemented on a computer, then the results provide strong evidence for preferring higher precision order kernels. The text also opens a general question of extracting a function-related quantity from behavior of the function on a small interval and promotes a method of higher precision order linked to a small parameter (interval length) expansion.

# References

[1] N. Cioranescu, Enseign. Math. 37 (1938) 292–302.

[2] C. Lanczos, Applied Analysis, Prentic-Hall, Englewood Clis, 1956.

[3] D.-Y. Liu, O. Gibaru, W.Perruquetti, T.-M. Laleg-Kirati, IEEE Transactions on Automatic Control 60(11) (2015) 2945 - 2960.

[4] C. W. Groetsch, Amer. Math. Monthly 105 (1998) 320-326.

[5] D. L. Hicks, L. M. Liebrock, Appl. Math. Comput. 112 (2000) 63–73.

[6] Z. Wang, R. Wen, J. Comput. Appl. Math. 234 (2010) 941–948.

[7] T. J. McDevitt , International Journal of Computer Mathematics 89(7) (2012) 916-931.

[8] X. Huang, Ch. Wu and J. Zhou, Math. Comp. 83 (2014), 789-807.

[9] J. Shen, Amer. Math. Monthly 106(8) (1999) 766-768.

[10] N. Burch, P. Fishback and R. Gordon, Mathematics Magazine, 78(5), (2005) 368-378.

[11] S.K. Rangarajana, S.P. Purushothaman, J. Comput. Appl. Math. 177 (2005) 461465.

[12] DY Liu, O. Gibaru, W. Perruquetti, J. Comput. Appl. Math. 235 (2011) 3015-3032.

[13] E. Diekema, T. H. Koornwinder, J. of Approximation Theory 164 (2012) 637-667.

[14] G. R. P. Teruel, Palestine J. of Math. 7(1) (2018) , 211–221.

[15] A. Savitzky and M. J. E. Golay, Analytical Chemistry 36 (8) (1964 )1627-1639.

[16] A. Liptaj, arXiv:1906.04921, (2019).

[17] J. W. Eaton, D. Bateman, S. Hauberg and R. Wehbring, GNU Octave (version 5.1.0 manual) (2019).

[18] Maxima, A Computer Algebra System. Version 5.34.1 (2014).

# Appendix

List of higher accuracy order weight and kernel functions. Kernel functions $k$ are to be used in formula (7), notation from the end of Sec. 2 is adopted.

- First derivative

$$\begin{aligned}\omega_1^{[2]}(x) &= \frac{45}{32}(1-x)(1+x)\left(1-\frac{7}{3}x^2\right)\\ k_1^{[2]}(x) &= \frac{1}{8}\left(105x^3-75x\right)\end{aligned}$$

$$\begin{aligned}\omega_1^{[4]}(x) &= \frac{525}{256}(1-x)(1+x)\left(1-6x^2+\frac{33}{5}x^4\right)\\ k_1^{[4]}(x) &= -\frac{1}{128}\left(10395x^5-13230x^3+3675x\right)\end{aligned}$$

$$\begin{aligned}\omega_1^{[6]}(x) &= \frac{11025}{4096}(1-x)(1+x)\left(1-11x^2+\frac{143}{5}x^4-\frac{143}{7}x^6\right)\\ k_1^{[6]}(x) &= \frac{1}{512}\left(225225x^7-405405x^5+218295x^3-33075x\right)\end{aligned}$$

$$\begin{aligned}\omega_1^{[8]}(x) &= \frac{218295}{65536}(1-x)(1+x)\left(1-\frac{52}{3}x^2+78x^4-\frac{884}{7}x^6+\frac{4199}{63}x^8\right)\\ k_1^{[8]}(x) &= -\frac{1}{32768}\left(72747675x^9-168468300x^7+133783650x^5-41621580x^3+4002075x\right)\end{aligned}$$

$$\begin{aligned}\omega_1^{[10]}(x) &= \frac{2081079}{524288}(1-x)(1+x)\left(1-25x^2+170x^4-\frac{3230}{7}x^6+\frac{1615}{3}x^8-\frac{7429}{33}x^{10}\right)\\ k_1^{[10]}(x) &= \frac{1}{131072}(1405485081x^{11}-3972023055x^9+4161167010x^7-1971079110x^5\\ &\quad +405810405x^3-27054027x)\end{aligned}$$

- Second derivative

$$\begin{aligned}\omega_2^{[2]}(x) &= \frac{105}{64}(1-x)^2(1+x)^2\left(1-3x^2\right)\\ k_2^{[2]}(x) &= -\frac{1}{32}\left(4725x^4-4410x^2+525\right)\end{aligned}$$

$$\begin{aligned}\omega_2^{[4]}(x) &= \frac{4725}{2048}(1-x)^2(1+x)^2\left(1-\frac{22}{3}x^2+\frac{143}{15}x^4\right)\\ k_2^{[4]}(x) &= \frac{1}{256}\left(315315x^6-467775x^4+178605x^2-11025\right)\end{aligned}$$

$$\omega_2^{[6]}(x) = \frac{24255}{8192}(1-x)^2(1+x)^2\left(1-13x^2+39x^4-\frac{221}{7}x^6\right)$$

$$k_2^{[6]}(x) = -\frac{1}{4096}\left(34459425x^8-69369300x^6+44594550x^4-9604980x^2+363825\right)$$

$$\omega_2^{[8]}(x) = \frac{945945}{262144}(1-x)^2(1+x)^2\left(1-20x^2+102x^4-\frac{1292}{7}x^6+\frac{323}{3}x^8\right)$$

$$\begin{aligned} k_2^{[8]}(x) = \frac{1}{65536}(&3360942585x^{10}-8511477975x^8+7665307650x^6-2898645750x^4\\ &+405810405x^2-10405395)\end{aligned}$$

$$\omega_2^{[10]}(x) = \frac{4459455}{1048576}(1-x)^2(1+x)^2\left(1-\frac{85}{3}x^2+\frac{646}{3}x^4-646x^6+\frac{7429}{9}x^8-\frac{37145}{99}x^{10}\right)$$

$$\begin{aligned} k_2^{[10]}(x) = -\frac{1}{524288}(&152260883775x^{12}-463810076730x^{10}+536223112425x^8\\ &-291281690700x^6+73915466625x^4-7304587290x^2+135270135)\end{aligned}$$

- Third derivative

$$\omega_3^{[2]}(x) = \frac{945}{512}(1-x)^3(1+x)^3\left(1-\frac{11}{3}x^2\right)$$

$$k_3^{[2]}(x) = \frac{1}{32}\left(72765x^5-85050x^3+19845x\right)$$

$$\omega_3^{[4]}(x) = \frac{10395}{4096}(1-x)^3(1+x)^3\left(1-\frac{26}{3}x^2+13x^4\right)$$

$$k_3^{[4]}(x) = -\frac{1}{256}\left(6081075x^7-10405395x^5+5145525x^3-654885x\right)$$

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

$$k_3^{[6]}(x) = \frac{1}{4096}\left(800224425x^9-1791890100x^7+1352701350x^5-386486100x^3+31216185x\right)$$

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

$$\begin{aligned} k_3^{[8]}(x) = -\frac{1}{65536}(&91356530265x^{11}-252070693875x^9+255344339250x^7\\ &-114979614750x^5+21739843125x^3-1217431215x)\end{aligned}$$

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

$$\begin{aligned} k_3^{[10]}(x) = \frac{1}{524288}(&4743512148375x^{13}-15530610145050x^{11}+19711928261025x^9\\ &-12154390548300x^7+3713841556425x^5-502625173050x^3+20696330655x)\end{aligned}$$

- Fourth derivative

$$
\begin{aligned}
\omega_4^{[2]}(x) &= \frac{2079}{1024}(1-x)^4(1+x)^4\left(1-\frac{13}{3}x^2\right)\\
k_4^{[2]}(x) &= -\frac{1}{64}\left(2837835x^6-4002075x^4+1403325x^2-72765\right)
\end{aligned}
$$

$$
\begin{aligned}
\omega_4^{[4]}(x) &= \frac{45045}{16384}(1-x)^4(1+x)^4\left(1-10x^2+17x^4\right)\\
k_4^{[4]}(x) &= \frac{1}{2048}\left(1137161025x^8-2213511300x^6+1352701350x^4-267567300x^2+8513505\right)
\end{aligned}
$$

$$
\begin{aligned}
\omega_4^{[6]}(x) &= \frac{225225}{65536}(1-x)^4(1+x)^4\left(1-17x^2+\frac{323}{5}x^4-\frac{323}{5}x^6\right)\\
k_4^{[6]}(x) &= -\frac{1}{8192}(43692253605x^{10}-108030297375x^8+94074230250x^6\\
&\quad -33817533750x^4+4347968625x^2-93648555)
\end{aligned}
$$

$$
\begin{aligned}
\omega_4^{[8]}(x) &= \frac{34459425}{8388608}(1-x)^4(1+x)^4\left(1-\frac{76}{3}x^2+\frac{798}{5}x^4-\frac{1748}{5}x^6+\frac{2185}{9}x^8\right)\\
k_4^{[8]}(x) &= \frac{1}{613312}(1631366611875x^{12}-4881048902730x^{10}+5509545166125x^8\\
&\quad -2854618767000x^6+645712192125x^4-46520223750x^2+237277755)
\end{aligned}
$$

$$
\begin{aligned}
\omega_4^{[10]}(x) &= \frac{160044885}{33554432}(1-x)^4(1+x)^4\left(1-35x^2+322x^4-1150x^6+1725x^8-\frac{10005}{11}x^{10}\right)\\
k_4^{[10]}(x) &= -\frac{1}{1048576}(334078784164125x^{14}-1171647500648625x^{12}+1622948760157725x^{10}\\
&\quad -1123579910878425x^8+404133485730975x^6-70562989572075x^4\\
&\quad +4774939143975x^2-56175754635)
\end{aligned}
$$