J. Approx. Softw., Vol. 1, No. 1 (2024) 3

Journal of Approximation Software

NDED - Numerical derivatives from equispaced data

N. Egidi [1,⋆], J. Giacomini [1], P. Maponi [1], and M. Youssef[1]

[1] *Scuola di Scienze e Tecnologie, Università di Camerino*

Received: 28/04/2023 – Published: 23/07/2024
Communicated by: R. Cavoretto and A. De Rossi

Abstract

Procedure NDED computes the numerical derivatives of order ν from equispaced data. This is based on the iterated application of a spectral algorithm for the computation of the first order derivative. A preliminary test of the procedure gives satisfactory results.

Keywords: numerical derivative, integral equation, singular value expansion, FFT (MSC2020: 45C05, 65-04, 65R20)

1 Introduction

A numerical differentiation problem consists in the computation of the derivative of order ν of an unknown function from the knowledge of the values of the function at prescribed points. Numerical differentiation is an interesting topic in many fields of applied sciences, such as biology, chemistry and physics, and it has a fundamental role in numerical analysis [3], [5], [21], [23]. For instance, operators approximating derivatives can be used to numerically solve differential equations [12], [13]. Due to its central role in scientific computing, several numerical differentiation methods are present in the scientific literature [4], [6], [7], [14], [15], [22], [24], [25]. All the methods for numerical differentiation are generally classified into these categories: finite difference methods, interpolation methods, regularization methods and integral methods. Finite difference methods and interpolation methods are well known and have the advantage of simplicity, moreover, they are considered to give satisfactory results when the function to be differentiated is given very precisely. Most of the regularization methods make use of the variational approach. The derivative is written as the solution of a Volterra integral equation and the resulting integral equation is reduced to a well-posed problem that depends on a regularization parameter. The main issue with these methods is the determination of the optimal parameter value that is generally a nontrivial task. Other interesting problems in the

⋆ Corresponding author: nadaniela.egidi@unicam.it

field of numerical differentiation are the differentiation of multivariate functions [2], [20] and the numerical differentiation from scattered data [8].

The main difficulty in the numerical differentiation is that small errors in the function data may cause large errors in the computed derivative, due to the unboundedness of the derivative operator. However, in practice, data are almost always corrupted by noise and in many applications it is necessary to estimate the derivative from known noisy data. Thus, proper regularization schemes are usually considered by methods for numerical differentiation [1], [16], [17], [18], [19], [26].

We present the procedure NDED for the numerical approximation of the derivatives of order $\nu \geq 1$. This procedure is based on a recursive application of an algorithm to compute first order derivatives from the Singular Value Expansion (SVE) of the derivative operator. In the present version, the procedure is intended for equispaced univariate data but the structure of the algorithm is easily generalizable to the cases of irregular grid spacing and multivariate data. The procedure NDED has been implemented in MATLAB and the "code metadata" are

Current code version	v. 1.0
Permanent link to repository	https://github.com/josgiac/NumDer.git
Code versioning system used	git
Software code languages	MATLAB

In Section 2, we summarize the theoretical basis of the proposed algorithm. In Section 3, we give the algorithm and its implementation in MATLAB. In Section 4, we show some numerical results. In Section 5, we provide conclusions and future developments.

2 Theoretical Background

The proposed algorithm is based on papers [9], [10] and [11], which we summarize in this section, for the reader's convenience.

We consider differentiable functions $f : I \to \mathbb{R}$ defined on a closed interval I; without losing generality, we can assume that $I = [0,1]$. The first derivative $f^{(1)}$ of f is the unique solution $w : I \to \mathbb{R}$ of

$$\int_0^1 K(t,s)w(s)ds = f(t) - f(0), \quad t \in [0,1], \tag{1}$$

which is a Volterra integral equation of first kind having kernel, $K : I \times I \to \mathbb{R}$,

$$K(t,s) = \begin{cases} 1, & 0 \leq s < t \leq 1, \\ 0, & 0 \leq t \leq s \leq 1. \end{cases} \tag{2}$$

We note that, with K defined as in (2), integral equation (1) is a direct consequence of the Fundamental Theorem of Calculus, that is, for each $t \in [0,1]$,

$$\int_0^t 1 \cdot f'(s)ds = f(t) - f(0).$$

In compact notation, equation (1) becomes

$$\mathscr{K} w = f - f_0, \tag{3}$$

where $f_0 = f(0)$ and $\mathscr{K}$ is the integral operator with kernel defined by (2). This integral operator $\mathscr{K}$ associated with the first order derivatives has a known SVE given by the following theorem.

Theorem 1. The SVE of kernel (2) is

$$K(t,s) = \sum_{k=0}^{\infty} \mu_k u_k(t) v_k(s), \quad t,s \in I, \tag{4}$$

where $\mu_k = \frac{2}{(2k+1)\pi}$, $k = 0,1,\ldots,$ are the singular values of $\mathscr{K}$ and the singular functions corresponding to μ_k are

$$v_k(s) = \sqrt{2}\cos\left(\frac{s}{\mu_k}\right), \qquad s \in I, \tag{5}$$

$$u_k(t) = \sqrt{2}\sin\left(\frac{t}{\mu_k}\right), \qquad t \in I. \tag{6}$$

Proof. See [11] for a detailed proof.□

The SVE of $\mathscr{K}$ allows the definition of an FFT method to compute the numerical derivatives of a given function starting from its values at prescribed points. Let $n > 0$ and $h = 1/n$, supposing that we know the values of f at $n+1$ equispaced points

$$\xi_j = jh,\ j = 0,1,\ldots,n, \tag{7}$$

that is $f_j = f(\xi_j)$ are known, then the following theorem gives such a method and the corresponding accuracy properties. This theorem considers the approximation of $f'(x_j)$, $j = 0,1,\ldots,n-1$ where

$$x_j = \left(j + \frac{1}{2}\right) h. \tag{8}$$

Moreover, the following notations are used: for $j,k = 0,1,\ldots,n-1$:

$$\gamma_k = \frac{1}{\mu_k}, \qquad \tilde{s}_{j,k} = \sin\left(\gamma_j \xi_k\right) \tag{9}$$

$$c_{j,k} = \cos\left(\gamma_j x_k\right), \qquad s_{j,k} = \sin\left(\gamma_j x_k\right). \tag{10}$$

Theorem 2. For $k = 0,1,\ldots,n-1$,

$$f'(x_k) = f_k^p + O(h^4), \tag{11}$$

where

$$f_k^p = \sqrt{2}\sum_{j=0}^{n-1} f_{v,j}^p c_{j,k}, \tag{12}$$

and for $j = 0,1,\ldots,n-1$,

$$f_{v,j}^p = \alpha_0 c_{j,0} + \beta_j (27 s_{j,0} - s_{j,1}) + \alpha_n c_{j,n}, \tag{13}$$

$$\alpha_0 = \frac{\sqrt{2}}{24}(2f_0 - 5f_1 + 4f_2 - f_3), \tag{14}$$

$$\alpha_n = \frac{\sqrt{2}}{24}(-f_{n-3} + 4f_{n-2} - 7f_{n-1} + 4f_n), \tag{15}$$

$$\beta_j = \frac{\sqrt{2}}{24}\left(2\sum_{l=1}^{n-1}(f_l - f_0)\tilde{s}_{j,l} + (-1)^j (f_n - f_0)\right). \tag{16}$$

Proof. See [10] for details.□

We note that (12) can be computed by the Discrete Fourier Transformation (DFT) of a vector that depends on $f^p_{\nu,j}$, $j=0,1,\dots,n-1$. Moreover, $f^p_{\nu,j}$, $j=0,1,\dots,n-1$, depend only on the data f_k, $k=0,1,\dots,n$, and (16) can be computed by the DFT. In particular, by using Theorem 2 and the FFT algorithm, in [10] we give two algorithms *FOD* and *NOD*. Algorithm *FOD* allows calculating the numerical derivative of order 1 by knowing the values of the function in equally spaced points of a closed interval $[a,b]$, while *NOD* computes the numerical derivative of order $\nu \geq 1$ by using *FOD* iteratively.

In the next section, we propose a revised version of FOD algorithm, that in some cases provides more accurate results than the original one.

3 The algorithm

We propose a new algorithm for numerical differentiation based on the following formulas. The algorithm has been coded in MATLAB, the current code version (v. 1.0) is available at https://github.com/josgiac/NumDer.git where the git code versioning system is used.

Let $f: I \to \mathbb{R}$ be a sufficiently regular function, by following the proof of Theorem 3.2 in [10], we can prove that, for $k=0,1,\dots,n-1$,

$$f'(x_k) = \tilde{f}^p_k + O(h^4), \tag{17}$$

where

$$\tilde{f}^p_k = \sqrt{2}\sum_{l=0}^{n-1} \tilde{f}^p_{\nu,l} c_{l,k}, \tag{18}$$

and for $j=0,1,\dots,n-1$,

$$\tilde{f}^p_{\nu,j} = \tilde{\alpha}_0 c_{j,0} + \beta_j(27 s_{j,0} - s_{j,1}) + \tilde{\alpha}_n c_{j,n}, \tag{19}$$

$$\tilde{\alpha}_0 = \frac{\sqrt{2}}{1920}(311 f_0 - 1075 f_1 + 1510 f_2 - 1110 f_3 + 435 f_4 - 71 f_5), \tag{20}$$

$$\tilde{\alpha}_n = \frac{\sqrt{2}}{1920}(471 f_n - 1235 f_{n-1} + 1510 f_{n-2} - 1110 f_{n-3} + +435 f_{n-4} - 71 f_{n-5}), \tag{21}$$

where we recall that β_j, $j=0,1,\dots,n-1$, are given by (16). We consider the following more general problem where the sampled function F is defined on a closed interval J not necessarily equal to I, that is the domain of f. We suppose that $\nu, n \in \mathbb{N}$, $\nu \geq 1$ and $n \geq \nu + 2$ and that $F: J \to \mathbb{R}$, with $J = [a,b] \subset \mathbb{R}$ and $a < b$, is a sufficiently regular function on the closed interval J of $\mathbb{R}$. Let $L = b - a$, $h = 1/n$ and $m = n - \nu + 1$, then for $i = 1,2,\dots,\nu$, we consider the following points in J:

$$x^{(i)}_k = a + hL\left(k + \frac{i}{2}\right), \quad k = 0,1,\dots,n-i, \tag{22}$$

$$\xi^{(i)}_j = a + hL\left(j + \frac{i-1}{2}\right), \quad j = 0,1,\dots,n-i+1. \tag{23}$$

We note that $x^{(1)}_k = a + x_k L$ for $k=0,1,\dots,n-1$ and $\xi^{(1)}_j = a + \xi_j L$ for $j=0,1,\dots,n$.

Suppose that we know the values of F at the $n+1$ uniformly distributed points $\xi_j^{(1)}$, $j = 0,1,\dots,n$, of J, the corresponding function data are $\underline{f} = (f_0, f_1, \dots, f_n) \in \mathbb{R}^{n+1,}$ where

$$f_j = F(a + \xi_j L), \quad j = 0,1,\dots,n. \tag{24}$$

We note that the vector of samples $\underline{f}$ may be considered as obtained from the function $f(t) = F((b-a)t + a))$ defined for $t \in I$, moreover $f'(t) = (b-a)F'((b-a)t+a))$. The proposed Algorithm 1, for $k = 0,1,\dots,m-1$, computes the approximation $D_k^{(\nu)}$ of $F^{(\nu)}\left(x_k^{(\nu)}\right)$.

Algorithm 1 (ν-order derivative) **NDED**$\left(a,b,n,\nu,\underline{f};\underline{D}^{(\nu)}\right)$
Input: $a,b \in \mathbb{R}$; $n,\nu \in \mathbb{N}$; $\underline{f} = (f_0, f_1, \dots, f_n) \in \mathbb{R}^{n+1}$.
Output: $\underline{D}^{(\nu)} = (D_0, D_1, \dots, D_{m-1}) \in \mathbb{R}^m$, $m = n - \nu + 1$.

```
for l = 0,...,n−1 do
    Compute the quantity f̃^p_{ν,l} by using formula (19)
end for
for k = 0,...,n−1 do
    Compute f̃^p_k by using formula (18)
    Compute D_k^(1) = f̃^p_k / (b−a)
end for
for l = 2,3,...,ν do
    m = n − l + 1;
    Compute D^(l) ∈ R^m by NDED(ξ_0^(l), ξ_m^(l), m, 1, D^(l−1); D^(l))
end for
return D^(ν)
```

We note that the FFT algorithm is used for computing formulae (19) and (18).

3.1 MATLAB implementation

Algorithm 1 has been implemented in MATLAB, here we illustrate the corresponding function.

- **Syntax.** $[d, \mathit{ifail}] = \mathit{NumDerEquispacedData}(a,b,nu,f)$
- **Purpose.** Compute the derivatives of a function F starting from its values at uniformly distributed points.
- **Description.** $[d, \mathit{ifail}] = \mathit{NumDerEquispacedData}(a,b,nu,f)$, given a vector f containing the $n+1$ values of function F at

  $$\xi_k^{(1)} = a + k\frac{b-a}{n}, \quad k = 0,1,\dots,n,$$

  computes $d = (d_1, d_2, \dots, d_m)$ the derivatives of order $\nu = nu$ of F at

  $$x_k^{(nu)} = a + \left(k + \frac{nu}{2}\right)\frac{b-a}{n}, \quad k = 0,1,\dots,m-1, \quad m = n - nu + 1,$$

  with Algorithm 1.

- **Parameters.**
  - **input** a, b - double scalar. The closed interval $[a, b]$ is the domain of F. *Constraints:* $a < b$.
  - **input** *nu* - integer scalar. The value of *nu* is the order of the searched derivatives. *Constraints:* $nu \geq 1$.
  - **input** f - double vector with $n+1$ components. $f(j+1)$ must contain the quantity $F\left(a + j(b-a)/n\right)$, $j = 0, 1, \ldots, n$. *Constraints:* $n \geq nu + 2$.
  - **output** d - double vector with $m = n - nu + 1$ components. $d(j+1)$ contains the approximation of

$$F^{(nu)}\left(a + \left(j + \frac{nu}{2}\right)\frac{b-a}{n}\right), \quad j = 0, 1, \ldots, m-1.$$

  - **output** *ifail* – integer scalar, $ifail = 0$ unless the function detects an error (see Error Indicators and Warnings).

- **Error Indicators and Warnings.** Here is the list of errors or warnings detected by the function:
  - $ifail = 1$ - on entry $a \geq b$ or $nu \leq 0$.
  - $ifail = 2$ - the method cannot be applied because $n < nu + 2$.

4 Numerical results

The performance of the proposed algorithm is tested against the following three functions:

- $F_1(x) = \frac{1}{1+x^2}, \quad x \in [0, 1]$,
- $F_2(x) = \cos\left((1+x)^2\right), \quad x \in [0, 1]$,
- $F_3(x) = e^x, \quad x \in [-0.1, 0.5]$.

We note that the first two functions are the same test functions chosen in [10] and are used for the comparison of the two algorithm versions. Let

- $f_k^{(\nu)}$ be the ν-derivative of F at $x_k^{(\nu)}$, $k = 0, 1, \ldots, n - \nu$,
- $\hat{f}_k^{(\nu)}$ be a computed approximation of $f_k^{(\nu)}$, $k = 0, 1, \ldots, n - \nu$,

We consider the following performance indices:

$$e_f = \left| f_0^{(1)} - \hat{f}_0^{(1)} \right|, \quad e_l = \left| f_{n-1}^{(1)} - \hat{f}_{n-1}^{(1)} \right|, \tag{25}$$

$$E_\infty = \max_{0 \leq k \leq n-\nu} \left| f_k^{(\nu)} - \hat{f}_k^{(\nu)} \right|, \tag{26}$$

$$E_\infty^I = \max_{1 \leq k \leq n-\nu-1} \left| f_k^{(\nu)} - \hat{f}_k^{(\nu)} \right|, \tag{27}$$

$$E_r = \sqrt{\frac{\sum_{k=0}^{n-\nu} \left(f_k^{(\nu)} - \hat{f}_k^{(\nu)} \right)^2}{\sum_{k=0}^{n-\nu} \left(f_k^{(\nu)} \right)^2}}. \tag{28}$$

The numerical results related to algorithm NDED have been obtained by using the MATLAB *Script* **Test** that uses **NumDerEquispacedData**.

The results of this test are reported in Tables 1-4 and Fig. 1. In particular, in Tables 1 and 2, we can see that, for both functions F_1 and F_2, the values of e_f and e_l obtained by NDED are significantly lower than those obtained with NOD. This shows that new formulae (19)-(21) actually give an improved approximation at the extremes of the computation interval with respect to the formula implemented in NOD, without changing the performance at the internal points, indeed, the errors E_∞^I computed with NDED are the same of those computed with NOD.

Table 1: The comparison of the absolute errors e_f and e_l for the first derivative of function F_1 obtained with NOD and NDED, where $x(z)$ denotes the real number $x \cdot 10^z$.

h	*NOD*			*NDED*		
	e_f	e_l	E_∞^I	e_f	e_l	E_∞^I
$4.00(-2)$	$6.18(-5)$	$9.92(-6)$	$1.20(-6)$	$1.90(-6)$	$1.27(-7)$	$1.20(-6)$
$2.00(-2)$	$7.93(-6)$	$1.12(-6)$	$7.53(-8)$	$7.04(-8)$	$4.50(-9)$	$7.53(-8)$
$1.00(-2)$	$9.98(-7)$	$1.32(-7)$	$4.71(-9)$	$2.29(-9)$	$1.45(-10)$	$4.71(-9)$

Table 2: The comparison of the absolute errors e_f and e_l for the first derivative of function F_2 obtained with NOD and NDED, where $x(z)$ denotes the real number $x \cdot 10^z$.

h	*NOD*			*NDED*		
	e_f	e_l	E_∞^I	e_f	e_l	E_∞^I
$4.00(-2)$	$1.33(-4)$	$7.66(-4)$	$1.07(-5)$	$7.38(-7)$	$1.20(-5)$	$1.07(-5)$
$2.00(-2)$	$1.54(-5)$	$9.92(-5)$	$6.69(-7)$	$7.32(-9)$	$5.23(-7)$	$6.69(-7)$
$1.00(-2)$	$1.84(-6)$	$1.26(-5)$	$4.18(-8)$	$1.93(-11)$	$1.87(-8)$	$4.18(-8)$

Table 3 and Table 4 report the errors obtained by NDED, in particular, they show the errors in the numerical derivatives of order $\nu \geq 1$ with $h = 1/100$, but similar behaviors are obtained for a different choice of h. From these tables, we can see that the precision decreases as the order of the derivative increases, which suggests that the algorithm needs a more in-depth study to limit, as far as possible, this natural behavior.

Table 3: The errors obtained by computing with NDED the derivatives of order $\nu = 1, 2, 3$ of functions F_1 and F_2 with step $h = 1/100$, where $x(z)$ denotes the real number $x \cdot 10^z$.

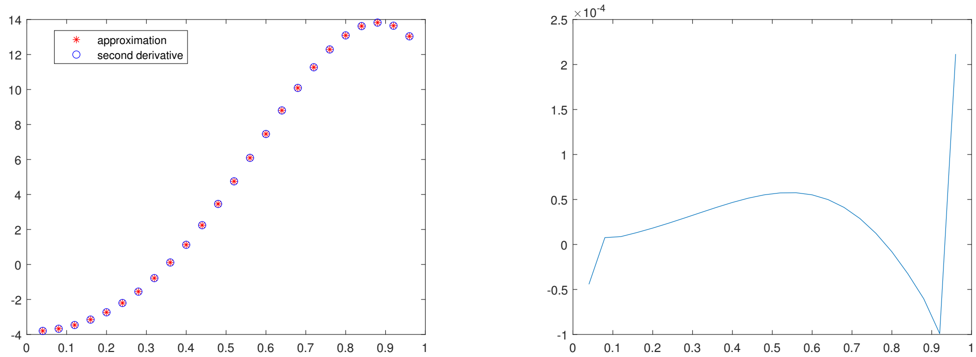
ν	F_1		F_2	
	E_∞	E_r	E_∞	E_r
1	$4.71(-9)$	$4.67(-9)$	$4.18(-8)$	$1.20(-8)$
2	$1.57(-7)$	$3.16(-8)$	$6.56(-7)$	$2.53(-8)$
3	$2.00(-5)$	$7.03(-7)$	$7.81(-5)$	$4.56(-7)$

Table 4: The errors obtained by computing with NDED the derivative of order ν of function F_3 with step $h = 1/100$, where $x(z)$ denotes the real number $x \cdot 10^z$.

ν	1	2	3	4	5
E_∞	$8.71(-12)$	$1.77(-9)$	$2.69(-7)$	$4.19(-5)$	$6.80(-3)$
E_r	$5.58(-12)$	$1.56(-10)$	$2.43(-8)$	$4.16(-6)$	$9.05(-4)$

Finally, in Figure 1 we have the graph of $F_2^{(2)}\left(x_k^{(2)}\right), k = 0, 1, \ldots, n-2$, its approximation $D_k^{(2)}$, $k = 0, 1, \ldots, n-2$, and the corresponding error $E_k = D_k^{(2)} - F_2^{(2)}\left(x_k^{(2)}\right)$, $k = 0, 1, \ldots, n-2$, when the approximation is computed with NDED and $h = 1/25$. Figure 1 gives graphical evidence of the accuracy of the derivative approximation obtained by NDED, even with few data points.

Figure 1: On the left, the graph of the second derivative of F_2 and its approximation computed with NDED and $h = 1/25$. On the right, the corresponding error $E_k = D_k^{(2)} - F_2^{(2)}\left(x_k^{(2)}\right)$, $k = 0, 1, \ldots, n-2$.



5 Conclusion

In the present paper, we described the procedure NDED for the numerical computation of the derivative of order ν, starting from function data obtained at $n+1$ uniformly distributed points on an interval $[a,b]$. This procedure is based on a recursive application of a numerical method to compute the first order derivative with an error $O(h^4)$. The procedure NDED has been implemented in MATLAB and the current code version is available on github. The current code version (v. 1.0) of the NDED procedure gives satisfactory results for equispaced univariate data. The next versions will be able to consider scattered function data and multivariate function data.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. S. Anderssen and P. Bloomfield. *Numerical differentiation procedures for non-exact data*. Numer. Math., vol. 22, (1973/74), 157–182. ISSN 0029-599X,0945-3245. URL http://dx.doi.org/10.1007/BF01436965

[2] R. S. Anderssen and M. Hegland. *For numerical differentiation, dimensionality can be a blessing!* Math. Comp., vol. 68, 227, (1999), 1121–1141. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.1090/S0025-5718-99-01033-9

[3] F. S. V. Bazán and L. Bedin. *Filtered spectral differentiation method for numerical differentiation of periodic functions with application to heat flux estimation*. Comput. Appl. Math., vol. 38, 4, (2019), Paper No. 165, 23. ISSN 2238-3603,1807-0302. URL http://dx.doi.org/10.1007/s40314-019-0968-4

[4] B. Chen, Z. Zhao, Z. Li, and Z. Meng. *Numerical differentiation by a Fourier extension method with super-order regularization*. Appl. Math. Comput., vol. 334, (2018), 1–10. ISSN 0096-3003,1873-5649. URL http://dx.doi.org/10.1016/j.amc.2018.04.005

[5] J. Cheng, X. Z. Jia, and Y. B. Wang. *Numerical differentiation and its applications*. Inverse Probl. Sci. Eng., vol. 15, 4, (2007), 339–357. ISSN 1741-5977,1741-5985. URL http://dx.doi.org/10.1080/17415970600839093

[6] O. Davydov and R. Schaback. *Error bounds for kernel-based numerical differentiation*. Numer. Math., vol. 132, 2, (2016), 243–269. ISSN 0029-599X,0945-3245. URL http://dx.doi.org/10.1007/s00211-015-0722-9

[7] O. Davydov and R. Schaback. *Minimal numerical differentiation formulas*. Numer. Math., vol. 140, 3, (2018), 555–592. ISSN 0029-599X,0945-3245. URL http://dx.doi.org/10.1007/s00211-018-0973-3

[8] F. Dell'Accio, F. Di Tommaso, N. Siar, and M. Vianello. *Numerical differentiation on scattered data through multivariate polynomial interpolation*. BIT, vol. 62, 3, (2022), 773–801. ISSN 0006-3835,1572-9125. URL http://dx.doi.org/10.1007/s10543-021-00897-6

[9] N. Egidi, J. Giacomini, and P. Maponi. *A Fredholm integral operator for the differentiation problem*. Comput. Appl. Math., vol. 41, 5, (2022), Paper No. 220, 21. ISSN 2238-3603,1807-0302. URL http://dx.doi.org/10.1007/s40314-022-01923-1

[10] N. Egidi, J. Giacomini, P. Maponi, and M. Youssef. *An FFT method for the numerical differentiation*. Appl. Math. Comput., vol. 445, (2023), Paper No. 127856, 13. ISSN 0096-3003,1873-5649. URL http://dx.doi.org/10.1016/j.amc.2023.127856

[11] N. Egidi and P. Maponi. *The singular value expansion of the Volterra integral equation associated to a numerical differentiation problem*. J. Math. Anal. Appl., vol. 460, 2, (2018), 656–681. ISSN 0022-247X,1096-0813. URL http://dx.doi.org/10.1016/j.jmaa.2017.12.007

[12] N. Egidi and P. Maponi. *An SVE Approach for the Numerical Solution of Ordinary Differential Equations*. Y. D. Sergeyev and D. E. Kvasov (eds.), *Numerical Computations: Theory and Algorithms*, (Springer International Publishing, Cham2020). ISBN 978-3-030-39081-5, 70–85. URL http://dx.doi.org/10.1007/978-3-030-39081-5_8

[13] N. Egidi and P. Maponi. *A spectral method for the solution of boundary value problems*. Appl. Math. Comput., vol. 409, (2021), Paper No. 125812, 12. ISSN 0096-3003,1873-5649. URL http://dx.doi.org/10.1016/j.amc.2020.125812

[14] X. Huang, C. Wu, and J. Zhou. *Numerical differentiation by integration*. Math. Comp., vol. 83, 286, (2014), 789–807. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.1090/S0025-5718-2013-02722-6

[15] I. Knowles and R. Wallace. *A variational method for numerical differentiation*. Numer. Math., vol. 70, 1, (1995), 91–110. ISSN 0029-599X,0945-3245. URL http://dx.doi.org/10.1007/s002110050111

[16] S. Lu and S. V. Pereverzev. *Numerical differentiation from a viewpoint of regularization theory*. Math. Comp., vol. 75, 256, (2006), 1853–1870. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.1090/S0025-5718-06-01857-6

[17] Y. Luo. *Galerkin method with trigonometric basis on stable numerical differentiation*. Appl. Math. Comput., vol. 370, (2020), 124912, 26. ISSN 0096-3003,1873-5649. URL http://dx.doi.org/10.1016/j.amc.2019.124912

[18] A. G. Ramm and A. B. Smirnova. *On stable numerical differentiation*. Math. Comp., vol. 70, 235, (2001), 1131–1153. ISSN 0025-5718,1088-6842. URL http://dx.doi.org/10.1090/S0025-5718-01-01307-2

[19] T. J. Rivlin. *Optimally stable Lagrangian numerical differentiation*. SIAM J. Numer. Anal., vol. 12, 5, (1975), 712–725. ISSN 0036-1429. URL http://dx.doi.org/10.1137/0712053

[20] T. Tsuda. *Numerical differentiation of functions of very many variables*. Numer. Math., vol. 18, (1971/72), 327–335. ISSN 0029-599X,0945-3245. URL http://dx.doi.org/10.1007/BF01404683

[21] X. Q. Wan, Y. B. Wang, and M. Yamamoto. *Detection of irregular points by regularization in numerical differentiation and application to edge detection*. Inverse Problems, vol. 22, 3, (2006), 1089–1103. ISSN 0266-5611,1361-6420. URL http://dx.doi.org/10.1088/0266-5611/22/3/022

[22] Y. B. Wang, X. Z. Jia, and J. Cheng. *A numerical differentiation method and its application to reconstruction of discontinuity*. Inverse Problems, vol. 18, 6, (2002), 1461–1476. ISSN 0266-5611,1361-6420. URL http://dx.doi.org/10.1088/0266-5611/18/6/301

[23] B. Yin and Y. Ye. *Recovering the local volatility in Black-Scholes model by numerical differentiation*. Appl. Anal., vol. 85, 6-7, (2006), 681–692. ISSN 0003-6811,1563-504X. URL http://dx.doi.org/10.1080/00036810500475025

[24] Z. Zhao. *A Hermite extension method for numerical differentiation*. Appl. Numer. Math., vol. 159, (2021), 46–60. ISSN 0168-9274,1873-5460. URL http://dx.doi.org/10.1016/j.apnum.2020.08.016

[25] Z. Zhao, Z. Meng, and G. He. *A new approach to numerical differentiation*. J. Comput. Appl. Math., vol. 232, 2, (2009), 227–239. ISSN 0377-0427,1879-1778. URL http://dx.doi.org/10.1016/j.cam.2009.06.001

[26] Z. Zhao and L. You. *A numerical differentiation method based on legendre expansion with super order Tikhonov regularization*. Appl. Math. Comput., vol. 393, (2021), Paper No. 125811, 11. ISSN 0096-3003,1873-5649. URL http://dx.doi.org/10.1016/j.amc.2020.125811