computation

Article

Cutting-Edge Computational Approaches for Approximating Nonlocal Variable-Order Operators

Nayereh Tanha [1], Behrouz Parsa Moghaddam [1,*] and Mousa Ilie [2]

[1] Department of Mathematics, Lahijan Branch, Islamic Azad University, Lahijan P.O. Box 1616, Iran; na.tanha@iau.ac.ir

[2] Department of Mathematics, Rasht Branch, Islamic Azad University, Rasht P.O. Box 3516-41335, Iran; ilie@iaurasht.ac.ir

* Correspondence: bparsa@fe.up.pt

Abstract: This study presents an algorithmically efficient approach to address the complexities associated with nonlocal variable-order operators characterized by diverse definitions. The proposed method employs integro spline quasi interpolation to approximate these operators, aiming for enhanced accuracy and computational efficiency. We conduct a thorough comparison of the outcomes obtained through this approach with other established techniques, including finite difference, IQS, and B-spline methods, documented in the applied mathematics literature for handling nonlocal variable-order derivatives and integrals. The numerical results, showcased in this paper, serve as a compelling validation of the notable advantages offered by our innovative approach. Furthermore, this study delves into the impact of selecting different variable-order values, contributing to a deeper understanding of the algorithm's behavior across a spectrum of scenarios. In summary, this research seeks to provide a practical and effective solution to the challenges associated with nonlocal variable-order operators, contributing to the applied mathematics literature.

Keywords: fractional calculus; integro spline; quasi interpolation; variable-order fractional derivatives and integrals; numerical computation using splines

MSC: 26A33; 41A05; 33F05; 65M70; 65D05; 65D07

Citation: Tanha, N.; Parsa Moghaddam, B.; Ilie, M. Cutting-Edge Computational Approaches for Approximating Nonlocal Variable-Order Operators. *Computation* **2024**, *12*, 14. https://doi.org/10.3390/computation12010014

Academic Editor: Simeone Marino

Received: 7 December 2023
Revised: 31 December 2023
Accepted: 4 January 2024
Published: 12 January 2024

1. Introduction

In recent years, the field of fractional calculus has garnered substantial interest among researchers, owing to its extensive applications in various scientific and engineering domains. This mathematical discipline proves invaluable in refining models employed in fluid mechanics, viscoelasticity, chemistry, physics, finance, and other scientific disciplines [1–4]. This surge in research activity underscores the dynamic nature of fractional calculus, reflecting its continual evolution and the ongoing quest to enhance mathematical models crucial for addressing real-world challenges.

Spline functions and their associated properties have garnered significant attention from researchers [5,6]. The research by Zahra and colleagues has made notable contributions to the field. In their work, Zahra et al. proposed a robust uniform B-spline collocation method for solving the generalized PHI-four equation [5]. Additionally, they developed a cubic B-spline collocation algorithm for the numerical solution of Newell Whitehead Segel-type equations [6]. The initial breakthrough in this field was made by Behforooz, who introduced a novel approach for constructing cubic integro splines. This approach departed from the traditional practice of using function values at nodes and instead relied on integral values denoted as $u(t)$, as documented in [7]. Subsequently, this method evolved into cubic Hermit integro spline interpolation, incorporating three additional boundary conditions. A thorough examination of its characteristics can be found in [8]. Behforooz further extended the range of integro splines by introducing fifth-order variants in [9].

However, these functions had complex structures, and the consideration of the derivative of $u(t)$ was not within the scope of the research. The quest for more localized cubic integro splines, specifically for computing derivatives $u^b(t),\ b = 0, 1, 2$, was addressed in [10]. Additionally, integro splines of fourth and sixth orders were put forth in [11,12], respectively. It is important to note that all the existing methods in integro spline interpolation are based on values within consecutive uniform subintervals. To enhance these algorithms, the concept of quadratic uniform B-spline was explored in [13,14]. Furthermore, the study of integro splines quasi-interpolants and their super convergence properties is documented in [15–17].

Nonlocal variable-order operators (NVOs) find their application in modeling systems that exhibit memory-related characteristics. These equations encompass both fixed and variable-order operators, as discussed in [18]. Differential equations rooted in NVOs, often referred to as NVODEs, play a pivotal role in exploring a wide spectrum of issues and fields, including economics [19], electric vehicles [20], and physics [21].

In-depth investigations into the existence and uniqueness of NVODEs have been undertaken, as documented in [22–24]. Furthermore, various efficient and practical numerical techniques have been proposed to solve NVODEs, encompassing methods such as Adams [25], cubic spline [26], finite difference [27,28], Legendre polynomial [29], Bernstein polynomial [30], Chebyshev polynomial [31].

We consider an integer $q \in \mathbb{N}$, and the unknown function $u(t)$ is assumed to be continuously differentiable up to $(q-1)$ times. We describe it as a jointly continuous function represented by $P : \Phi \times \mathbb{R} \to \mathbb{R}$, where $\Phi := [0, T]$. Consequently, we employ the variable-order (VO) nonlocal operators as introduced in [32] and further developed in [18,33]. These operators are defined as follows:

Definition 1 ([18]). *The VO nonlocal derivative is stated as*

$$^{v}\mathscr{D}_{0,t}^{\varrho(t)}u(t) = \int_0^t \frac{(t-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))} \cdot u^{(q)}(\varsigma)d\varsigma, \ \ 0 \le q-1 < \varrho(t) \le q \in \mathbb{N}, \tag{1}$$

where $t, \varsigma \in \mathbb{R}^+$ and $\Gamma(\cdot)$ denotes the Gamma function.

Definition 2 ([18]). *The VO nonlocal integral is stated as*

$$^{v}\mathscr{I}_{0,t}^{\varrho(t)}u(t) = \int_0^t \frac{(t-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))} \cdot u(\varsigma)d\varsigma, \ \ Re(\varrho(t)) > 0, \tag{2}$$

where $t, \varsigma \in \mathbb{R}^+$ and $\Gamma(\cdot)$ denotes the Gamma function.

The subsequent sections of this manuscript follow the following structure. In Section 2, we introduce an effective strategy that employs integro spline quasi interpolation to discretize nonlocal derivatives and integrals, as defined in Definitions 1 and 2, respectively. This section provides a detailed exploration of the theoretical aspects and methodologies associated with this approach. To evaluate the precision and dependability of the proposed technique, a series of examples undergo a thorough examination in Section 3. Lastly, Section 4 presents a comprehensive overview of the primary discoveries and conclusions drawn from our investigation, along with valuable insights and recommendations for prospective research directions.

2. Theoretical Results

Let us consider a discrete time interval Φ where $t_m = m\Delta$ for $m = 0, 1, \ldots, M$. Here, Δ represents the uniform step size, and h denotes the size of each subinterval. The values of m and M are positive integers.

We define $\tau(t)$ as a quadratic polynomial on each subinterval $[t_j, t_{j+1}]$, where $0 = t_0 < t_1 < \ldots < t_m = T$. Specifically, $\tau(t)$ is referred to as an integro quadratic spline quasiinterpolant (IntQuaSpline-QI) function, constructed with respect to the given mesh points $t = [t_0, t_1, \dots, t_m]$. Assuming that J_l represents the integral of $u(t)$ over each subinterval $[t_l, t_{l+1}]$, we can express this relationship as follows:

$$J_l = \int_{t_l}^{t_{l+1}} \tau(t)\mathrm{d}t = \int_{t_l}^{t_{l+1}} u(t)\mathrm{d}t, \;\; l = 0, 1, \dots, m-1, \tag{3}$$

then,

$$\tau(t) = \frac{1}{12\Delta^3}\left((t - t_{l+1})^2\lambda_{l-2} - ((t - t_{l-1})(t - t_{l+1}) + (t - t_l)(t - t_{l+2}))\lambda_{l-1} + (t - t_l)^2\lambda_l\right), \tag{4}$$

where

$$\lambda_l = \begin{cases} 11J_0 - 7J_1 + 2J_2 & l = -2 \\ 5J_0 + 2J_1 - J_2 & l = -1 \\ 8J_{l+1} - J_l - J_{l+2} & l = 0, \dots, m, \;\; m = 0, 1, \dots, M-3 \\ 5J_{m-1} + 2J_{m-2} - J_{m-3} & l = M-2 \\ 11J_{m-1} - 7J_{m-2} + 2J_{m-3} & l = M-1 \end{cases}. \tag{5}$$

consequently, λ_l is solely determined by the integral values over the interval $[t_l, t_{l+3}]$.

Corollary 1 ([16])**.** *Assume $u(t) \in C^3(\Phi)$; hence,*

$$\| \partial_t^{(n)}\tau(t) - \partial_t^{(n)}u(t) \|_\infty = \mathcal{O}(\triangle^{3-n}), \qquad n = 0, 1. \tag{6}$$

Corollary 2 ([15])**.** *Assume $\Delta = \frac{T}{M}$, Φ is divided to m uniform sub-intervals and $u(t) \in C^\infty(\Phi)$, we have*

$$\tau(t_l) = u(t_l) - \frac{h^4}{30}\partial_t^{(4)}u(t_l) + higher \;\; term, \;\; l = 1, 2, \dots, m-2, \tag{7}$$

and

$$\max_{2 \le l \le m-2} |\tau(t_l) - u(t_l)| = \mathcal{O}(\triangle^4), \tag{8}$$

where the term $\frac{h^4}{30}\partial_t^{(4)}u(t_l)$ is the fourth-order accurate approximation of $u(t_l)$, and the "higher term" accounts for the error beyond this fourth-order approximation. The "higher term" includes all contributions from terms of fifth order and higher in the error expansion.

For the time points t_m, where $m = 1, \dots, M$, we have the following relationships:

$$\begin{aligned} {}^v\mathscr{D}_{0,t_m}^{\varrho(t)}u(t) &= \int_0^{t_m} \frac{(t_m - \varsigma)^{q-\varrho(t)-1}}{\Gamma(q - \varrho(t))} \cdot u^{(q)}(\varsigma)d\varsigma \\ &= \sum_{l=0}^{m-1} \int_{t_l}^{t_{l+1}} \frac{(t_m - \varsigma)^{q-\varrho(t)-1}}{\Gamma(q - \varrho(t))} \cdot u^{(q)}(\varsigma)d\varsigma, \end{aligned} \tag{9}$$

and

$$\begin{aligned} {}^v\mathscr{I}_{0,t_m}^{\varrho(t)}u(t) &= \int_0^{t_m} \frac{(t_m - \varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))} \cdot u(\varsigma)d\varsigma \\ &= \sum_{l=0}^{m-1} \int_{t_l}^{t_{l+1}} \frac{(t_m - \varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))} \cdot u(\varsigma)d\varsigma. \end{aligned} \tag{10}$$

For each $l = 0, 1, \dots, m-1$, we utilize an IntQuaSpline-QI function $\tau(t)$ with mesh points at t_l to approximate the function $u(t)$, resulting in the following expressions:

$$\begin{aligned} u(t) \approx \tau_m(t) &= \frac{1}{12\Delta^3}\sum_{l=0}^{m-1}\left((t - t_{l+1})^2\lambda_{l-2}\right. \\ &\quad \left. - ((t - t_{l-1})(t - t_{l+1}) + (t - t_l)(t - t_{l+2}))\lambda_{l-1} + (t - t_l)^2\lambda_l\right) \end{aligned} \tag{11}$$

and

$$u^{(q)}(t) \approx \tilde{\tau}_m(t) = \frac{1}{12\Delta^3}\sum_{l=0}^{m-1}\left((t-t_{l+1})^2\lambda_{l-2}^{(q)} - \left((t-t_{l-1})(t-t_{l+1})+(t-t_l)(t-t_{l+2})\right)\lambda_{l-1}^{(q)} + (t-t_l)^2\lambda_l^{(q)}\right). \tag{12}$$

By substituting Equation (12) into Equation (9), we obtain

$$\begin{aligned} {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}u(t) &\approx \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}}\frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))}\tilde{\tau}_l(\varsigma)d\varsigma \\ &= \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}}\frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{12\Delta^3\Gamma(q-\varrho(t))}\Big((\varsigma-t_{l+1})^2\lambda_{l-2}^{(q)} \\ &\quad -\Big((\varsigma-t_{l-1})(\varsigma-t_{l+1})+(\varsigma-t_l)(\varsigma-t_{l+2})\Big)\lambda_{l-1}^{(q)}+(\varsigma-t_l)^2\lambda_l^{(q)}\Big)d\varsigma. \end{aligned} \tag{13}$$

Moreover, by substituting Equation (11) into Equation (10), we obtain

$$\begin{aligned} {}^{v}\mathscr{I}_{0,t_m}^{\varrho(t)}u(t) &\approx \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}}\frac{(t_m-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))}\tau_l(\varsigma)d\varsigma \\ &= \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}}\frac{(t_m-\varsigma)^{\varrho(t)-1}}{12\Delta^3\Gamma(\varrho(t))}\Big((\varsigma-t_{l+1})^2\lambda_{l-2} \\ &\quad -\Big((\varsigma-t_{l-1})(\varsigma-t_{l+1})+(\varsigma-t_l)(\varsigma-t_{l+2})\Big)\lambda_{l-1}+(\varsigma-t_l)^2\lambda_l\Big)d\varsigma. \end{aligned} \tag{14}$$

Consequently, we derive the following three propositions:

Proposition 1. *Assume that $u(t) \in C^{q+4}(\Phi)$ be a function, $q-1 < \varrho(t) \leq q$. The discretization of the nonlocal derivative can be stated from the IntQuaSpline-QI approximation as shown below*

$${}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}u(t) = \sum_{l=0}^{m-1}\frac{\Delta^{q-\varrho_m-1}}{6\Gamma(q-\varrho_m+3)}\left(\alpha_{l,l-2}\lambda_{l-2}^{(q)}+\alpha_{l,l-1}\lambda_{l-1}^{(q)}+\alpha_{l,l}\lambda_l^{(q)}\right), \tag{15}$$

where for $l = 0, 1, \ldots, m$, λ_l is defined in (5), and

$$\alpha_{l,k} = \begin{cases} -(m-l)^{q-\varrho_m+2} \\ \quad +\Big(\frac{(q-\varrho_m)^2}{2}+(2l-2m+1)\frac{q-\varrho_m}{2}+(l-m)^2\Big)(m-l+1)^{q-\varrho_m}, & k=l-2, \\ \Big(\frac{(q-\varrho_m)^2}{2}+(2m-2l+5)\frac{q-\varrho_m}{2} \\ \quad -2(l-m)^2+(1+2l-2m)\Big)(m-l+1)^{q-\varrho_m}+\Big(-\frac{(q-\varrho_m)^2}{2} \\ \quad +(2m-2l-3)\frac{q-\varrho_m}{2}+2(l-m)^2+(2m-2l-1)\Big)(m-l)^{q-\varrho_m}, & k=l-1, \\ \big(-\frac{(q-\varrho_m)^2}{2}+(2l-2m-3)\frac{q-\varrho_m}{2}-(l-m)^2 \\ \quad +(2l-2m-1)\big)(m-l)^{q-\varrho_m}+(m-l+1)^{q-\varrho_m+2}, & k=l. \end{cases} \tag{16}$$

Proposition 2. *Assume that $u(t) \in C^4(\Phi)$ be a function, $Re(\varrho(t)) > 0$. The discretization of the nonlocal integral can be stated from the IntQuaSpline-QI approximation as shown below*

$${}^{v}\mathscr{I}_{0,t_m}^{\varrho(t)}u(t) = \sum_{l=0}^{m-1}\frac{\Delta^{\varrho_m-1}}{6\Gamma(\varrho_m+3)}\left(\beta_{l,l-2}\lambda_{l-2}+\beta_{l,l-1}\lambda_{l-1}+\beta_{l,l}\lambda_l\right), \tag{17}$$

where for $l = 0, 1, \ldots, m$, λ_l is defined in (5), and

$$\beta_{l,k} = \begin{cases} \begin{aligned} &-(m-l)^{\varrho_m+2} \\ &\quad + \Big(\tfrac{\varrho_m^2}{2} + (2l-2m+1)\tfrac{\varrho_m}{2} + (l-m)^2\Big)(m-l+1)^{\varrho_m}, \end{aligned} & k = l-2, \\[2ex] \begin{aligned} &\Big(\tfrac{\varrho_m^2}{2} + (2m-2l+5)\tfrac{\varrho_m}{2} \\ &\quad -2(l-m)^2 + (1+2l-2m)\Big)(m-l+1)^{\varrho_m} + \Big(-\tfrac{\varrho_m^2}{2} \\ &\quad +(2m-2l-3)\tfrac{\varrho_m}{2} + 2(l-m)^2 + (2m-2l-1)\Big)(m-l)^{\varrho_m}, \end{aligned} & k = l-1, \\[2ex] \begin{aligned} &\big(-\tfrac{\varrho_m^2}{2} + (2l-2m-3)\tfrac{\varrho_m}{2} - (l-m)^2 \\ &\quad +(2l-2m-1)\big)(m-l)^{\varrho_m} + (m-l+1)^{\varrho_m+2}, \end{aligned} & k = l. \end{cases} \tag{18}$$

Proposition 3. *Let $u(t) \in C^{q+3}(\Phi)$ be a function, $q-1 < \varrho(t) \leq q$, and $\|\partial_t^{(q+3)} u(t)\|_\infty \leq \Xi$, where $\Xi > 0$. Under these assumptions, the truncated error of the presented algorithm is bounded, satisfying the following inequality:*

$$AE_m = \left\| {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)] - \left({}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)]\right)_{approx} \right\|_\infty \leq \frac{\Xi m^{q-\varrho(t_m)}}{\Gamma(q-\varrho(t_m)+1)} \Delta^{q-\varrho(t_m)+3}. \tag{19}$$

Proof. Suppose $\tilde{\tau}_\Phi(t)$ is an IntQuaSpline-QI function that approximates $u(t)$ within the subinterval $[t_l, t_{l+1}] \subseteq \Phi$, where $l = 0, 1, \ldots, m-1$. For an arbitrary value $\mu_l \in (t_l, t_{l+1})$, we can establish the following relationship:

$$\mathscr{E}_\Phi(t) = u^{(q)}(t) - \tilde{\tau}_\Phi^{(q)}(t) = \frac{\Delta^3}{12} \partial_t^{(q+3)} u(\mu_l),$$

thus,

$$\begin{aligned} &\left\| {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)] - \left({}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)]\right)_{approx} \right\|_\infty \\ &\qquad = \left\| {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)] - {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[\tilde{\tau}_\Phi(t)] \right\|_\infty \\ &\qquad = \int_0^{t_m} \left\| \frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))} \mathscr{E}_\Phi(\varsigma) \right\|_\infty d\varsigma \\ &\qquad = \sum_{l=0}^{m-1} \int_{t_l}^{t_{l+1}} \frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))} \left\| \frac{\Delta^3}{12} \partial_t^{(q+3)} u(\mu_l) \right\|_\infty d\varsigma \\ &\qquad \leq \frac{t_m^{q-\varrho(t_m)} \Xi}{\Gamma(q-\varrho(t_m)+1)} \Delta^3 = \frac{\Xi m^{\varrho(t_m)}}{\Gamma(q-\varrho(t_m)+1)} \Delta^{q-\varrho(t_m)+3}. \end{aligned}$$

□

Proposition 4. *Let $u(t) \in C^{q+4}(\Phi_1)$ be a function defined on the interval $\Phi_1 = [t_2, t_{M-2}] \subseteq \Phi$. Here, $q-1 < \varrho(t) \leq q$ and $\|\partial_t^{(q+4)} u(t)\|\infty \leq \Xi_1$, where $\Xi_1 > 0$. Under these conditions, the truncated error of the presented algorithm is bounded and can be expressed as follows:*

$$\left\| {}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)] - \left({}^{v}\mathscr{D}_{0,t_m}^{\varrho(t)}[u(t)]\right)_{approx} \right\|_\infty \leq \frac{m^{q-\varrho(t_m)} \Xi_1}{\Gamma(q-\varrho(t_m)+1)} \Delta^{q-\varrho(t_m)+4}, \tag{20}$$

where $m = 2, 3, \ldots, M-3$.

Proof. Consider $\tilde{\tau}_{\Phi_1}(t)$ as an IntQuaSpline-QI function utilized to approximate $u(t)$ within the subinterval $[t_l, t_{l+1}] \subseteq \Phi$, where $l = 2, 3, \ldots, m$. Hence, for any arbitrary value $\psi_l \in (t_l, t_{l+1})$, the following relation holds:

$$\mathscr{E}_{\Phi_1}(t) = u^{(q)}(t) - \tilde{\tau}^{(q)}_{\Phi_1}(t) = \frac{(t-t_l)^2(t-t_{l+1})^2}{30}\partial_t^{(q+4)}u(\psi_l),$$

hence,

$$\begin{aligned}
&\left\| {}^{v}\mathscr{D}^{\varrho(t)}_{0,t_m}[u(t)] - \left({}^{v}\mathscr{D}^{\varrho(t)}_{0,t_m}[u(t)]\right)_{approx}\right\|_\infty \\
&= \left\| {}^{v}\mathscr{D}^{\varrho(t)}_{0,t_m}[u(t)] - {}^{v}\mathscr{D}^{\varrho(t)}_{0,t_m}[\tilde{\tau}_{\Phi_1}(t)]\right\|_\infty \\
&= \left\| \int_0^{t_m} \frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))}\mathscr{E}_{\Phi_1}(\varsigma)\right\|_\infty d\varsigma \\
&= \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}} \frac{(t_m-\varsigma)^{q-\varrho(t)-1}}{\Gamma(q-\varrho(t))}\left\|\frac{(\varsigma-t_l)^2(\varsigma-t_{l+1})^2}{30}\partial_t^{(q+4)}u(\psi_l)\right\|_\infty d\varsigma \\
&\leq \frac{t_m^{q-\varrho(t_m)}\Xi_1}{\Gamma(q-\varrho(t_m)+1)}\Delta^4 = \frac{m^{q-\varrho(t_m)}\Xi_1}{\Gamma(q-\varrho(t_m)+1)}\Delta^{q-\varrho(t_m)+4}.
\end{aligned}$$

□

Proposition 5. *Let $u(t) \in C^3(\Phi)$ be a function, and $\|\partial_t^3 u(t)\|_\infty \leq \chi$, where $\chi > 0$. Under these assumptions, the truncated error of the presented algorithm is bounded, satisfying the following inequality:*

$$AE_m = \left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - \left({}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)]\right)_{approx}\right\|_\infty \leq \frac{\chi m^{\varrho(t_m)}}{\Gamma(\varrho(t_m)+1)}\Delta^{\varrho(t_m)+3}. \tag{21}$$

Proof. Suppose $\tau_\Phi(t)$ is an IntQuaSpline-QI function that approximates $u(t)$ within the subinterval $[t_l, t_{l+1}] \subseteq \Phi$, where $l = 0, 1, \ldots, m-1$. For an arbitrary value $\mu_l \in (t_l, t_{l+1})$, we can establish the following relationship:

$$\mathscr{E}_\Phi(t) = u(t) - \tau_\Phi(t) = \frac{\Delta^3}{12}\partial_t^3 u(\mu_l),$$

thus,

$$\begin{aligned}
&\left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - \left({}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)]\right)_{approx}\right\|_\infty \\
&= \left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[\tau_\Phi(t)]\right\|_\infty \\
&= \int_0^{t_m} \left\| \frac{(t_m-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))}\mathscr{E}_\Phi(\varsigma)\right\|_\infty d\varsigma \\
&= \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}} \frac{(t_m-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))}\left\|\frac{\Delta^3}{12}\partial_t^3 u(\mu_l)\right\|_\infty d\varsigma \\
&\leq \frac{t_m^{\varrho(t_m)}\chi}{\Gamma(\varrho(t_m)+1)}\Delta^3 = \frac{\chi m^{\varrho(t_m)}}{\Gamma(\varrho(t_m)+1)}\Delta^{\varrho(t_m)+3}.
\end{aligned}$$

□

Proposition 6. *Let $u(t) \in C^4(\Phi_1)$ be a function defined on the interval $\Phi_1 = [t_2, t_{M-2}] \subseteq \Phi$. Here, $Re(\varrho(t)) > 0$ and $\|\partial_t^4 u(t)\|\infty \leq \chi_1$, where $\chi_1 > 0$. Under these conditions, the truncated error of the presented algorithm is bounded and can be expressed as follows:*

$$\left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - \left({}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)]\right)_{approx} \right\|_\infty \leq \frac{m^{\varrho(t_m)}\chi_1}{\Gamma(\varrho(t_m)+1)}\Delta^{\varrho(t_m)+4}, \tag{22}$$

where $m = 2, 3, \ldots, M-3$.

Proof. Consider $\tau_{\Phi_1}(t)$ as an IntQuaSpline-QI function utilized to approximate $u(t)$ within the subinterval $[t_l, t_{l+1}] \subseteq \Phi$, where $l = 2, 3, \ldots, m$. Hence, for any arbitrary value $\psi_l \in (t_l, t_{l+1})$, the following relation holds:

$$\mathscr{E}_{\Phi_1}(t) = u(t) - \tau_{\Phi_1}(t) = \frac{(t-t_l)^2(t-t_{l+1})^2}{30}\partial_t^4 u(\psi_l),$$

hence,

$$\begin{aligned}
&\left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - \left({}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)]\right)_{approx} \right\|_\infty \\
&= \left\| {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[u(t)] - {}^{v}\mathscr{I}^{\varrho(t)}_{0,t_m}[\tau_{\Phi_1}(t)] \right\|_\infty \\
&= \left\| \int_0^{t_m} \frac{(t_m-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))}\mathscr{E}_{\Phi_1}(\varsigma) \right\|_\infty d\varsigma \\
&= \sum_{l=0}^{m-1}\int_{t_l}^{t_{l+1}} \frac{(t_m-\varsigma)^{\varrho(t)-1}}{\Gamma(\varrho(t))}\left\| \frac{(\varsigma-t_l)^2(\varsigma-t_{l+1})^2}{30}\partial_t^4 u(\psi_l) \right\|_\infty d\varsigma \\
&\leq \frac{t_m^{\varrho(t_m)}\Xi_1}{\Gamma(\varrho(t_m)+1)}\Delta^4 = \frac{m^{q-\varrho(t_m)}\chi_1}{\Gamma(\varrho(t_m)+1)}\Delta^{\varrho(t_m)+4}.
\end{aligned}$$

□

3. Numerical Demonstrations

Now, the accuracy and computational efficiency of the developed approach are studied. For this purpose, the *mean absolute error* (MAE) ($\mathscr{E}_M$) and the *convergence order* (ECO) are considered

$$\mathscr{E}_M = \sum_{m=1}^{M}\frac{AE_m}{M}, \tag{23}$$

and

$$ECO = \log_\Delta(\mathscr{E}_M). \tag{24}$$

The MAE is used to measure the average absolute error difference between the approximated and exact solutions, and the ECO is used to measure the order of convergence of the method. These metrics are computed using the error formula in (19) and (21), where AE_M denotes the absolute error and M represents the number of interior mesh points.

All the computational results are implemented with Matlab v2019 running in an Intel (R) Core (TM) i7-8850 H CPU @ 2.60 GHz machine. Furthermore, a comparison with the other algorithms is conducted.

Example 1 ([34]). *Let*

$${}^{v}\mathscr{D}^{\varrho(t)}_{0,t}(t\cdot\sin(t)) =$$

$$\begin{cases} \dfrac{t^{\frac{5}{2}-\varrho(t)}{}_2F_3\left([1,\frac{3}{2}],[\frac{1}{2},2-\frac{\varrho(t)}{2},\frac{3}{2}-\frac{\varrho(t)}{2};-\frac{t^2}{4}\right)+(\varrho(t)-2)\left(t\varrho(t)S_{\frac{1}{2}-\varrho(t),\frac{3}{2}}(t)-S_{\frac{3}{2}-\varrho(t),\frac{1}{2}}(t)\right)}{\sqrt{t}\Gamma(3-\varrho(t))}, & 0<\varrho(t)\leq 1 \\ \dfrac{(\varrho(t)-2)(\varrho(t)-3)(\varrho(t)-4)\left((t^2-\varrho(t))S_{\frac{3}{2}-\varrho(t),\frac{1}{2}}(t)+t\varrho^2(t)S_{\frac{1}{2}-\varrho(t),\frac{3}{2}}(t)-t^{\frac{5}{2}-\varrho(t)}\right)}{\sqrt{t}\Gamma(5-\varrho(t))}, & 1<\varrho(t)\leq 2 \end{cases}, \tag{25}$$

where ${}_sF_v(b_1,\ldots,b_s;a_1,\ldots,a_v;t)$ *and* $S_{\mu,\nu}(t)$ *are defined as the hypergeometric and Lommel functions [35], respectively.*

Additionally, for $Re(\varrho(t)) > 0$, we have

$$ {}^{v}\mathscr{I}_{0,t}^{\varrho(t)}(t\cdot\sin(t)) = -\frac{(\varrho(t)+2)\big((t^2+\varrho(t))S_{\frac{3}{2}+\varrho(t),\frac{1}{2}}(t) + t\varrho^2(t)S_{\frac{1}{2}+\varrho(t),\frac{3}{2}}(t) - t^{\frac{5}{2}+\varrho(t)}\big)}{\sqrt{t}\,\Gamma(3+\varrho(t))}, \qquad (26)$$

The efficiency of the developed algorithm is described by ${}^{v}\mathscr{D}_{0,t}^{\varrho(t)}(t\cdot\sin(t))$ and ${}^{v}\mathscr{I}_{0,t}^{\varrho(t)}(t\cdot\sin(t))$ in Example 1. Tables 1 and 2 display the values of $\mathscr{E}_M$, ECO and computational times (based on sec.) of (25) and (26) with $\Delta = \{\frac{1}{16},\frac{1}{32},\frac{1}{64},\frac{1}{128}\}$ in $t\in[0,2]$ for

$$\begin{cases} \varrho_1(t) = 0.4 - 0.01\sqrt[5]{t} \\ \varrho_2(t) = 0.8 + 0.03\left|\cos\left(\frac{t}{3\pi}\right)\right| \\ \varrho_3(t) = 1.4 - 0.01\sqrt[5]{t} \\ \varrho_4(t) = 1.8 + 0.03\left|\cos\left(\frac{t}{3\pi}\right)\right| \end{cases}.$$

According to Tables 1 and 2, all results are improved compared to the IQS [36] and B-spline [25] -algorithms, respectively. Table 1 provides a comprehensive comparison between the developed algorithm and the IQS algorithm across varying values of $\varrho(t)$ and Δ within the interval $t\in[0,2]$. In assessing key performance indicators, the error ($\mathscr{E}_M$) of the developed algorithm consistently decreases significantly with finer discretization (Δ), showcasing improved accuracy. The IQS algorithm also demonstrates a reduction in error with decreasing Δ, though the magnitude of improvement is generally smaller. Regarding convergence order (ECO), both algorithms exhibit stability across different Δ values, with the developed algorithm consistently maintaining competitive or superior convergence order compared to the IQS algorithm. Notably, the developed algorithm outperforms in terms of accuracy and convergence order, albeit with a slightly higher computational time, suggesting enhanced performance at the cost of increased computational complexity. This emphasizes the potential of the developed algorithm for applications prioritizing accuracy where ample computational resources are available.

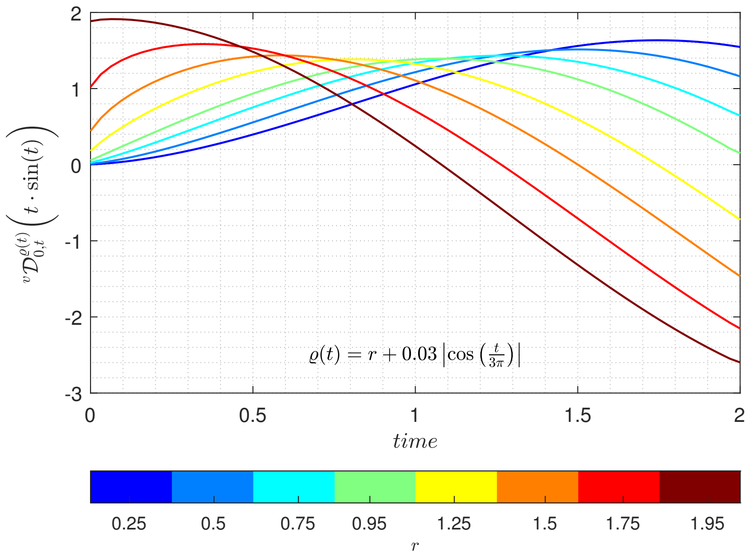
Table 1. Comparison of $\mathscr{E}_M$, ECO, and computational time (based on sec.) of (25) using the IQS algorithm [36] and developed algorithm, with various values of $\varrho(t)$ and Δ in $t\in[0,2]$.

		IQS Algorithm			**Developed Algorithm**		
$\boldsymbol{\varrho(t)}$	$\boldsymbol{\Delta}$	$\mathscr{E}_M$	***ECO***	***CPu Time***	$\mathscr{E}_M$	***ECO***	***CPu Time***
$\varrho_1(t)$	$\frac{1}{16}$	3.65×10^{-3}	2.02	0.719	6.59×10^{-4}	2.64	5.310
	$\frac{1}{32}$	1.23×10^{-3}	1.93	2.672	1.55×10^{-4}	2.53	8.216
	$\frac{1}{64}$	4.13×10^{-4}	1.87	9.563	3.74×10^{-5}	2.45	14.203
	$\frac{1}{128}$	1.38×10^{-4}	1.83	36.954	9.12×10^{-6}	2.39	28.765
$\varrho_2(t)$	$\frac{1}{16}$	3.78×10^{-2}	1.18	0.640	9.07×10^{-4}	2.53	5.281
	$\frac{1}{32}$	1.68×10^{-2}	1.18	2.344	2.12×10^{-4}	2.44	8.187
	$\frac{1}{64}$	7.49×10^{-3}	1.18	8.313	4.98×10^{-5}	2.38	14.109
	$\frac{1}{128}$	3.33×10^{-3}	1.18	31.344	1.17×10^{-5}	2.34	28.562
$\varrho_3(t)$	$\frac{1}{16}$	8.75×10^{-3}	1.71	0.703	9.24×10^{-5}	3.35	6.297
	$\frac{1}{32}$	2.97×10^{-3}	1.68	2.688	1.49×10^{-5}	3.21	8.297
	$\frac{1}{64}$	1.00×10^{-3}	1.66	10.047	2.42×10^{-6}	3.11	14.375
	$\frac{1}{128}$	3.37×10^{-4}	1.64	39.328	3.95×10^{-7}	3.04	29.578
$\varrho_4(t)$	$\frac{1}{16}$	9.18×10^{-2}	0.86	0.688	4.93×10^{-4}	2.75	6.297
	$\frac{1}{32}$	4.11×10^{-2}	0.92	2.516	1.09×10^{-4}	2.63	8.172
	$\frac{1}{64}$	1.83×10^{-2}	0.96	8.172	2.42×10^{-5}	2.56	14.250
	$\frac{1}{128}$	8.17×10^{-3}	0.99	30.203	5.37×10^{-6}	2.50	28.921

In addition, the outcomes are examined in Figures 1a and 2a for $\Delta = \frac{1}{32}$ and various values of $\varrho(t) = r + 0.03\left|\cos\left(\frac{t}{3\pi}\right)\right|$ and $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$. Consequences in Figures 1b and 2b are shown in which the logarithm of absolute error, $\log(AE)$, of the proposed scheme is in the whole interval $t \in [0, 2]$.

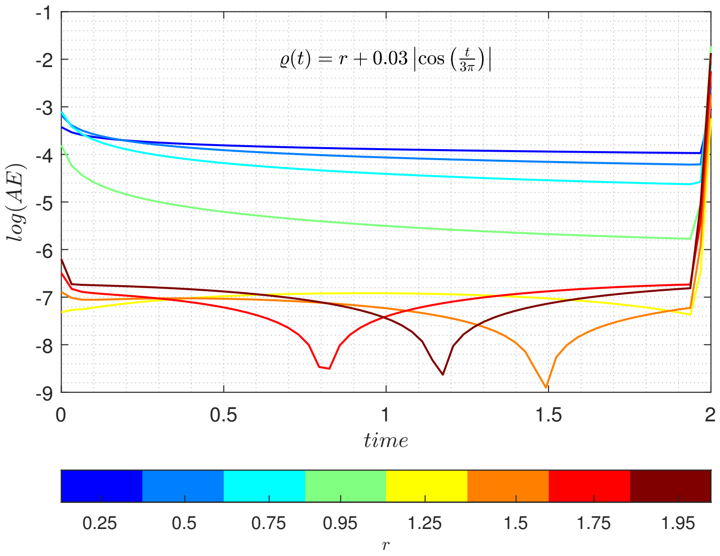
Table 2. Comparison of $\mathscr{E}_M$, *ECO*, and computational time (based on sec.) of (26) using the B-spline algorithm [25] and developed algorithm, with various values of $\varrho(t)$ and Δ in $t \in [0, 2]$.

		B-Spline Algorithm			**Developed Algorithm**		
$\varrho(t)$	Δ	$\mathscr{E}_M$	*ECO*	*CPu Time*	$\mathscr{E}_M$	*ECO*	*CPu Time*
$\varrho_1(t)$	$\frac{1}{16}$	2.85×10^{-4}	2.94	0.328	1.01×10^{-5}	4.15	5.719
	$\frac{1}{32}$	7.28×10^{-5}	2.74	0.672	1.73×10^{-6}	3.83	8.625
	$\frac{1}{64}$	1.86×10^{-5}	2.61	2.594	3.21×10^{-7}	3.59	15.328
	$\frac{1}{128}$	8.73×10^{-6}	2.40	8.610	6.09×10^{-8}	3.42	29.391
$\varrho_2(t)$	$\frac{1}{16}$	2.84×10^{-4}	2.94	0.297	2.87×10^{-6}	4.60	5.921
	$\frac{1}{32}$	7.05×10^{-5}	2.75	0.657	2.88×10^{-7}	4.35	8.734
	$\frac{1}{64}$	1.76×10^{-5}	2.63	2.047	3.46×10^{-8}	4.13	14.687
	$\frac{1}{128}$	4.40×10^{-6}	2.54	7.938	4.54×10^{-9}	3.96	29.359
$\varrho_3(t)$	$\frac{1}{16}$	3.22×10^{-4}	2.90	0.313	1.45×10^{-6}	4.85	5.906
	$\frac{1}{32}$	7.99×10^{-5}	2.72	0.672	8.80×10^{-8}	4.69	9.266
	$\frac{1}{64}$	1.99×10^{-5}	2.39	10.047	5.75×10^{-9}	4.56	14.828
	$\frac{1}{128}$	7.96×10^{-6}	2.42	9.594	4.00×10^{-10}	4.46	29.390
$\varrho_4(t)$	$\frac{1}{16}$	3.07×10^{-4}	2.91	0.312	1.25×10^{-6}	4.90	5.731
	$\frac{1}{32}$	7.57×10^{-5}	2.73	0.641	6.74×10^{-8}	4.76	8.781
	$\frac{1}{64}$	1.88×10^{-5}	2.61	2.532	3.98×10^{-9}	4.65	14.859
	$\frac{1}{128}$	4.68×10^{-6}	2.53	7.390	2.43×10^{-10}	4.56	29.406



(**a**)

Figure 1. *Cont.*



(**b**)

Figure 1. Collation of the analytical and numerical results for (25) applying the developed algorithm with a step size of $\Delta = \frac{1}{32}$ for $\varrho(t) = r + 0.03\left|\cos\left(\frac{t}{3\pi}\right)\right|$, $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$ and $t \in [0, 2]$. (**a**) The logarithm of the absolute error resulting from the numerical computation (25) is depicted using the implemented algorithm. (**b**) The logarithm of the absolute error resulting from the numerical computation (25) is depicted using the implemented algorithm.

Example 2. *Let*

$$^{v}\mathscr{D}_{0,t}^{\varrho(t)}\left(J_0\left(\frac{t}{4}\right)\right) = -\frac{1}{32}\frac{t^{2-\varrho(t)}}{\Gamma(3-\varrho(t))}\begin{cases} {}_2F_3\left([1,\frac{3}{2}],[2,2-\frac{\varrho(t)}{2},\frac{3}{2}-\frac{\varrho(t)}{2}];-\frac{t^2}{64}\right), & if\, 0<\varrho(t)\le 1 \\ \left(2\,{}_1F_2\left(\frac{1}{2};2-\frac{\varrho(t)}{2},\frac{3}{2}-\frac{\varrho(t)}{2};-\frac{t^2}{64}\right)\right. & \\ \quad \left.+{}_2F_3\left(\frac{1}{2},1;2,2-\frac{\varrho(t)}{2},\frac{3}{2}-\frac{\varrho(t)}{2};-\frac{t^2}{64}\right)\right), & if\, 1<\varrho(t)\le 2. \end{cases} \tag{27}$$

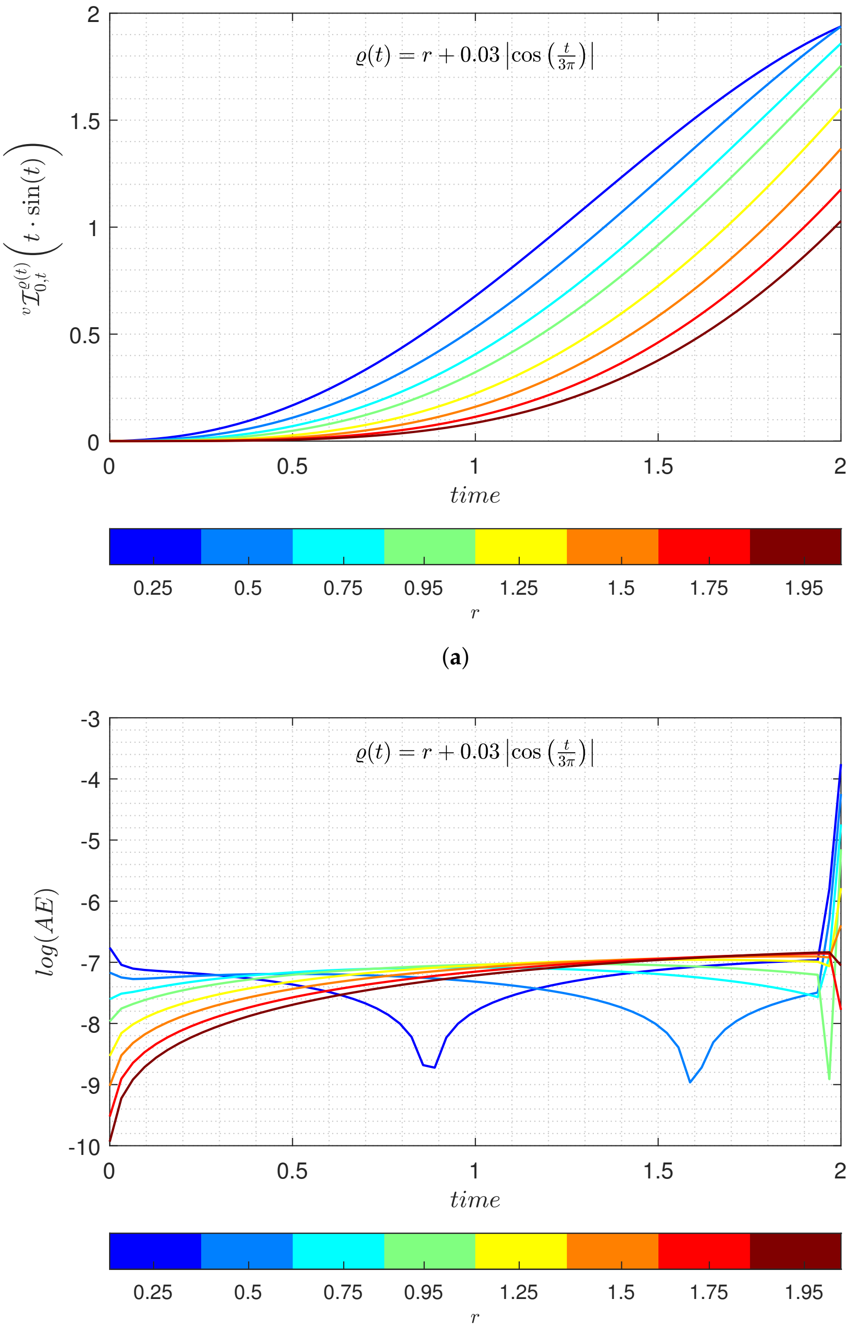
Additionally, for $Re(\varrho(t)) > 0$, we have

$$^{v}\mathscr{I}_{0,t}^{\varrho(t)}\left(J_0\left(\frac{t}{4}\right)\right) = \frac{t^{\varrho(t)}{}_1F_2([\frac{1}{2}],[1+\frac{\varrho(t)}{2},\frac{1}{2}+\frac{\varrho(t)}{2}];-\frac{t^2}{64})}{\Gamma(\varrho(t)+1)}. \tag{28}$$

Similarly, the performance of developed algorithm is presented by $^{v}\mathscr{D}_{0,t}^{\varrho(t)}(J_0(\frac{t}{4}))$ and $^{v}\mathscr{I}_{0,t}^{\varrho(t)}(J_0(\frac{t}{4}))$ in Example 2. Tables 3 and 4 display the values of $\mathscr{E}_M$, *ECO* and computational times (based on sec.) of (27) and (28) with $\Delta = \{\frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}\}$ in $t \in [0, 3]$ for

$$\begin{cases} \varrho_1^*(t) = 0.1 + 0.1t \\ \varrho_2^*(t) = 0.6 - 0.25\sqrt{t} \\ \varrho_3^*(t) = 0.9 - 0.01\left|\cos\left(\frac{t}{3\pi}\right)\right| \\ \varrho_4^*(t) = 0.9 + 0.01\left|\cos\left(\frac{t}{3\pi}\right)\right| \end{cases}.$$

Based on Tables 3 and 4, all results are improved compared to the finite difference [34] and IQS [37] algorithms, respectively. Furthermore, the results are also studied in Figures 3a and 4a for $\Delta = \frac{1}{32}$ and various values of $\varrho^*(t) = r + 0.01\left|\cos\left(\frac{t}{3\pi}\right)\right|$ and $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$. Consequences in Figure 3b and 4b are shown in which the logarithm of absolute error, $\log(AE)$, of the proposed scheme is in the whole interval $t \in [0, 3]$.



(**a**)

(**b**)

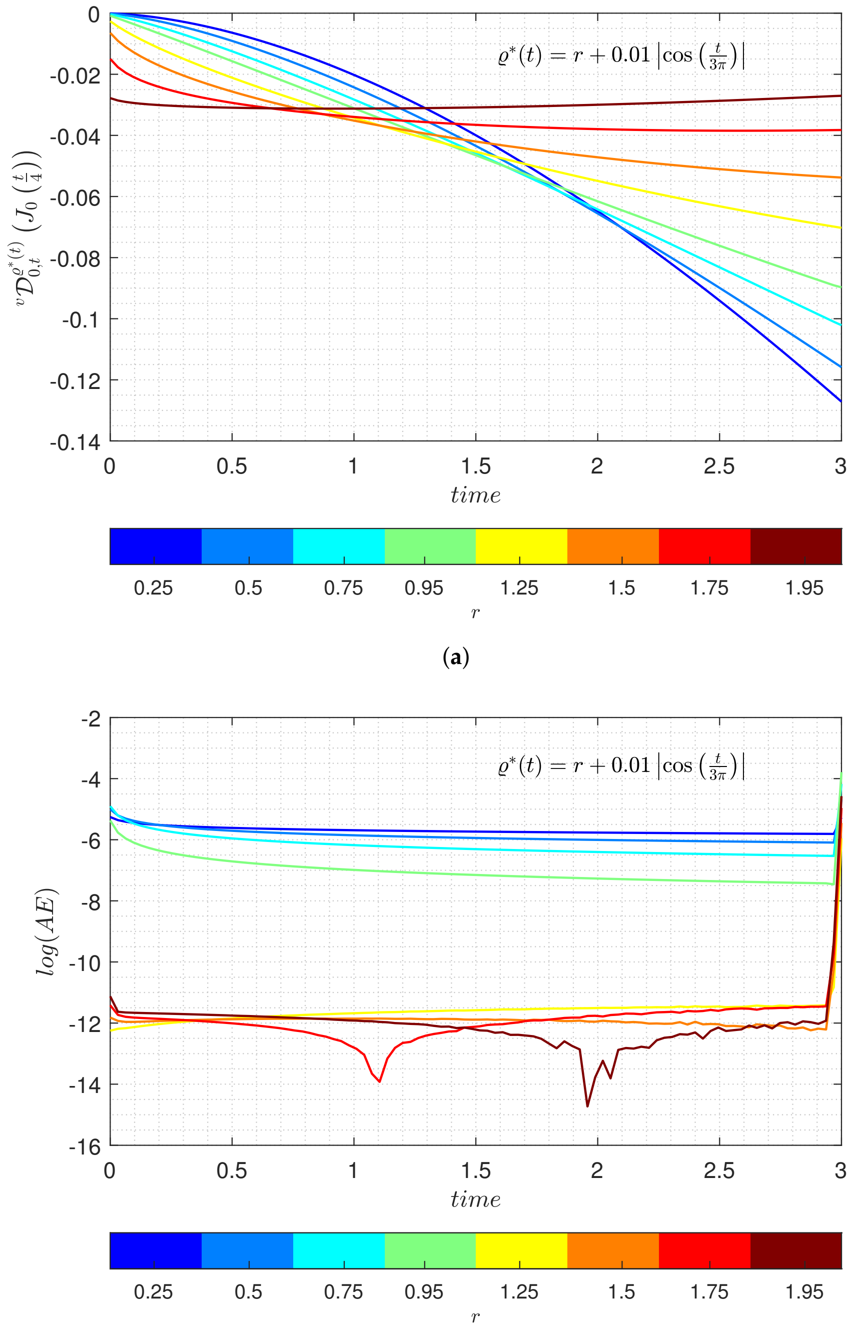
Figure 2. Collation of the analytical and numerical results for (26) applying the developed algorithm with a step size of $\Delta = \frac{1}{32}$ for $\varrho(t) = r + 0.03\left|\cos\left(\frac{t}{3\pi}\right)\right|$, $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$ and $t \in [0, 2]$. (**a**) The computed results for equation (26) obtained through the developed algorithm. (**b**) The logarithm of the absolute error resulting from the numerical computation (26) is depicted using the implemented algorithm.

Table 3. Comparison of $\mathscr{E}_M$, *ECO*, and computational time (based on sec.) of (27) using the Finite difference [34] and developed algorithm, with various values of $\varrho(t)$ and Δ in $t \in [0,3]$.

		Finite Difference Algorithm			**Developed Algorithm**		
$\boldsymbol{\varrho(t)}$	$\boldsymbol{\Delta}$	$\mathscr{E}_M$	***ECO***	***CPu Time***	$\mathscr{E}_M$	***ECO***	***CPu Time***
$\varrho_1^*(t)$	$\frac{1}{16}$	1.31×10^{-3}	2.39	0.297	8.34×10^{-6}	4.22	7.047
	$\frac{1}{32}$	6.42×10^{-4}	2.12	0.829	2.05×10^{-6}	3.78	11.796
	$\frac{1}{64}$	3.18×10^{-4}	1.93	3.141	5.04×10^{-7}	3.49	23.265
	$\frac{1}{128}$	1.58×10^{-4}	1.80	12.485	1.25×10^{-7}	3.28	40.703
$\varrho_2^*(t)$	$\frac{1}{16}$	1.45×10^{-3}	2.36	0.172	8.43×10^{-6}	4.21	6.984
	$\frac{1}{32}$	7.13×10^{-4}	2.09	0.797	2.15×10^{-6}	3.77	11.906
	$\frac{1}{64}$	3.53×10^{-4}	1.92	3.125	5.44×10^{-7}	3.47	23.156
	$\frac{1}{128}$	1.75×10^{-4}	1.79	12.656	1.37×10^{-7}	3.26	40.609
$\varrho_3^*(t)$	$\frac{1}{16}$	1.57×10^{-3}	2.33	0.313	7.13×10^{-6}	4.27	6.953
	$\frac{1}{32}$	7.61×10^{-4}	2.07	0.844	1.73×10^{-6}	3.83	11.781
	$\frac{1}{64}$	3.70×10^{-4}	1.90	2.203	4.19×10^{-7}	3.53	23.312
	$\frac{1}{128}$	1.81×10^{-4}	1.77	12.375	1.02×10^{-7}	3.32	40.671
$\varrho_4^*(t)$	$\frac{1}{16}$	1.60×10^{-3}	2.32	0.281	7.23×10^{-6}	4.27	7.062
	$\frac{1}{32}$	7.81×10^{-4}	2.06	0.797	1.76×10^{-6}	3.82	11.765
	$\frac{1}{64}$	3.81×10^{-4}	1.89	3.156	4.28×10^{-7}	3.53	23.406
	$\frac{1}{128}$	1.86×10^{-4}	1.77	12.266	1.04×10^{-7}	3.31	40.781

Table 4. Comparison of $\mathscr{E}_M$, *ECO*, and computational time (based on sec.) of (28) using the IQS algorithm [37] and the developed algorithm, with various values of $\varrho(t)$ and Δ in $t \in [0,3]$.

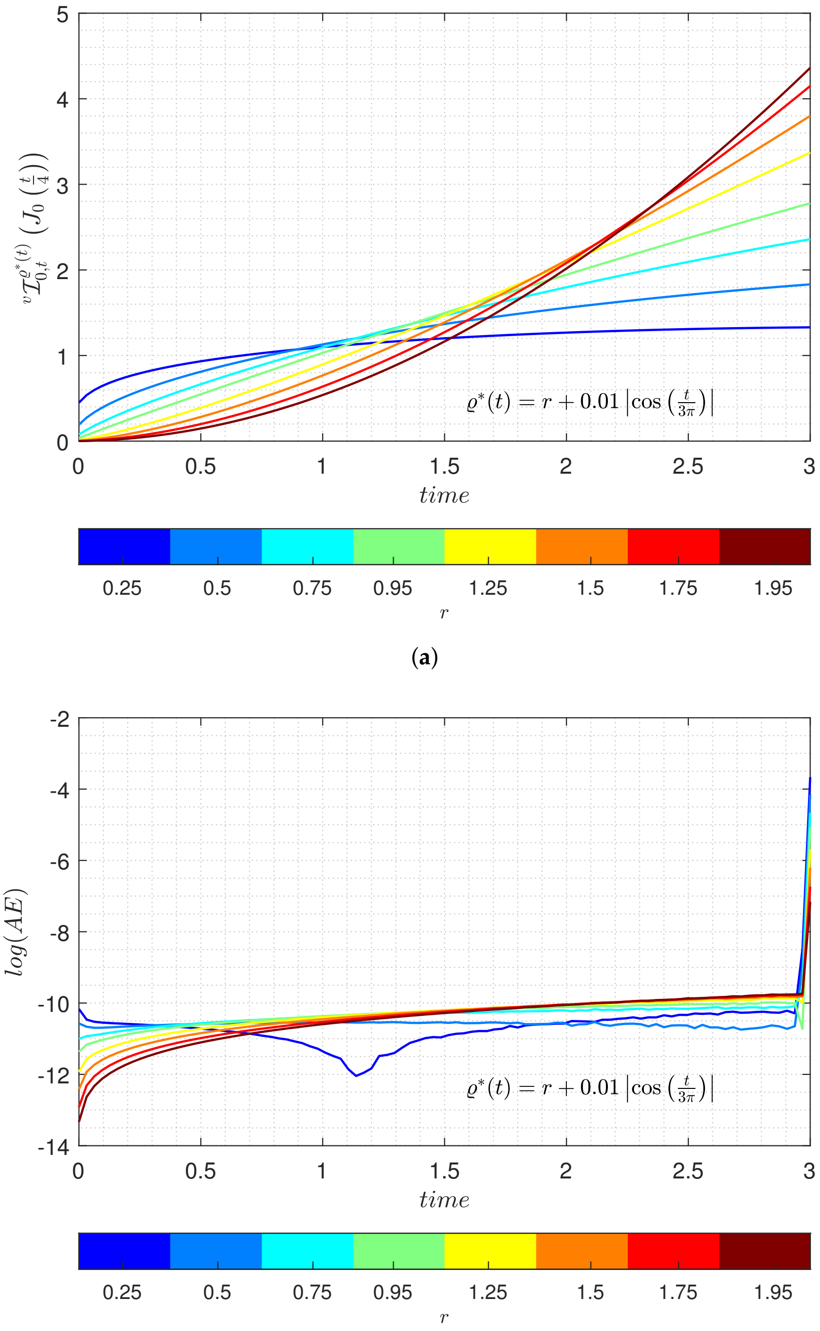
		IQS Algorithm			**Developed Algorithm**		
$\boldsymbol{\varrho(t)}$	$\boldsymbol{\Delta}$	$\mathscr{E}_M$	***ECO***	***CPu Time***	$\mathscr{E}_M$	***ECO***	***CPu Time***
$\varrho_1^*(t)$	$\frac{1}{16}$	1.33×10^{-2}	1.55	1.687	6.29×10^{-6}	4.32	1.968
	$\frac{1}{32}$	6.26×10^{-3}	1.46	5.765	1.19×10^{-6}	3.94	1.765
	$\frac{1}{64}$	2.93×10^{-3}	1.40	21.641	2.26×10^{-7}	3.68	2.062
	$\frac{1}{128}$	1.37×10^{-3}	1.35	86.219	4.28×10^{-8}	3.50	3.281
$\varrho_2^*(t)$	$\frac{1}{16}$	2.26×10^{-3}	2.19	1.234	1.53×10^{-5}	4.00	1.828
	$\frac{1}{32}$	8.09×10^{-4}	2.05	6.328	3.40×10^{-6}	3.63	1.796
	$\frac{1}{64}$	2.97×10^{-4}	1.95	22.703	7.57×10^{-7}	3.39	2.125
	$\frac{1}{128}$	1.12×10^{-4}	1.87	90.031	1.68×10^{-7}	3.21	3.437
$\varrho_3^*(t)$	$\frac{1}{16}$	1.17×10^{-4}	3.26	1.625	9.20×10^{-7}	5.01	1.656
	$\frac{1}{32}$	3.36×10^{-5}	2.97	5.844	1.24×10^{-7}	4.59	1.828
	$\frac{1}{64}$	9.60×10^{-6}	2.77	21.125	1.67×10^{-8}	4.31	2.156
	$\frac{1}{128}$	2.72×10^{-6}	2.64	84.600	2.27×10^{-9}	4.10	3.515
$\varrho_4^*(t)$	$\frac{1}{16}$	9.03×10^{-5}	3.35	1.703	8.53×10^{-7}	5.04	1.687
	$\frac{1}{32}$	2.57×10^{-5}	3.05	5.750	1.13×10^{-7}	4.61	1.760
	$\frac{1}{64}$	7.28×10^{-6}	2.84	21.125	1.51×10^{-8}	4.33	2.156
	$\frac{1}{128}$	2.05×10^{-6}	2.70	85.344	2.03×10^{-9}	4.13	3.453



(**a**)

(**b**)

Figure 3. Collation of the analytical and numerical results for (27) applying the developed algorithm with a step size of $\Delta = \frac{1}{32}$ for $\varrho^*(t) = r + 0.01\left|\cos\left(\frac{t}{3\pi}\right)\right|$, $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$ and $t \in [0, 3]$. (**a**) The computed results for equation (27) obtained through the developed algorithm. (**b**) The logarithm of the absolute error resulting from the numerical computation (27) is depicted using the implemented algorithm.



(**a**)

(**b**)

Figure 4. Collation of the analytical and numerical results for (28) applying the developed algorithm with a step size of $\Delta = \frac{1}{32}$ for $\varrho^*(t) = r + 0.01\left|\cos\left(\frac{t}{3\pi}\right)\right|$, $r = \{0.25, 0.5, 0.75, 0.95, 1.25, 1.5, 1.75, 1.95\}$ and $t \in [0, 3]$. (**a**) The computed results for equation (28) obtained through the developed algorithm. (**b**) The logarithm of the absolute error resulting from the numerical computation (28) is depicted using the implemented algorithm.

4. Conclusions

The application of integro quadratic spline quasi-interpolants to address nonlocal variable-order derivatives and integrals offers a promising solution with versatile applications across scientific domains. Through a comprehensive comparison with alternative algorithms using illustrative examples, we have substantiated the efficacy of our approach across diverse variable-order functions and step sizes. The figures and tables illustrating the results provide compelling evidence of our proposed method's superior performance in terms of accuracy and convergence order. It is imperative to recognize that the convergence order of Integro spline quasi-interpolation has inherent limitations. While we emphasize the merits of our approach, we acknowledge the need to explore its constraints. In contrast, traditional finite difference methods, despite their simplicity, do not necessarily enhance convergence order with increased mesh points; their primary impact lies in reducing approximation error. Our method addresses this limitation by employing integro spline quasi-interpolation, preserving the simplicity characteristic of finite difference methods. Additionally, we conducted a detailed analysis of the implications of varying variable-order function values.

Moreover, our proposed algorithm, based on integro spline quasi-interpolation, showcases potential applications as numerical solvers for variable-order dynamical systems. Furthermore, the adaptability of our method opens avenues for integration with optimization techniques. By employing our algorithm as part of optimization processes, one can explore enhanced solutions for problems involving nonlocal variable-order operators. This intersection of numerical solvers and optimization techniques represents a promising direction for future research, offering the prospect of addressing complex problems in a more efficient and effective manner.

Author Contributions: Research Investigation, Development of Methodology, Initial Manuscript Drafting, and Software: N.T.; Conceptualization, Manuscript Review and Editing, and Supervision: B.P.M.; Data Visualization and Advisory Role: M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable to this article, as no datasets were generated or analyzed during the current study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Moniri, Z.; Moghaddam, B.P.; Roudbaraki, M.Z. An Efficient and Robust Numerical Solver for Impulsive Control of Fractional Chaotic Systems. *J. Funct. Spaces* **2023**, *2023*, 9077924. [CrossRef]
2. Babaei, A.; Ahmadi, M.; Jafari, H.; Liya, A. A Mathematical Model to Examine the Effect of Quarantine on the Spread of Coronavirus. *Chaos Solitons Fractals* **2021**, *142*, 110418. [CrossRef] [PubMed]
3. Dabiri, A.; Butcher, E.A. Numerical solution of multi-order fractional differential equations with multiple delays via spectral collocation methods. *Appl. Math. Model.* **2018**, *56*, 424–448. [CrossRef]
4. Moghaddam, B.P.; Dabiri, A.; Mostaghim, Z.S.; Moniri, Z. Numerical solution of fractional dynamical systems with impulsive effects. *Int. J. Mod. Phys. C* **2023**, *34*, 2350013. [CrossRef]
5. Zahra, W.K.; Ouf, W.A.; El-Azab, M.S. A robust uniform B-spline collocation method for solving the generalized PHI-four equation. *Appl. Appl. Math. Int. J. (AAM)* **2016**, *11*, 24.
6. Zahra, W.K.; Ouf, W.A.; El-Azab, M.S. Cubic B-spline collocation algorithm for the numerical solution of Newell Whitehead Segel type equations. *Electron. J. Math. Anal. Appl.* **2014**, 2, 81–100.
7. Behforooz, H. Approximation by integro cubic splines. *Appl. Math. Comput.* **2006**, *175*, 8–15. [CrossRef]
8. Zhanlav, T.; Mijiddorj, R. Integro cubic splines and their approximation properties. *Appl. Math. Ser. Tver State Univ. Russ.* **2008**, *26*, 65–77. [CrossRef]
9. Behforooz, H. Interpolation by integro quintic splines. *Appl. Math. Comput.* **2010**, *216*, 364–367. [CrossRef]
10. Zhanlav, T.; Mijiddorj, R. The local integro cubic splines and their approximation properties. *Appl. Math. Comput.* **2010**, *216*, 2215–2219. [CrossRef]
11. Lang, F.G.; Xu, X.P. On integro quartic spline interpolation. *J. Comput. Appl. Math.* **2012**, *236*, 4214–4226. [CrossRef]
12. Wu, J.; Zhang, X. Integro sextic spline interpolation and its super convergence. *Appl. Math. Comput.* **2013**, *219*, 6431–6436. [CrossRef]
13. Wu, J.; Zhang, X. Integro quadratic spline interpolation. *Appl. Math. Model.* **2015**, *39*, 2973–2980. [CrossRef]
14. Lang, F.G.; Xu, X.P. On the superconvergence of some quadratic integro-splines at the mid-knots of a uniform partition. *Appl. Math. Comput.* **2018**, *338*, 507–514. [CrossRef]
15. Wu, J.; Shan, T.; Chungang, Z. Integro quadratic spline quasi-interpolants. *J. Syst. Sci. Math. Sci.* **2018**, *38*, 1407.
16. Wu, J.; Ge, W.; Zhang, X. Integro spline quasi-interpolants and their super convergence. *Comput. Appl. Math.* **2020**, *39*, 1–11. [CrossRef]

17. Lang, F.G.; Xu, X.P. Some new super convergence of a quartic integro-spline at the mid-knots of a uniform partition. *ScienceAsia* **2022**, *48*, 479. [CrossRef]
18. Samko, S. Fractional integration and differentiation of variable order: An overview. *Nonlinear Dyn.* **2012**, *71*, 653–662. [CrossRef]
19. Hao, W.K.; Wang, J.S.; Li, X.D.; Song, H.M.; Bao, Y.Y. Probability distribution arithmetic optimization algorithm based on variable order penalty functions to solve combined economic emission dispatch problem. *Appl. Energy* **2022**, *316*, 119061. [CrossRef]
20. Zhang, Q.; Shang, Y.; Li, Y.; Cui, N.; Duan, B.; Zhang, C. A novel fractional variable-order equivalent circuit model and parameter identification of electric vehicle Li-ion batteries. *ISA Trans.* **2020**, *97*, 448–457. [CrossRef]
21. Mehta, P.P.; Pang, G.; Song, F.; Karniadakis, G.E. Discovering a universal variable-order fractional model for turbulent Couette flow using a physics-informed neural network. *Fract. Calc. Appl. Anal.* **2019**, *22*, 1675–1688. [CrossRef]
22. Zhang, S. The uniqueness result of solutions to initial value problems of differential equations of variable-order. *Rev. Real Acad. Cienc. Exactas FíSicas y Naturales. Serie A. MatemáTicas* **2017**, *112*, 407–423. [CrossRef]
23. Sarwar, S. On the Existence and Stability of Variable Order Caputo Type Fractional Differential Equations. *Fractal Fract.* **2022**, *6*, 51. [CrossRef]
24. Telli, B.; Souid, M.S.; Alzabut, J.; Khan, H. Existence and Uniqueness Theorems for a Variable-Order Fractional Differential Equation with Delay. *Axioms* **2023**, *12*, 339. [CrossRef]
25. Moghaddam, B.P.; Yaghoobi, S.; Machado, J.A.T. An Extended Predictor–Corrector Algorithm for Variable-Order Fractional Delay Differential Equations. *J. Comput. Nonlinear Dyn.* **2016**, *11*, 061001. [CrossRef]
26. Zhao, T.; Wu, Y. Hermite Cubic Spline Collocation Method for Nonlinear Fractional Differential Equations with Variable-Order. *Symmetry* **2021**, *13*, 872. [CrossRef]
27. Sun, L.Y.; Lei, S.L.; Sun, H.W. Efficient Finite Difference Scheme for a Hidden-Memory Variable-Order Time-Fractional Diffusion Equation. *Comp. Appl. Math.* **2023**, *42*, 362. [CrossRef]
28. Sun, H.G.; Chang, A.; Zhang, Y.; Chen, W. A Review on Variable-Order Fractional Differential Equations: Mathematical Foundations, Physical Models, Numerical Methods and Applications. *Fract. Calc. Appl. Anal.* **2019**, *22*, 27–59. [CrossRef]
29. Shah, K.; Naz, H.; Sarwar, M.; Abdeljawad, T. On spectral numerical method for variable-order partial differential equations. *AIMS Math.* **2022**, *7*, 10422–10438. [CrossRef]
30. Kadkhoda, N. A numerical approach for solving variable order differential equations using Bernstein polynomials. *Alex. Eng. J.* **2020**, *59*, 3041–3047. [CrossRef]
31. Tural-Polat, S.N.; Dincel, A.T. Numerical solution method for multi-term variable order fractional differential equations by shifted Chebyshev polynomials of the third kind. *Alex. Eng. J.* **2022**, *61*, 5145–5153. [CrossRef]
32. Samko, S.G.; Ross, B. Integration and Differentiation to a Variable Fractional Order. *Integral Transform. Spec. Funct.* **1993**, *1*, 277–300. [CrossRef]
33. Samko, S.G. Fractional Integration and Differentiation of Variable Order. *Ann. Math.* **1995**, *21*, 213–236. [CrossRef]
34. Moghaddam, B.P.; Machado, J.A.T. Extended Algorithms for Approximating Variable Order Fractional Derivatives with Applications. *J. Sci. Comput.* **2016**, *71*, 1351–1374. [CrossRef]
35. Abramowitz, M.; Stegun, I.A. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*; Dover Publications: Mineola, NY, USA, 1964.
36. Moghaddam, B.P.; Machado, J.A.T.; Behforooz, H. An integro quadratic spline approach for a class of variable-order fractional initial value problems. *Chaos Solitons Fractals* **2017**, *102*, 354–360. [CrossRef]
37. Keshi, F.K.; Moghaddam, B.P.; Aghili, A. A numerical approach for solving a class of variable-order fractional functional integral equations. *Comput. Appl. Math.* **2018**, *37*, 4821–4834. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.