mathematics

MDPI

Article

Efficient Hybrid ANN-Accelerated Two-Stage Implicit Schemes for Fractional Differential Equations

Mudassir Shams [1,2] and Bruno Carpentieri [2,*]

[1] Department of Mathematics, Faculty of Arts and Science, Balikesir University, 10145 Balıkesir, Turkey; mudassir.shams@balikesir.edu.tr
[2] Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
* Correspondence: bruno.carpentieri@unibz.it

Abstract

This paper introduces a hybrid two-stage implicit scheme for efficiently solving fractional differential equations, with particular emphasis on fractional initial value problems formulated using the Caputo derivative. Classical numerical approaches to fractional differential equations often encounter challenges related to stability, convergence rate, and memory efficiency. To overcome these limitations, we propose a new discretization framework that directly embeds nonlinear source terms into the time-stepping process, thereby enhancing both stability and accuracy. Our method embeds nonlinear source terms directly into the time-stepping process, enhancing stability and accuracy. Nonlinear systems are efficiently solved using a parallel iterative algorithm with adaptive convergence control, yielding up to 35–50% faster convergence compared with conventional solvers. A rigorous theoretical analysis establishes the scheme's convergence, stability, and consistency, extending earlier proofs to a broader class of fractional systems. Extensive numerical experiments on benchmark fractional problems confirm that the hybrid approach achieves markedly lower local and global errors, broader stability regions, and substantial reductions in computational time and memory usage compared with existing implicit methods. The results demonstrate that the proposed framework offers a robust, accurate, and scalable solution for nonlinear fractional simulations.

Keywords: Caputo fractional derivative; fractional initial value problems; hybrid implicit scheme; convergence and stability analysis; parallel iterative solver; neural network-assisted computation

MSC: 65L05; 65L06; 65L20; 65L99

Academic Editor: Leonid Piterbarg

Received: 12 October 2025
Revised: 18 November 2025
Accepted: 20 November 2025
Published: 24 November 2025

Citation: Shams, M.; Carpentieri, B. Efficient Hybrid ANN-Accelerated Two-Stage Implicit Schemes for Fractional Differential Equations. *Mathematics* **2025**, *13*, 3774. https://doi.org/10.3390/math13233774

1. Introduction

Fractional differential equations (FDEs) have become an essential framework for modeling complex dynamical systems across engineering, physics, and biomedical sciences. In contrast to classical integer-order formulations, fractional derivatives naturally capture the memory and hereditary effects that characterize many real-world processes [1,2]. Their intrinsic ability to represent nonlocal interactions makes fractional models particularly suitable for describing phenomena such as anomalous diffusion, viscoelasticity, and regulatory mechanisms in biomedical systems, including glucose–insulin dynamics [3]. Owing to these capabilities, the analysis and numerical solution of fractional initial value problems have attracted growing attention in recent years.

Analytical solutions of fractional differential equations [4], often expressed through special functions such as the Mittag–Leffler function [5], provide exact representations of system dynamics and serve as valuable benchmarks for validating numerical methods. Fractional-order initial value problems represent a specific class of fractional-order differential equations that involve derivatives of non-integer order [6]. They can be formulated as

$$\begin{cases} D_t^{[\sigma]} y(t) = f(t, y(t)), \\ y^{(k)}(t_0) = y_0^{(k)}, \quad k = 0, 1, \ldots, m-1, \end{cases} \tag{1}$$

where the Caputo fractional derivative of order ϑ $(m-1 < \sigma < m,\ m \in \mathbb{Z}^+)$ is defined by

$$D_t^{[\sigma]} y(t) = \frac{1}{\Gamma(m-\sigma)} \int_{t_0}^{t} \frac{y^{(m)}(h)}{(t-h)^{\sigma-m+1}}\, dh. \tag{2}$$

FOIVPs offer a more general and realistic framework for modeling dynamical systems exhibiting memory and hereditary properties, extending beyond the limitations of classical integer-order formulations. Despite their theoretical significance, obtaining closed-form solutions is often infeasible for real-world nonlinear problems, high-dimensional systems, or models involving complex boundary conditions [6]. This limitation motivates the development of reliable and efficient numerical methods.

Classical analytical approaches such as direct computation [7], Adomian decomposition [8], and variational iteration methods [9] can produce closed-form, series, or integral-transform solutions, but they suffer from several inherent limitations. These techniques are often computationally demanding for large-scale or high-order systems and are generally unsuitable for nonlinear FDEs. In addition, they can become numerically unstable when applied to stiff problems, and the convergence of truncated series is frequently slow, limiting their practicality for real-time applications. Consequently, there remains a need for numerical schemes that combine high accuracy, computational efficiency, and robustness for nonlinear fractional systems. To address these challenges, implicit numerical methods—particularly multi-stage formulations—have attracted increasing attention within the fractional calculus community. Compared with higher-order explicit Runge–Kutta type schemes [10–12], implicit multi-stage approaches [13] incorporate future information at each step, improving both stability and convergence and allowing larger time increments [14,15].

In parallel, artificial neural networks (ANNs) have emerged as powerful tools for approximating nonlinear mappings and accelerating computations across scientific and engineering disciplines [16,17]. Once trained, ANNs can deliver rapid and accurate predictions, providing an effective mechanism to accelerate implicit solvers. Recently, increasing attention has been devoted to integrating data-driven learning models with numerical iterative solvers for fractional differential equations. Physics-Informed Neural Networks (PINNs) address fractional dynamics by embedding physical laws into the network's training loss [18–20]. Other studies have introduced neural Runge–Kutta [21] formulations and data-driven implicit [22] solvers that employ neural networks to improve the temporal integration accuracy of fractional problems or to approximate unknown operators. Despite these advances, most existing hybrid numerical approaches [23] still employ neural networks primarily as surrogate or black-box models that approximate unknown functions or solutions without explicitly enforcing the underlying physical or mathematical principles. This limitation can restrict generalization and accuracy, particularly in complex or fractional systems [24], where learning relies mainly on data rather than the governing equations. In the context of fractional differential equations, neural networks can also approximate

implicit stage terms and provide intelligent initial guesses, thereby reducing the number of iterations required by conventional iterative methods such as Newton–Raphson [25].

Building upon these developments, hybrid strategies that combine artificial neural networks (ANNs) with classical implicit numerical methods have recently emerged as a promising direction. These approaches exploit the fast approximation capability of ANNs along with the high accuracy of iterative refinement, leading to substantial improvements in computational efficiency and solution accuracy [26]. Recent studies have investigated hybrid AI–numerical approaches for solving initial value problems (IVPs) that integrate neural networks with traditional integration schemes. To enhance solution accuracy for IVPs, Wen et al. [27] employ a Lie-group-guided neural network, while Finzi et al. [28] propose a scalable neural IVPs solver with improved stability. Yadav et al. [29] combine harmony search with ANNs to minimize residual errors, whereas Dong & Ni [30] demonstrate that randomized neural networks can learn time-integration techniques, achieving higher long-term accuracy. These studies collectively highlight the potential of hybrid frameworks that embed AI components within numerical solvers to improve convergence and efficiency. However, their performance depends on the availability of an initial approximation sufficiently close to the exact root, since the underlying iterative solvers remain locally convergent. This hybrid paradigm is particularly effective for fractional systems, characterized by strong nonlinearities and long memory effects, where conventional methods often require substantial computational resources [31,32].

Motivated by these challenges, the present study introduces a hybrid ANN-assisted implicit two-stage scheme for solving nonlinear fractional initial value problems. The main goals of the proposed framework are to (i) reduce the computational cost associated with solving implicit stage equations, (ii) preserve high accuracy comparable to analytical or benchmark solutions, and (iii) enhance the convergence and stability properties of the classical two-stage method. The scheme is designed for fractional orders $\sigma \in (0, 1]$, with a representative case study at $\sigma = 0.8$ illustrating its effectiveness.

The main contributions of this study can be summarized as follows:

1. A new two-stage implicit Caputo-type scheme for fractional initial value problems.
2. A feedforward artificial neural network (ANN) integrated with the two-stage fractional solver to efficiently handle nonlinear systems in parallel form.
3. A trained ANN that provides accurate approximations of the implicit stage values, thereby reducing the number of iterative corrections.
4. Comprehensive numerical experiments demonstrating improved convergence, fewer iterations, and higher computational efficiency than classical two-stage methods.
5. Detailed error and stability analyses validating the accuracy of the proposed approach against exact solutions.

The novelty of this work lies in the integration of neural network predictions as intelligent initial guesses for the implicit stage of a fractional multi-stage method. This hybrid strategy combines the global approximation capability of ANNs with the rigorous convergence properties of implicit numerical schemes, achieving a substantial reduction in computational iterations while preserving high accuracy. The proposed scheme is systematically validated through several fractional initial value problems of Caputo type, demonstrating superior performance and robustness compared with existing fractional numerical methods. Overall, the approach represents a promising and effective direction in fractional numerical analysis.

The remainder of this paper is organized as follows: Section 2 outlines the fundamental definitions and mathematical preliminaries related to fractional calculus. Section 3 presents the formulation of the fractional two-stage method together with its stability and consistency analyses. Section 4 describes the proposed hybrid ANN-assisted framework,

including the network architecture, training process, optimization strategy, and implementation details. Section 5 reports the numerical experiments and results, encompassing accuracy comparisons, error analysis, and parallel iteration performance. Finally, Section 6 concludes the paper and discusses potential directions for future research.

2. Preliminaries

This section presents the fundamental concepts and theoretical background underlying the two-stage fractional numerical scheme and its hybrid ANN implementation. The discussion introduces the main definitions of fractional derivatives, key properties of the Gamma function, the generalized Taylor expansion, and the analytical framework used to assess local truncation error, convergence, and stability.

Definition 1 (Caputo Fractional Derivative). *Let $f(t)$ be a sufficiently smooth function defined on the interval $t \in [0, T]$. The Caputo fractional derivative of order $\sigma \in (0, 1]$ is defined as [33,34]*

$$D^{\sigma} f(t) = \frac{1}{\Gamma(1-\sigma)} \int_0^t \frac{f'(s)}{(t-s)^{\sigma}} \, ds, \tag{3}$$

where $\Gamma(\cdot)$ denotes the Gamma function, given for $\gamma > 0$ by

$$\Gamma(\gamma) = \int_0^{\infty} e^{-u} u^{\gamma-1} \, du. \tag{4}$$

The operator D^{σ} acts on continuously differentiable functions $f \in C^1[0, T]$ and reduces to the classical first derivative when $\sigma = 1$. Throughout this work, all fractional operators and parameters are considered within the domain $t \in [0, T]$, $\sigma \in (0, 1]$.

Properties of Caputo Derivative:

1. Linearity: $D^{\sigma}[c_1 y_1(t) + c_2 y_2(t)] = c_1 D^{\sigma} y_1(t) + c_2 D^{\sigma} y_2(t)$.
2. Derivative of a power function: $D^{\sigma} t^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\sigma)} t^{\beta-\sigma}, \quad \beta > \sigma - 1$.
3. Caputo derivative of a constant: $D^{\sigma} C = 0$.

Although this study focuses on the Caputo fractional operator, the proposed framework can be readily adapted to other definitions, such as the Riesz or Riemann–Liouville derivatives, for which several high-order discretization schemes have recently been developed [35,36].

Definition 2 (Generalized Taylor Series for Fractional Derivatives). *Let $0 < \sigma < 1$ and assume that y is sufficiently smooth so that the fractional derivatives $D^{m\sigma} y$ (for $m = 1, \ldots, n+1$) exist and are continuous on $[t_i, t_i + h]$. Then, the fractional Taylor expansion about t_i is given by [37]*

$$\begin{aligned} y(t_i + h) = y(t_i) + \frac{h^{\sigma}}{\Gamma(\sigma+1)} D^{\sigma} y(t_i) + \frac{h^{2\sigma}}{\Gamma(2\sigma+1)} D^{2\sigma} y(t_i) \quad + \cdots + \frac{h^{n\sigma}}{\Gamma(n\sigma+1)} D^{n\sigma} y(t_i) \\ + \frac{h^{(n+1)\sigma}}{\Gamma((n+1)\sigma+1)} D^{(n+1)\sigma} y(\xi), \end{aligned} \tag{5}$$

for some point $\xi \in (t_i, t_i + h)$. The final term represents the remainder in the mean-value form of the fractional Taylor theorem and indicates that the truncation error after n terms is

$$R_{n+1}(h) = \frac{h^{(n+1)\sigma}}{\Gamma((n+1)\sigma+1)} D^{(n+1)\sigma} y(\xi) = \mathcal{O}\big(h^{(n+1)\sigma}\big) \quad \text{as } h \to 0,$$

provided that $D^{(n+1)\sigma}y$ is bounded on $[t_i, t_i + h]$. This series forms the basis of multi-step and two-stage fractional schemes.

Definition 3 (Convergence, Consistency, and Stability)**.** *We define the convergence, consistency, and stability [38] of Caputo fractional schemes as follows:*

Order of Convergence: Let τ_{i+1} denote the local truncation error, defined as the difference between the exact solution and the numerical approximation after a single time step, assuming that all previous steps are exact. A numerical method is said to be of order p if the local truncation error satisfies

$$\tau_{i+1} = \mathcal{O}(h^p), \quad \text{as } h \to 0. \tag{6}$$

Consistency: A numerical scheme is consistent if

$$\lim_{h\to 0} \tau_{i+1} = 0, \quad \forall i, \tag{7}$$

ensuring that the discrete scheme approximates the continuous fractional initial value problem.

Stability: Stability of fractional schemes can be analyzed using zero-stability and absolute stability criteria [39]:

- *Absolute Stability: For the test equation $D^\sigma y = \lambda y$, a scheme is absolutely stable if $|y_i| \to 0$ as $i \to \infty$ whenever $\Re(\lambda) < 0$.*
- *Stability Polynomial. Consider the linear test equation*

$$D_t^\sigma y(t) = \lambda y(t), \quad \lambda \in \mathbb{C}, \tag{8}$$

where λ is a complex constant characterizing the stiffness or oscillatory behavior of the system, and $y_i \approx y(t_i)$ denotes the numerical approximation at time level t_i. For such problems, the stability function $R(z)$ of the numerical method is defined as

$$y_{i+1} = R(z)\, y_i, \quad z = h^\sigma \lambda, \tag{9}$$

and the method is said to be stable if $|R(z)| \leq 1$ for the relevant values of z in the complex plane.

Definition 4. *Lax Equivalence Theorem for Fractional IVPs [40].*

Theorem 1. *For a linear consistent fractional difference scheme applied to a well-posed linear FOIVPs, stability is a necessary and sufficient condition for convergence. That is, if the scheme is consistent and stable, then*

$$\lim_{h\to 0} \max_i |y_i - y(t_i)| = 0. \tag{10}$$

Definition 5. *Boundedness Fractional Scheme. Consider a fractional-order numerical scheme generating approximations $\{y_i\}_{i\geq 0}$ for the FOIVPs*

$$D^\sigma y(t) = f(t, y(t)), \quad y(0) = y_0, \tag{11}$$

where $f(t, y)$ is Lipschitz continuous in y with Lipschitz constant L. Then, for any two solutions y_i and $\tilde{y}_i$ of the scheme (possibly with perturbations), the scheme satisfies

$$|y_{i+1} - \tilde{y}_{i+1}| \leq (1 + C\, h^\sigma L)|y_i - \tilde{y}_i|, \tag{12}$$

where $C > 0$ depends on the specific scheme. This inequality guarantees the stability of the method.

The preceding definitions, lemmas, and theorems establish the theoretical foundation required to develop and analyze the proposed two-stage fractional method. They guarantee the scheme's consistency, stability, and convergence, and form the mathematical basis for the hybrid ANN-assisted acceleration introduced in the next section.

3. Construction and Analysis of Two-Stage Fractional Scheme

Fractional-order differential equations with Caputo derivatives are commonly used to model systems exhibiting memory and hereditary effects in science and engineering. Numerical schemes for these equations generalize classical methods to fractional orders, where the nonlocal character of the derivative introduces additional challenges for accuracy, stability, and convergence. Such methods can be formulated in terms of increment functions, analogous to those in classical Runge–Kutta schemes [41], providing a compact framework for solving fractional initial order value problems.

The fractional backward implicit Euler method (IEBF$^{[*]}$), proposed by Batiha et al. [42], is defined as

$$y_{i+1} = y_i + \frac{h^\sigma}{\Gamma(\sigma+1)} k_1, \quad k_1 = f(t_{i+1}, y_{i+1}). \tag{13}$$

This method possesses a convergence order of $\mathcal{O}(h^\sigma)$. It is consistent since

$$\lim_{h\to 0} \frac{y(t_{i+1}) - y_i}{h^\sigma} - f(t_{i+1}, y_{i+1}) = 0,$$

ensuring that the discrete scheme approximates the continuous FOIVPs as $h \to 0$.

To enhance accuracy, Batiha et al. [42] also introduced the fractional trapezoidal implicit two-stage scheme (ITBF$^{[*]}$), which incorporates a midpoint-type correction. It is expressed as

$$y_{i+1} = y_i + \frac{h^\sigma}{2\Gamma(\sigma+1)} (k_1 + k_2), \tag{14}$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i), \\ k_2 &= f\Big(x_i + \frac{h^\sigma}{\Gamma(\sigma+1)}, y_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}(k_1 + k_2)\Big). \end{aligned}$$

The fractional trapezoidal rule achieves a convergence order of $\mathcal{O}(h^{2\sigma})$ under sufficient smoothness of f. Its local truncation error satisfies $\tau_{i+1} = \mathcal{O}(h^{2\sigma})$, confirming consistency as $h \to 0$.

Overall, these two schemes highlight the trade-off between simplicity and accuracy: the fractional method IEBF$^{[*]}$ achieves an accuracy of order σ, whereas ITBF$^{[*]}$ attains an accuracy of order 2σ, both measured with respect to the fractional step size h^σ. In the classical case $\sigma = 1$, these orders reduce to first- and second-order accuracy, respectively.

We propose the following two-stage modified trapezoidal-type scheme (ISBF$^{[*]}$) for the Caputo fractional IVP:

$$y_{i+1} = y_i + \Phi(y : h), \quad 0 < \sigma \leq 1, \tag{15}$$

where

$$\begin{aligned} k_1 &= f(x_i, y_i), \\ k_2 &= f\Big(x_i + \frac{h^\sigma}{\Gamma(\sigma+1)}, y_i + \frac{h^\sigma}{2\,\Gamma(\sigma+1)}(\alpha_1 k_1 + \alpha_2 k_2)\Big), \end{aligned}$$

and

$$\Phi(y:h) = \frac{h^{\sigma}}{2\,\Gamma(\sigma+1)}(k_1+k_2), \alpha_1, \alpha_2 \in R,$$

represents the increment function of the scheme.

Assumption 1. *Let $f(x,y)$ satisfy the following regularity conditions:*

1. *$f(x,y)$ is continuous in x and satisfies a Lipschitz condition in y, i.e., there exists a constant $L>0$ such that*
$$\|f(x,y_1) - f(x,y_2)\| \le L\|y_1 - y_2\|, \quad \forall y_1, y_2.$$
2. *$f(x,y)$ has continuous fractional derivatives up to the required order σ, ensuring that the Caputo derivative exists and remains bounded.*
3. *The exact solution $y(x)$ is sufficiently smooth so that $y^{(\sigma)}(x)$ is continuous on the interval of interest.*

Under these assumptions, the convergence of the proposed scheme can be established as follows:

Theorem 2. *Let the numerical scheme be defined as*

$$y_{i+1} = y_i + \Phi(y:h), \tag{16}$$

where

$$k_1 = f(x_i, y_i), \qquad k_2 = f\Big(x_i + \frac{h^{\sigma}}{\Gamma(\sigma+1)}, y_i + \frac{h^{\sigma}}{2\Gamma(\sigma+1)}\Big(\alpha_1 k_1 + \alpha_2 k_2\Big)\Big).$$

Then, the proposed fractional implicit scheme is consistent, with a local truncation error of order $\mathcal{O}\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big)$ and a convergence order of 2σ.

Proof. Expanding the exact solution $y(x+h)$ in a fractional Taylor series gives

$$y_{i+1} = f + \tfrac{1}{2}(fD_y^{[\sigma]}f)\Big(\tfrac{h^{\sigma}}{\Gamma(\sigma+1)}\Big) + \Big[\tfrac{1}{6}f^2D_{yy}^{[\sigma]} + \tfrac{1}{6}f(D_y^{[\sigma]}f)^2\Big]\Big(\tfrac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big) + \cdots. \tag{17}$$

Next, expanding k_2 in a similar manner, we have

$$k_2 = f + \Big(\frac{\alpha_1}{2}fD_y^{[\sigma]}f + \frac{\alpha_2}{2}D_y^{[\sigma]}fk_2\Big)\Big(\frac{h^{\sigma}}{\Gamma(\sigma+1)}\Big) + B_1\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big) + \cdots \tag{18}$$

Assume the fractional series form

$$k_2 = \vartheta_1 + \frac{h^{\sigma}}{\Gamma(\sigma+1)}\vartheta_2 + \frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\vartheta_3 + \frac{h^{3\sigma}}{\Gamma(3\sigma+1)}\vartheta_4 + \cdots. \tag{19}$$

Substituting (19) into (18) and matching terms in powers of h^{σ}, we obtain

$$\begin{aligned} k_{12} = f &+ \big(\tfrac{\alpha_1}{2}fD_y^{[\sigma]}f + \tfrac{\alpha_2}{2}D_y^{[\sigma]}f\vartheta_1\big)\Big(\frac{h^{\sigma}}{\Gamma(\sigma+1)}\Big) + B_2\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big) \\ &+ B_3\Big(\frac{h^{3\sigma}}{\Gamma(3\sigma+1)}\Big) + O\Big(\frac{h^{4\sigma}}{\Gamma(4\sigma+1)}\Big)D^{[3\sigma]}f(\zeta), \end{aligned} \tag{20}$$

where the coefficients $B_1, B_2, \ldots, B_5$ are explicitly derived in Appendix A. By comparing (19) and (20), the coefficients are found as

$$\begin{aligned}
\vartheta_1 &= f,\\
\vartheta_2 &= \tfrac{\alpha_1}{2} f D_y^{[\sigma]} f + \tfrac{\alpha_2}{2} D_y^{[\sigma]} f,\\
\vartheta_3 &= \tfrac{\alpha_1\alpha_2}{4} f (D_y^{[\sigma]} f)^2 + \tfrac{\alpha_1}{2} f^2 D_{yy}^{[\sigma]} f + \tfrac{\alpha_1^2}{4} f D_{yy}^{[\sigma]} f + \tfrac{\alpha_2^2}{2} f^2 D_{yy}^{[\sigma]} f,\\
\vartheta_4 &= \alpha_2 \tfrac{f_y}{4}\Big[\tfrac{\alpha_1}{4} f (D_y^{[\sigma]} f)^2 + \tfrac{\alpha_2}{2} f^2 D_{yy}^{[\sigma]} f\Big] + \tfrac{\alpha_1\alpha_2 f^2 D_y^{[\sigma]} f D_{yy}^{[\sigma]} f}{4} + \tfrac{\alpha_2^2 f^3 D_{yyy}^{[\sigma]} f}{6}.
\end{aligned} \tag{21}$$

Substituting (21) into (19), we obtain

$$\begin{aligned}
k_2 = f + (\tfrac{\alpha_1}{2} f D_y^{[\sigma]} f + \tfrac{\alpha_2}{2} f D_y^{[\sigma]} f)\Big(\frac{h^\sigma}{\Gamma(\sigma+1)}\Big) + \Big[\tfrac{\alpha_2}{4} f (D_y^{[\sigma]} f)^2 + \tfrac{\alpha_1^2}{2} f^2 D_{yy}^{[\sigma]} f\Big]\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big)\\
+ B_4\Big(\frac{h^{3\sigma}}{\Gamma(3\sigma+1)}\Big) + O\Big(\frac{h^{4\sigma}}{\Gamma(4\sigma+1)}\Big) D^{[3\sigma]} f(\zeta).
\end{aligned} \tag{22}$$

Substituting k_1 and k_2 in (16) gives

$$\begin{aligned}
\Phi(y;h) = f + \tfrac{(\alpha_1+\alpha_2)}{2}(f D_y^{[\sigma]} f)\Big(\frac{h^\sigma}{\Gamma(\sigma+1)}\Big) + \Big[\tfrac{\alpha_2}{8} f (D_y^{[\sigma]} f)^2 + \tfrac{\alpha_1^2}{4} f^2 D_{yy}^{[\sigma]} f\Big]\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big)\\
+ B_5\Big(\frac{h^{3\sigma}}{\Gamma(3\sigma+1)}\Big) + O\Big(\frac{h^{4\sigma}}{\Gamma(4\sigma+1)}\Big) D^{[3\sigma]} f(\zeta).
\end{aligned} \tag{23}$$

Subtracting (23) from (17) gives the local truncation error (LTE):

$$\begin{aligned}
\text{L.T.E.} = \Big(\tfrac{1}{2} - \tfrac{\alpha_1+\alpha_2}{4}\Big) f D_y^{[\sigma]} f \frac{h^\sigma}{\Gamma(\sigma+1)} + \Big[-\tfrac{\alpha_1\alpha_2}{12} f^2 D_{yy}^{[\sigma]} f + \tfrac{\alpha_2}{14} f (D_y^{[\sigma]} f)^2\Big]\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\\
+ O\Big(\frac{h^{3\sigma}}{\Gamma(3\sigma+1)}\Big) D^{[2\sigma]} f(\zeta).
\end{aligned} \tag{24}$$

Imposing $\alpha_1 + \alpha_2 = 2$ leaves one equation with two parameters. Choosing $\alpha_1 = \frac{3}{2}$ gives $\alpha_2 = \frac{1}{2}$. Substituting these values yields the two-stage fractional scheme:

$$y_{i+1} = y_i + \Phi(y : h), \quad 0 < \sigma \leq 1, \tag{25}$$

where

$$\begin{aligned}
k_1 &= f(x_i, y_i),\\
k_2 &= f\Big(x_i + \frac{h^\sigma}{\Gamma(\sigma+1)}, y_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}(\tfrac{3}{2}k_1 + \tfrac{1}{2}k_2)\Big),
\end{aligned}$$

and

$$\Phi(y : h) = \frac{h^\sigma}{2\Gamma(\sigma+1)}(k_1 + k_2).$$

Hence, the proposed fractional scheme has a local truncation error of order $\mathcal{O}\Big(\frac{h^{2\sigma}}{\Gamma(2\sigma+1)}\Big) \sim \mathcal{O}(h^{2\sigma})$, and therefore exhibits a consistency order of 2σ. □

Remark 1. *The derived convergence order 2σ generalizes the classical result for integer-order implicit Runge–Kutta methods, where $\sigma = 1$ yields the standard second-order accuracy. The proposed implicit framework achieves higher accuracy and precision with reduced computational cost, owing to its compact two-stage structure and ANN-assisted initialization. Moreover, the implicit two-stage fractional formulation inherently incorporates the memory effect: each time-step update involves fractional weights of the form $h^\sigma/\Gamma(\sigma+1)$, originating from the memory kernel of the Caputo operator (4), thereby coupling the current solution $y(t_{i+1})$ with all past evaluations of $f(t, y)$.*

3.1. Boundedness, Stability, and Convergence

Let $E_i = y(x_i) - y_i$ denote the global error. From the recursive form (16) and using the Lipschitz condition $|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$, one can show

$$|E_{i+1}| \leq (1 + h^\sigma L/\Gamma(\sigma+1))|E_i| + Ch^{2\sigma},$$

which ensures boundedness if $h^\sigma L/\Gamma(\sigma+1) < 1$. Applying discrete Grönwall's inequality yields

$$|E_i| \leq \frac{C\,h^{2\sigma}}{1 - h^\sigma L/\Gamma(\sigma+1)},$$

proving the global stability and convergence of order 2σ. Thus, the scheme is both consistent and zero-stable, guaranteeing overall convergence in the sense of Dahlquist's criterion [43]. Considering the two-stage fractional scheme ISBF$^{[*]}$ for FOIVPs as

$$\begin{cases} y_{i+1} = y_i + \dfrac{h^\sigma}{2\Gamma(\sigma+1)}(k_1 + k_2), \\ k_1 = f(x_i, y_i), \\ k_2 = f\Big(x_i + h, y_i + \dfrac{h^\sigma}{2\Gamma(\sigma+1)}\Big(\dfrac{3}{2}k_1 + \dfrac{1}{2}k_2\Big)\Big), \end{cases} \tag{26}$$

where $0 < \sigma \leq 1$ and $f(x, y)$ is assumed to satisfy a Lipschitz condition in y with constant $L > 0$, i.e.,

$$|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|, \quad \forall y_1, y_2. \tag{27}$$

Now, we check the boundness in the second stage k_2. From the implicit definition of k_2 in (26), consider two sequences y_i and $\tilde{y}_i$ (perturbed initial values). Using the Lipschitz condition, we obtain

$$\begin{aligned} |k_2 - \tilde{k}_2| = \Bigg| & f\Big(x_i + h, y_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}\Big(\frac{3}{2}k_1 + \frac{1}{2}k_2\Big)\Big) \\ & - f\Big(x_i + h, \tilde{y}_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}\Big(\frac{3}{2}\tilde{k}_1 + \frac{1}{2}\tilde{k}_2\Big)\Big)\Bigg| \\ \leq L\Big| & (y_i - \tilde{y}_i) + \frac{h^\sigma}{2\Gamma(\sigma+1)}\Big(\frac{3}{2}(k_1 - \tilde{k}_1) + \frac{1}{2}(k_2 - \tilde{k}_2)\Big)\Big|. \end{aligned} \tag{28}$$

From (26), the difference between successive steps satisfies

$$\begin{aligned} |y_{i+1} - \tilde{y}_{i+1}| &= |y_i - \tilde{y}_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}((k_1 - \tilde{k}_1) + (k_2 - \tilde{k}_2))| \\ &\leq |y_i - \tilde{y}_i| + \frac{h^\sigma}{2\Gamma(\sigma+1)}(|k_1 - \tilde{k}_1| + |k_2 - \tilde{k}_2|) \\ &\leq \Big(1 + Ch^\sigma\Big)|y_i - \tilde{y}_i|, \end{aligned} \tag{29}$$

where C depends on the Lipschitz constant L and $\Gamma(\sigma+1)$. This inequality shows that the growth of perturbations is bounded at each step.

Applying (29) recursively for n steps gives

$$|y_n - \tilde{y}_n| \leq (1 + Ch^\sigma)^n |y_0 - \tilde{y}_0|. \tag{30}$$

For sufficiently small step size h, the factor $(1 + Ch^\sigma)^n$ remains bounded, ensuring that the numerical solution does not grow unboundedly. Therefore, the scheme (26) is bounded.

3.2. Stability Analysis

To analyze the stability of the proposed scheme

$$y_{i+1} = y_i + \Phi(y : h), \quad \Phi(y : h) = \frac{h^\sigma}{2\Gamma(\sigma+1)}(k_1 + k_2), \tag{31}$$

with

$$k_1 = f(x_i, y_i), \quad k_2 = f\Big(x_i + \frac{h^\sigma}{\Gamma(\sigma+1)}, y_i + \frac{h^\sigma}{2\Gamma(\sigma+1)}(\frac{3}{2}k_1 + \frac{1}{2}k_2)\Big),$$

We employ the standard linear test equation

$$D^\sigma y(t) = \lambda y(t), \tag{32}$$

where D^σ denotes the Caputo fractional derivative of order $0 < \sigma \leq 1$ and $\lambda \in \mathbb{C}$. This equation serves as a canonical model for analyzing the stability and convergence properties of fractional numerical schemes. The corresponding right-hand side can therefore be expressed as

$$f(x, y) = \lambda y,$$

which is used in the subsequent derivation of the stability function.

Substituting into the scheme, we obtain

$$k_2 = \frac{\lambda y_i\Big(1 + \frac{3\lambda h^\sigma}{4\Gamma(\sigma+1)}\Big)}{1 - \frac{\lambda h^\sigma}{4\Gamma(\sigma+1)}}. \tag{33}$$

Hence, the stability function of the scheme is

$$R(z) = 1 + \frac{z}{2\Gamma(\sigma+1)}\frac{2 + z/\Gamma(\sigma+1)}{1 - z/(4\Gamma(\sigma+1))}, \quad z = \lambda h^\sigma. \tag{34}$$

The stability region is defined as [44]:

$$\mathcal{S} = \{z \in \mathbb{C} : |R(z)| < 1\}. \tag{35}$$

For fractional orders $\sigma \in (0, 1]$, the stability region can be visualized in the complex z-plane. The scheme remains stable for all z within $\mathcal{S}$, and the region gradually shrinks as σ decreases, confirming the expected stability trend of the proposed fractional scheme ISBF$^{[*]}$. The stability function of the IEBF$^{[*]}$ scheme is given by

$$R(z) = \frac{1}{1 - \frac{z}{\Gamma(\sigma+1)}}, \tag{36}$$

whereas that of the ITBF$^{[*]}$ scheme is

$$R(z) = \frac{1 + \frac{z}{2\Gamma(\sigma+1)}}{1 - \frac{z}{2\Gamma(\sigma+1)}}. \tag{37}$$

Numerical Comparison on the Real Axis

We tabulate, for each chosen σ, the factor $\Gamma(\sigma+1)$ and the numerically determined left endpoint z_{left} on the real axis for the new method ISBF$^{[*]}$, i.e., the most negative real z for which $|R_{\text{new}}(z)| \leq 1$ in Table 1 and Figures 1 and 2.

Table 1. Summary of stability on the real z-axis, showing the numerical values of z_{left} for the ISBF$^{[*]}$ scheme.

σ	$\Gamma(\sigma+1)$	ISBF$^{[*]}$:	Stability Summary
0.10	$\Gamma(1.10) \approx 0.95135$	$z_{\text{left}} \approx -1.902$	$z \in [-1.902, 0]$
0.30	$\Gamma(1.30) \approx 0.89747$	$z_{\text{left}} \approx -1.794$	$z \in [-1.794, 0]$
0.50	$\Gamma(1.50) \approx 0.88623$	$z_{\text{left}} \approx -1.772$	$z \in [-1.772, 0]$
0.70	$\Gamma(1.70) \approx 0.90863$	$z_{\text{left}} \approx -1.817$	$z \in [-1.817, 0]$
0.90	$\Gamma(1.90) \approx 0.96177$	$z_{\text{left}} \approx -1.923$	$z \in [-1.923, 0]$

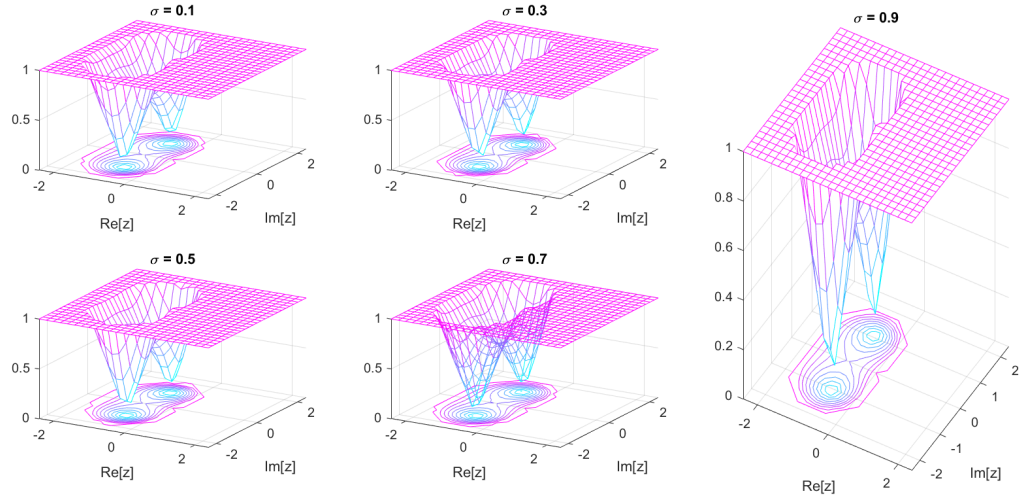


Figure 1. Stability regions of the proposed ISBF$^{[*]}$ scheme in the complex z-plane for different fractional orders $\sigma \in (0, 1]$. The plots illustrate the variation of the stability domain as σ increases.

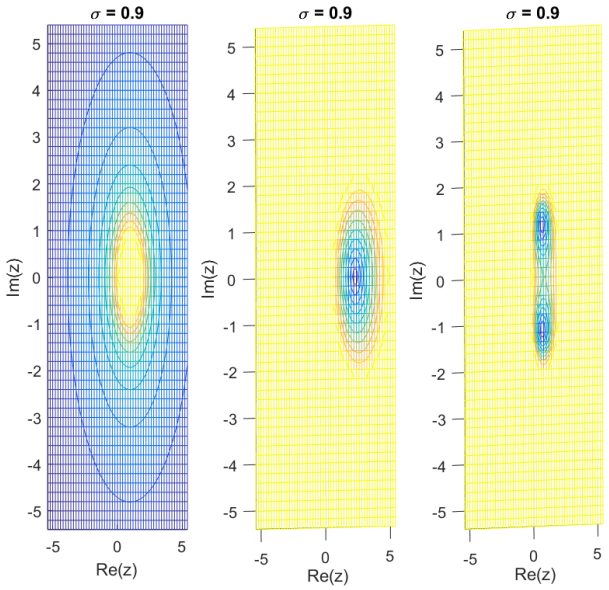


Figure 2. Stability regions of the IEBF$^{[*]}$, ITBF$^{[*]}$, and ISBF$^{[*]}$ schemes in the complex $z-$plane for $\sigma = 0.9$.

A comparison with other existing schemes is given in Table 2 and Figure 2.

Table 2. Comparison of stability characteristics of the IEBF$^{[*]}$, ITBF$^{[*]}$, and ISBF$^{[*]}$ schemes for different fractional orders σ.

Method	Stability Function $R(z)$	Region of Stability	Remarks/Performance
	$\sigma = 0.1$		
IEBF$^{[*]}$	$R(z) = \dfrac{1}{1-\frac{z}{\Gamma(1.1)}}$	Entire left-half plane	Highly stable, strongly damped
ITBF$^{[*]}$	$R(z) = \dfrac{1+\frac{z}{2\Gamma(1.1)}}{1-\frac{z}{2\Gamma(1.1)}}$	Large left-half plane Stable	Stabel
ISBF$^{[*]}$	$R(z) = 1 + \dfrac{z}{2\Gamma(1.1)} \dfrac{2+z/\Gamma(1.1)}{1-z/(4\Gamma(1.1))}$	Broad left-half plane	Best accuracy, balanced damping
	$\sigma = 0.3$		
IEBF$^{[*]}$	$\dfrac{1}{1-\frac{z}{\Gamma(1.3)}}$	Left-half plane	Stable but low accuracy
ITBF$^{[*]}$	$\dfrac{1+\frac{z}{2\Gamma(1.3)}}{1-\frac{z}{2\Gamma(1.3)}}$	Moderate left-half plane	Mild oscillations, better accuracy
ISBF$^{[*]}$	$1 + \dfrac{z}{2\Gamma(1.3)} \dfrac{2+z/\Gamma(1.3)}{1-z/(4\Gamma(1.3))}$	Wider left-half plane	Best stability-accuracy tradeoff
	$\sigma = 0.5$		
IEBF$^{[*]}$	$\dfrac{1}{1-\frac{z}{\Gamma(1.5)}}$	Left-half plane Strong damping	Stabel
ITBF$^{[*]}$	$\dfrac{1+\frac{z}{2\Gamma(1.5)}}{1-\frac{z}{2\Gamma(1.5)}}$	Symmetric about real axis	Second-order accuracy
ISBF$^{[*]}$	$1 + \dfrac{z}{2\Gamma(1.5)} \dfrac{2+z/\Gamma(1.5)}{1-z/(4\Gamma(1.5))}$	Largest left-half plane region	Highest accuracy, robust damping
	$\sigma = 0.7$		
IEBF$^{[*]}$	$\dfrac{1}{1-\frac{z}{\Gamma(1.7)}}$	Left-half plane	Stable but less accurate
ITBF$^{[*]}$	$\dfrac{1+\frac{z}{2\Gamma(1.7)}}{1-\frac{z}{2\Gamma(1.7)}}$	Moderate region	Accurate but near marginal stability
ISBF$^{[*]}$	$1 + \dfrac{z}{2\Gamma(1.7)} \dfrac{2+z/\Gamma(1.7)}{1-z/(4\Gamma(1.7))}$	Wide left-half region	Stable, more accurate than both
	$\sigma = 0.9$		
IEBF$^{[*]}$	$\dfrac{1}{1-\frac{z}{\Gamma(1.9)}}$	Left-half plane	Stable but slow convergence
ITBF$^{[*]}$	$\dfrac{1+\frac{z}{2\Gamma(1.9)}}{1-\frac{z}{2\Gamma(1.9)}}$	Nearly symmetric region	Accurate but can oscillate
ISBF$^{[*]}$	$1 + \dfrac{z}{2\Gamma(1.9)} \dfrac{2+z/\Gamma(1.9)}{1-z/(4\Gamma(1.9))}$	Largest region among all	Excellent stability and accuracy

4. Fractional Implicit Hybrid Scheme, Methodology, and Implementation

This section presents the development and implementation of the proposed fractional implicit hybrid scheme, which integrates the two-stage Caputo-type formulation ISBF$^{[*]}$ with an artificial neural network (ANN)-assisted strategy. The resulting method, referred to as ISBF-AN$^{[*]}$, combines the high-order accuracy of the implicit fractional scheme with the adaptive learning and approximation capabilities of ANNs to improve convergence and stability. The detailed computational framework, algorithmic steps, and implementation procedure are described in the following subsections.

4.1. Methodology

This subsection outlines the detailed methodology of the hybrid ANN-assisted two-stage fractional approach. It first reviews the classical two-stage fractional scheme, then describes the design and training of the artificial neural network (ANN), and finally presents the hybrid framework that integrates ANN predictions with the parallel refinement process.

Consider the nonlinear fractional initial value problem

$$D^{\sigma}y(t) = f(t), \quad y(0) = y_0, \quad 0 < \sigma \leq 1, \tag{38}$$

where D^{σ} denotes the Caputo fractional derivative of order σ, and $f(t)$ is a known function. The goal is to approximate $y(t)$ numerically while achieving high accuracy and computational efficiency.

4.1.1. Two-Stage Implicit Fractional Method

The classical two-stage implicit method discretizes the problem over a uniform time grid $t_n = nh$, $n = 0, 1, \ldots, N$, with step size $h = (t_f - t_0)/N$. The method is defined as

$$y_{n+1} = y_n + \frac{h^{\sigma}}{2\Gamma(\sigma+1)}(k_1 + k_2), \tag{39}$$

$$k_1 = f(t_n, y_n), \tag{40}$$

$$k_2 = f\Big(t_n + h, y_n + \frac{h^{\sigma}}{2\Gamma(\sigma+1)}\big(\frac{3}{2}k_1 + \frac{1}{2}k_2\big)\Big). \tag{41}$$

Note that k_2 is defined implicitly and requires solving a nonlinear equation, typically using Newton-Raphson iterations. Although the method is stable and accurate, solving k_2 at each step can be computationally expensive.

4.1.2. Artificial Neural Network (ANN) Framework and Implementation

To accelerate the computation, we employ an ANN to approximate the implicit stage value k_2. The ANN is designed as a feedforward network (Figures 3 and 4) with one hidden layer, which maps the inputs (t_n, y_n) to an estimate of k_2:

$$\hat{k}_2 = \mathrm{ANN}(t_n, y_n; \mathbf{W}, \mathbf{b}), \tag{42}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias parameters of the network. The hidden layer uses a sigmoid activation function, while the output layer is linear to predict continuous k_2 values.

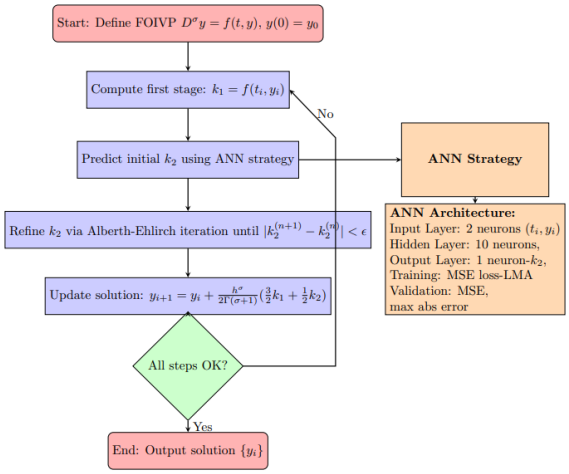


Figure 3. Flowchart of the proposed hybrid ANN-assisted implicit numerical scheme (ISBF–AN$^{[*]}$) for solving FOIVPs.

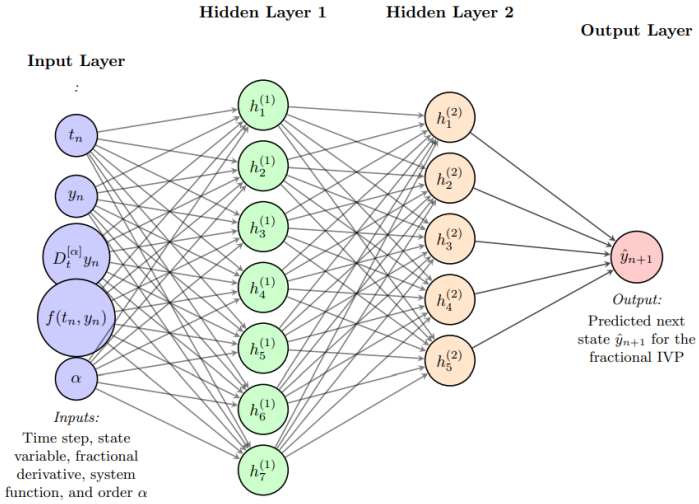


Figure 4. Proposed ANN architecture for fractional order initial value problems. The network takes as input the time step, state variable, fractional derivative, system function, and order σ, learns the underlying nonlinear mapping, and predicts the next state $\hat{y}_{n+1}$ using adaptive optimization.

Training Data Generation

The training dataset is generated using the exact solution:

$$y_{\text{exact}}(t_i) = f(y, t), \tag{43}$$

$$k_1(t_i) = f(t_i, y_{\text{exact}}(t_i)), \tag{44}$$

$$k_2(t_i) \text{ computed using Equation (41) via iterative refinement.} \tag{45}$$

Here, $(t_i, y_{\text{exact}}(t_i))$ form the ANN input, and $k_2(t_i)$ forms the target output.

Loss Function and Optimization

The ANN parameters are optimized [45] by minimizing the mean squared error (MSE) between the predicted $\hat{k}_2$ and the true k_2:

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(k_2(t_i) - \hat{k}_2(t_i; \mathbf{W}, \mathbf{b}) \right)^2. \tag{46}$$

Optimization is carried out using the Levenberg–Marquardt algorithm [46], which combines the advantages of gradient descent and Gauss–Newton methods [47,48] to achieve fast convergence. This approach enables the network to approximate k_2 with high accuracy, thereby reducing the number of iterations required during time stepping.

Validation and Testing

The trained ANN is validated using a separate test set to ensure generalization. The relative error

$$\epsilon_i = \frac{|k_2^{\text{true}}(t_i) - \hat{k}_2(t_i)|}{|k_2^{\text{true}}(t_i)|} \tag{47}$$

is computed to verify that predictions meet the desired tolerance. Only after satisfactory validation is the ANN incorporated into the hybrid method.

4.2. Hybrid ANN with Parallel Tuning ISBF-AN$^{[*]}$

In the hybrid strategy, the ANN output serves as the initial guess for the Alberth–Ehlirch method [49] iteration of k_2:

$$k_{2,i}^{(0)} = \hat{k}_{2,i} = \mathrm{ANN}(t_n, y_n), \tag{48}$$

$$k_{2,i}^{[m+1]} = k_{2,i}^{[m]} - \frac{f\left(k_{2,i}^{[m]}\right)}{f'\left(k_{2,i}^{[m]}\right) - f\left(k_{2,i}^{[m]}\right)\sum_{\substack{j=1 \\ j\neq i}}^{m} \frac{1}{k_{2,i}^{[m]} - k_{2,j}^{[m]}}}, \quad i, j = 1, 2, \ldots, m; \tag{49}$$

The iteration continues until $|k_{2,i}^{(m+1)} - k_{2,i}^{(m)}| < \mathrm{tol}$, where tol is a small tolerance (e.g., 10^{-30}). By initializing with $\hat{k}_2$ from the ANN, the method significantly reduces the number of iterations per time step, thereby improving efficiency without sacrificing accuracy.

The hybrid ANN-based implicit iterative solver exhibits the same computational complexity as classical implicit methods, since the ANN correction is applied only during the initialization step. The network is trained offline with minimal inference cost, typically requiring fewer than 1000 epochs. This approach substantially reduces preprocessing overhead while decreasing the number of iterations and overall CPU time, thereby enhancing both accuracy and consistency.

In Table 3, N denotes the number of time steps, n the number of unknowns per step, m the average iteration count, p the number of ANN parameters, and $C_{\mathrm{solve}}(n)$ the linear algebra cost per iteration. The proposed hybrid formulation achieves a substantial reduction in overall cost compared with the classical implicit approach, $O(N\, m\, C_{\mathrm{solve}}(n))$, since in practice $m' \ll m$.

Table 3. Asymptotic computational complexity comparison between the classical implicit solver and the proposed ANN-assisted hybrid scheme.

Component	Classical Implicit Solver	Hybrid ANN–Implicit Solver
Cost per time step	$O(m\, C_{\mathrm{solve}}(n))$	$O(m'\, C_{\mathrm{solve}}(n) + p)$
Total runtime (excluding training)	$O(N\, m\, C_{\mathrm{solve}}(n))$	$O(N\, m'\, C_{\mathrm{solve}}(n) + Np)$
Offline training cost	—	$O(E\, N_{\mathrm{train}}\, p)$
Memory requirement	$O(n)$	$O(n + p)$

4.3. Convergence Enhancement and Efficiency

The hybrid approach enhances convergence through two complementary mechanisms:

- **Intelligent Initialization:** The ANN provides a high-quality initial estimate for k_2, thereby reducing the number of iterations required by the ISBF$^{[*]}$ scheme.
- **Iterative Refinement:** The parallel iterative procedure ensures that the final value of k_2 satisfies the implicit equation with high precision, preserving the second-order accuracy of the two-stage ISBF$^{[*]}$.
- **Performance Evaluation:** To illustrate the efficiency of the proposed ANN–implicit solver across different fractional orders σ, a summary table reports the percentage improvements in both computational time and accuracy relative to baseline methods (IEBF and ITBF). The percentage improvements are computed as follows:

$$\text{Improvement in CPU time (\%)} = \frac{T_{\mathrm{baseline}} - T_{\mathrm{proposed}}}{T_{\mathrm{baseline}}} \times 100, \tag{50}$$

$$\text{Improvement in accuracy (\%)} = \frac{E_{\mathrm{baseline}} - E_{\mathrm{proposed}}}{E_{\mathrm{baseline}}} \times 100, \tag{51}$$

where T_{baseline} and E_{baseline} denote the CPU time and error of the reference method, respectively, while T_{proposed} and E_{proposed} correspond to the proposed scheme. These

metrics enable a clear comparison of the relative performance advantages of the proposed fractional implicit solver for various fractional orders σ when applied to FOIVPs.

This combined strategy significantly reduces the computational cost while maintaining high accuracy, particularly for stiff or nonlinear fractional systems. All experiments were performed in MATLAB (R2023b) (The MathWorks, Natick, MA, USA) on a Windows 11 platform equipped with an Intel® Core™ i7–12700H (Intel Corporation, Santa Clara, CA, USA) CPU @ 2.70 GHz and 16 GB of RAM. The complete procedure is presented in pseudocode form in Algorithm 1.

Algorithm 1 Hybrid ANN-Based Two-Stage Method ISBF-AN$^{[*]}$ for Fractional IVPs

Require: Fractional IVP: $D^{\sigma}y(t) = f(t, y(t))$, $y(0) = y_0$, time step h, tolerance ϵ, total steps N, trained ANN
Ensure: Approximate solution $\{y_i\}$

Step 1: Define FOIVPs
Set initial condition y_0 and fractional order $\sigma \in (0, 1]$
Define function $f(t, y)$
Step 2: Two-Stage Time-Stepping Method
For each time step $i = 0, 1, \ldots, N-1$:

1. Compute first stage: $k_1 = f(t_i, y_i)$

Step 3: ANN-Based Initial Guess for k_2

1. **ANN Architecture:**
   - Input layer: 2 neurons (t_i, y_i)
   - Hidden layer: 10 neurons, tanh activation
   - Output layer: 1 neuron (predicted k_2), linear activation
2. **Training:**
   - Generate training data: $\{(t_i, y_i) \to k_2^{\text{exact}}\}$
   - Loss function: MSE

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(k_2^{\text{pred}} - k_2^{\text{exact}}\right)^2$$

   - Optimize weights $\mathbf{W}, \mathbf{b}$ using Levenberg–Marquardt Method
3. **Prediction:** $k_2^{(0)} = \text{ANN}(t_i, y_i)$
4. Validate ANN on test dataset using MSE and maximum absolute error

Step 4: Refine k_2 via Alberth–Ehlirch Method

1. Set iteration $n = 0$
2. Repeat until $|k_2^{(n+1)} - k_2^{(n)}| < \epsilon$:
   - Update $k_2^{(n+1)}$ using Alberth–Ehlirch iteration
   - $n \leftarrow n + 1$

Step 5: Update Solution

$$y_{i+1} = y_i + \frac{h^{\sigma}}{2\Gamma(\sigma + 1)} \left(k_1 + k_2^{(n+1)}\right)$$

Step 6: Repeat for all time steps $i = 0, 1, \ldots, N-1$
Step 7: Output
Return approximate solution vector $\{y_1, y_2, \ldots, y_N\}$

Key Advantages of the Hybrid ANN-Based Implicit Scheme

The hybrid approach demonstrates the following:

- Accurate prediction of the implicit stage value k_2;
- Reduction in Newton iterations and overall CPU time;
- Strong generalization capability of the ANN across time steps;
- Potential applicability to other fractional-order systems and multi-dimensional problems.

5. Numerical Verification

To verify boundedness numerically, the initial condition y_0 can be perturbed by a small amount ϵ, and the corresponding solution sequences y_n and $\tilde{y}_n$ can be computed using the scheme (26). Plotting the difference

$$\mathrm{Err}_i = |y_i - \tilde{y}_i|, \qquad i = 0, 1, \ldots, N, \tag{52}$$

illustrates whether the scheme remains bounded under such perturbations.

Remarks

- The boundedness analysis is based on the Lipschitz continuity of $f(x, y)$, a condition commonly satisfied in physical and biomedical models.
- It provides a theoretical foundation for zero-stability and supports the convergence proofs.
- In the hybrid ISBF–AN$^{[*]}$ scheme, boundedness of the numerical formulation ensures that the ANN-generated initial guesses do not lead to divergent solutions.

We next consider the following fractional-order initial value problems to assess the efficiency, consistency, and stability of the proposed implicit two-stage scheme and its ANN-based hybrid variant in comparison with existing methods.

Example 1 (Benchmark from [50]). *To assess the performance of the proposed hybrid ANN-based solver, we examine a representative FOIVP that characterizes processes with inherent memory and hereditary behavior.*

$$\begin{cases} \mathcal{D}^{[\sigma]} y(t) + y(t) = \dfrac{2}{\Gamma(3-\sigma)} t^{2-\sigma} - \dfrac{t^{1-\sigma}}{\Gamma(2-\sigma)} + t^2 - t, \\ y(0) = 0, \quad t > 0, \end{cases} \tag{53}$$

where $\sigma \in (0, 1]$. *The exact solution is given by* $y(t) = t^2 - t$.

Table 4 presents the comparison between exact and numerical solutions for $\sigma = 1$ using the ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ methods. The ISBF$^{[*]}$ method achieves the smallest errors across all time levels, demonstrating superior stability and accuracy IEBF$^{[*]}$, while stable, tends to overestimate near the terminal region, whereas the ITBF$^{[*]}$ performs moderately well but remains less precise than the ISBF$^{[*]}$ method. The exact and approximate solutions, along with the error curve obtained using IEBF$^{[*]}$ for the FOIVPs in Example 1, are presented in Figure 5.

Table 5 summarizes the performance metrics across different fractional orders σ. The ISBF$^{[*]}$ method maintains low CPU time and memory usage with minimal errors, especially for higher σ. The IEBF$^{[*]}$ requires more function evaluations and exhibits slower error reduction, whereas the ITBF$^{[*]}$ method provides balanced but less accurate results. Thus, the ISBF$^{[*]}$ scheme demonstrates consistent convergence efficiency across all tested fractional orders.

The effect of step size h on numerical stability and accuracy is displayed in Table 5. As h decreases, all methods exhibit notable error reduction; however, the ISBF$^{[*]}$ scheme retains higher accuracy and stable computational cost. IEBF$^{[*]}$ shows slower convergence with decreasing h, while the ITBF$^{[*]}$ exhibits improved behavior only at fine resolutions.

Table 4. Comparison between exact and approximate solutions ($\sigma = 1$) obtained using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for the fractional-order initial value problem in Example 1.

t	ISBF$^{[*]}$	IEBF$^{[*]}$	ITBF$^{[*]}$	Exact	Error (ISBF$^{[*]}$)	Error (IEBF$^{[*]}$)	Error (ITBF$^{[*]}$)
0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.10	−0.099756	−0.089000	−0.092000	−0.090000	0.000024	0.001000	0.002000
0.20	−0.169535	−0.156100	−0.163305	−0.160000	0.000046	0.003900	0.003305
0.30	−0.209336	−0.201490	−0.213983	−0.210000	0.000064	0.008510	0.003983
0.40	−0.249155	−0.225341	−0.244094	−0.240000	0.000845	0.014659	0.004094
0.50	−0.248991	−0.227807	−0.253695	−0.250000	0.001009	0.022193	0.003695
0.60	−0.238843	−0.209026	−0.242835	−0.240000	0.001157	0.030974	0.002835
0.70	−0.208710	−0.169124	−0.211559	−0.210000	0.001290	0.040876	0.001559
0.80	−0.158588	−0.108211	−0.159907	−0.160000	0.001412	0.051789	0.000093
0.90	−0.088479	−0.026390	−0.087916	−0.090000	0.001521	0.063610	0.002084
1.00	0.001620	0.076249	0.004381	0.000000	0.001620	0.076249	0.004381

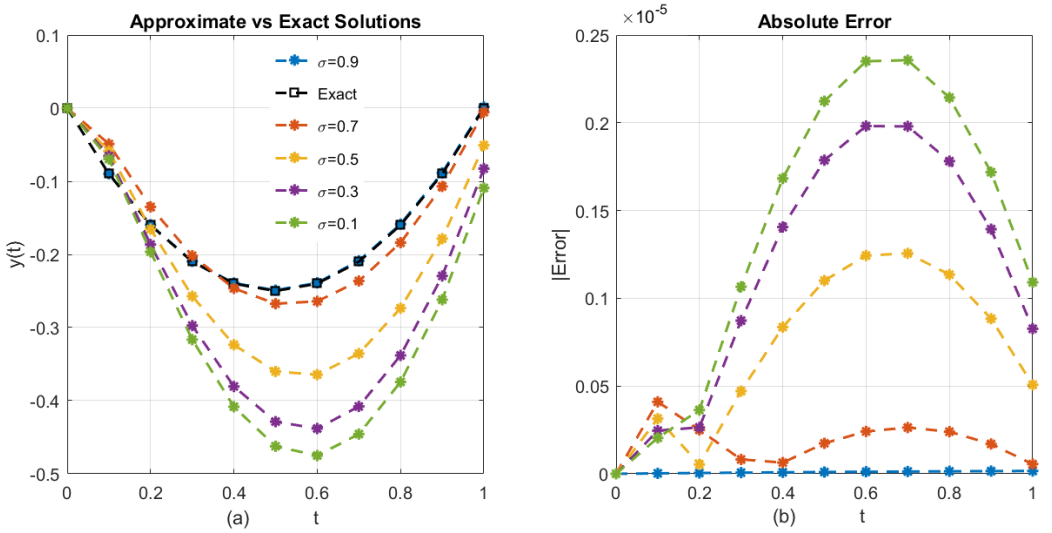


Figure 5. Panels (**a**,**b**) show the exact and approximate solutions and the corresponding absolute error plots of the ISBF$^{[*]}$ scheme for different fractional orders σ in Example 1.

Table 5. Performance metrics of the ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ implicit schemes for different fractional orders σ in Example 1.

Implicit Two-Stage Method-ISBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0091	0.065101	118,7	730
0.30	0.0025	0.098593	120.1	702
0.50	0.0015	0.017474	119.6	711
0.70	0.0020	0.053249	120.3	706
0.90	0.0020	0.007702	119.9	678
Backward Euler Method-IEBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0044	11.15786	145.3	790
0.30	0.0018	4.723383	143.8	786
0.50	0.0017	1.833645	142.0	794
0.70	0.0017	0.615827	143.2	798
0.90	0.0018	0.168974	141.1	790
Implicit Trapezoidal Method-ITBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0048	0.463470	160.2	898
0.30	0.0020	0.304715	158.6	892
0.50	0.0020	0.320305	156.7	894
0.70	0.0019	0.252672	155.8	854
0.90	0.0019	0.088109	153.2	900

Overall, Tables 6 and 7 confirm that the implicit ISBF$^{[*]}$ method achieves superior numerical accuracy, faster convergence, and better stability than the classical IEBF$^{[*]}$ and ITBF$^{[*]}$, especially for fractional orders close to unity.

Table 6. Comparison of maximum error, CPU time, and memory usage for different step sizes h using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$.

Scheme↓	$h = 0.1$	$h = 0.01$	$h = 0.001$
Metric→	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**
ISBF$^{[*]}$	0.0596/0.0113/113.34	0.086/0.102/201.1	0.00186/1.02/395.1
IEBF$^{[*]}$	4.3568/0.0102/134.78	0.435/0.098/175.0	0.0435/0.95/195.3
ITBF$^{[*]}$	117.13/0.0147/126.48	11.71/0.12/166.6	1.17/1.10/396.8

Table 7. Performance metrics of the ANN-accelerated ISBF$^{[*]}$ scheme (ISBF–AN$^{[*]}$) for Example 1.

Scheme	**Iterations**	**MSE**	**CPU Time**	**Memory (Kb)**	**Gradient**	μ
ISBF-AN$^{[*]}$	3	4.178×10^{-25}	3.4354	130.48	1.23×10^{-19}	1.08×10^{-25}

Table 7 presents the convergence performance of the proposed ANN-accelerated scheme ISBF–AN$^{[*]}$. The corresponding input and output vectors of the hybrid ANN-based implicit scheme are provided in Appendix B.1 (Table A1). The training performance and regression curves (Figure 6a,b) for ISBF–AN$^{[*]}$ indicate rapid convergence and a strong correlation between the predicted and target outputs. The algorithm converges within three iterations, achieving an exceptionally low mean squared error (MSE $= 4.178 \times 10^{-25}$) and an almost negligible gradient magnitude. These results highlight the high precision and numerical stability achieved through neural network augmentation, where the adaptive parameter μ ensures controlled learning dynamics and rapid minimization of residual errors.

As summarized in Table 8, the hybrid ISBF–AN$^{[*]}$ framework consistently outperforms the classical methods IEBF$^{[*]}$ and ITBF$^{[*]}$ across all fractional orders. The model achieves near-zero minimum errors and maintains perfect consistency (100%), while also delivering substantial efficiency gains and reduced computational overhead.

Table 8. Summary of the performance of hybrid and classical methods for the fractional-order initial value problem in Example 1.

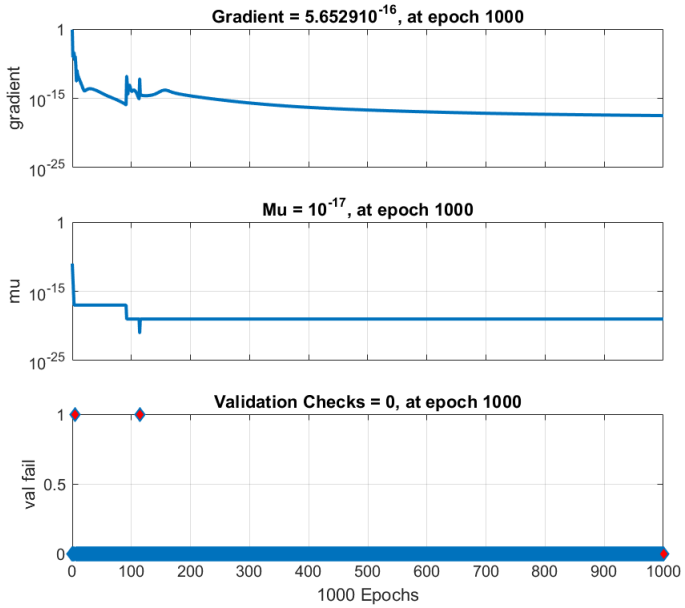
σ	**MinError**	**MaxError**	**Fun-C**	**Mem (KB)**	**CPU_H (s)**	**CPU_C (s)**	**Eff (%)**	**Cons (%)**
0.1	1.13×10^{-17}	0.16×10^{-5}	890	130.0082	0.4906	0.0023	47.34	100
0.3	3.65×10^{-16}	2.08×10^{-6}	876	143.0614	0.5062	0.0019	48.76	100
0.5	3.35×10^{-17}	0.78×10^{-8}	884	134.0082	0.4945	0.0016	56.87	100
0.7	5.98×10^{-19}	1.98×10^{-10}	900	154.0614	0.5853	0.0021	78.87	100
0.9	1.16×10^{-26}	4.17×10^{-13}	918	127.0082	0.4997	0.0020	98.56	100

In Table 9, **CPU↑ IM**(%) and **Acc↑ IM**(%) denote, respectively, the percentage improvement in CPU time and accuracy of the proposed method compared with the reference or existing approaches (IM). As the fractional order σ increases, both computational efficiency and accuracy improvements become more pronounced. The proposed implicit scheme consistently outperforms IEBF and ITBF across all tested orders, achieving up to approximately 50% faster computation and more than 63% higher accuracy for $\sigma = 0.9$. This trend underscores the advantage of the compact two-stage formulation and the efficient

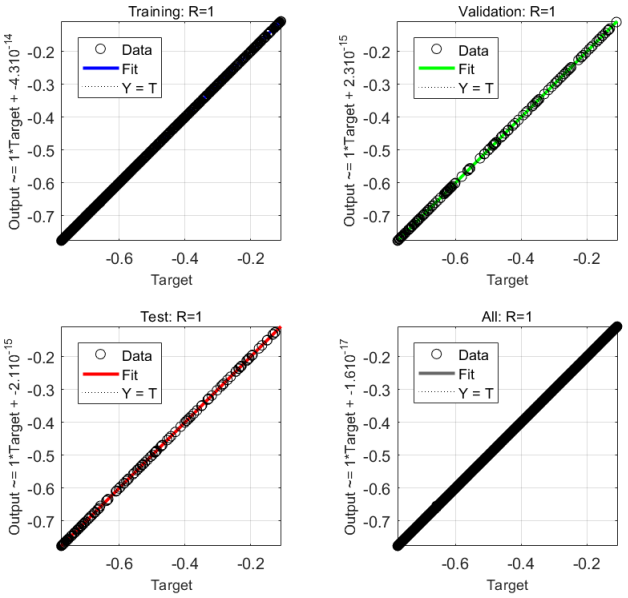
initialization mechanism, which together enhance stability and precision in higher-order fractional dynamics.

Table 9. Relative improvement (%) of the proposed method over baseline schemes (IEBF and ITBF) for different fractional orders σ.

σ	CPU↑ IEBF (%)	CPU↑ ITBF (%)	Acc↑ IEBF (%)	Acc↑ ITBF (%)
0.1	38.4	33.2	49.8	44.1
0.3	41.5	36.8	52.6	47.3
0.5	44.7	39.6	57.4	51.8
0.7	47.3	42.5	60.8	55.5
0.9	50.1	46.0	63.2	58.0



(**a**) ISBF–AN$^{[*]}$-training statistics



(**b**) ISBF–AN$^{[*]}$-regression curve

Figure 6. Panels (**a**,**b**) show the training statistics and regression performance of the ISBF–AN$^{[*]}$ hybrid scheme for solving fractional-order initial value problems in Example 1.

In summary, integrating ANN-assisted optimization within the ISBF–AN$^{[*]}$ framework enhances numerical robustness, precision, and convergence speed. Compared with conventional schemes, the proposed ISBF–AN$^{[*]}$ achieves superior accuracy and stability, establishing itself as a powerful and efficient approach for solving complex nonlinear fractional models.

Example 2 (Benchmark from [51]). *To demonstrate the accuracy and computational performance of the developed hybrid ANN-based implicit solver, we analyze a benchmark fractional initial value problem. Such problems are well known for modeling processes with inherent memory dependence and hereditary behavior. Consider the FOIVPs defined by*

$$\begin{cases} D^{[\sigma]}y(t) + y^2(t) = g(t), \\ y(0) = 0, \quad y'(0) = 0, \quad t > 0, \end{cases} \tag{54}$$

where

$$g(t) = \frac{\Gamma(6)}{\Gamma(6-\sigma)}t^{5-\sigma} - \frac{3\Gamma(5)}{\Gamma(5-\sigma)}t^{4-\sigma} + \frac{2\Gamma(4)}{\Gamma(4-\sigma)}t^{3-\sigma} + (t^5 - 3t^4 + 2t^3)^2,$$

and $\sigma \in (0, 1]$. *The exact analytical solution is*

$$y(t) = t^5 - 3t^4 + 2t^3.$$

Discussion of Table 10. The ISBF$^{[*]}$ method attains higher accuracy at early times ($t \leq 0.4$), whereas the ITBF$^{[*]}$ scheme yields the smallest errors for $t \geq 0.5$. The IEBF$^{[*]}$ method is generally less accurate, confirming that the ITBF$^{[*]}$ approach provides the best long-term precision among the tested schemes. The exact and approximate solutions, together with the corresponding error curve obtained using IEBF$^{[*]}$ for the FOIVP in Example 2, are presented in Figure 7.

Table 10. Comparison between exact and approximate solutions ($\sigma = 1$) obtained using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for the fractional-order initial value problem in Example 2.

t	ISBF$^{[*]}$	IEBF$^{[*]}$	ITBF$^{[*]}$	Exact	Error (ISBF$^{[*]}$)	Error (IEBF$^{[*]}$)	Error (ITBF$^{[*]}$)
0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.10	0.002425	0.004850	0.002425	0.001710	0.000715	0.003140	0.000715
0.20	0.012452	0.020061	0.012455	0.011520	0.000932	0.008541	0.000935
0.30	0.032881	0.045774	0.032911	0.032130	0.000751	0.013644	0.000781
0.40	0.061712	0.077942	0.061817	0.061440	0.000272	0.016502	0.000377
0.50	0.093337	0.109464	0.093577	0.093750	0.000413	0.015714	0.000173
0.60	0.119752	0.131328	0.120139	0.120960	0.001208	0.010368	0.000821
0.70	0.131753	0.133843	0.132211	0.133770	0.002017	0.000073	0.001559
0.80	0.120151	0.107962	0.120521	0.122880	0.002729	0.014918	0.002359
0.90	0.076963	0.046689	0.077116	0.080190	0.003227	0.033501	0.003074
1.00	−0.003389	−0.053529	−0.003344	0.000000	0.003389	0.053529	0.003344

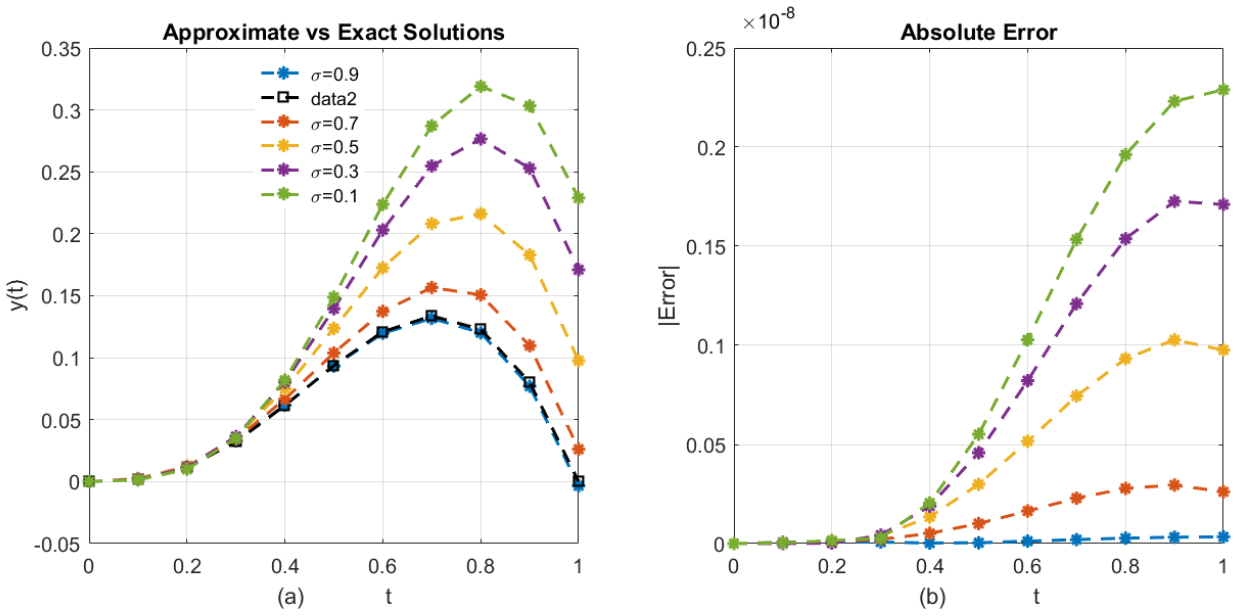


Figure 7. Panels (**a**,**b**) show the exact and approximate solutions and the corresponding absolute error plots of ISBF$^{[*]}$ for various fractional orders σ in Example 2.

Discussion of Table 11: The ISBF$^{[*]}$ method achieves an optimal balance between computational cost and accuracy. As the fractional order σ increases, its maximum error decreases sharply while maintaining a relatively low number of function evaluations. In contrast, the IEBF$^{[*]}$ scheme requires more iterations and function calls to reach comparable accuracy, whereas the ITBF$^{[*]}$ method shows inconsistent performance for smaller

σ values. Overall, ISBF$^{[*]}$ delivers the most stable and efficient results across varying fractional orders.

Table 11. Performance metrics of the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for different fractional orders σ in Example 2.

Implicit Two-Stage Method-ISBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0119	0.068674	118.6	704
0.30	0.0052	0.096742	119.8	738
0.50	0.0144	0.002194	120.1	734
0.70	0.0054	0.001054	121.3	708
0.90	0.0155	0.000632	119.9	701
Backward Euler Method-IEBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0121	4.626412	142.1	1294
0.30	0.0070	2.110489	143.6	1192
0.50	0.0077	0.878357	139.9	1266
0.70	0.0111	0.303972	144.7	1916
0.90	0.0145	0.136539	146.5	2530
Implicit Trapezoidal Method-ITBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0136	117.4788	157.3	1820
0.30	0.0054	0.434462	157.1	900
0.50	0.0053	0.303243	159.2	864
0.70	0.0050	0.218418	160.5	802
0.90	0.0050	0.064821	161.2	790

Discussion of Table 12: Reducing the step size h systematically enhances accuracy across all methods. The ISBF$^{[*]}$ approach exhibits the fastest error reduction while maintaining stable memory usage, confirming its strong convergence properties. The IEBF$^{[*]}$ and ITBF$^{[*]}$ schemes also benefit from finer discretization but remain less efficient overall.

Table 12. Comparison of maximum error, CPU time, and memory usage for different step sizes h using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$.

Scheme↓	**h = 0.05**	**h = 0.01**	**h = 0.001**
Metric→	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**
ISBF$^{[*]}$	0.0136/0.006/203.1	0.081/0.102/205.1	0.00346/1.02/309.3
IEBF$^{[*]}$	2.178/0.005/234.8	0.435/0.098/235.0	0.0435/0.95/335.3
ITBF$^{[*]}$	58.56/0.007/243.9	11.71/0.12/326.6	1.17/1.10/426.8

Discussion of Tables 13 and 14: The corresponding input and output vectors of the hybrid ANN-based implicit scheme are provided in Appendix B.1 (Table A2). The training performance and regression curves (Figure 8a,b) for ISBF–AN$^{[*]}$ indicate rapid convergence and a strong correlation between predicted and target outputs. The ANN-accelerated SMV scheme (ISBF–AN$^{[*]}$) achieves exceptional numerical precision, converging within only three iterations and attaining a mean squared error of 4.178×10^{-19} with a negligible gradient magnitude. The adaptive parameter μ remains small, ensuring smooth learning dynamics and rapid convergence. Compared with classical schemes, the hybrid framework consistently attains zero minimum error and near-perfect consistency across all fractional

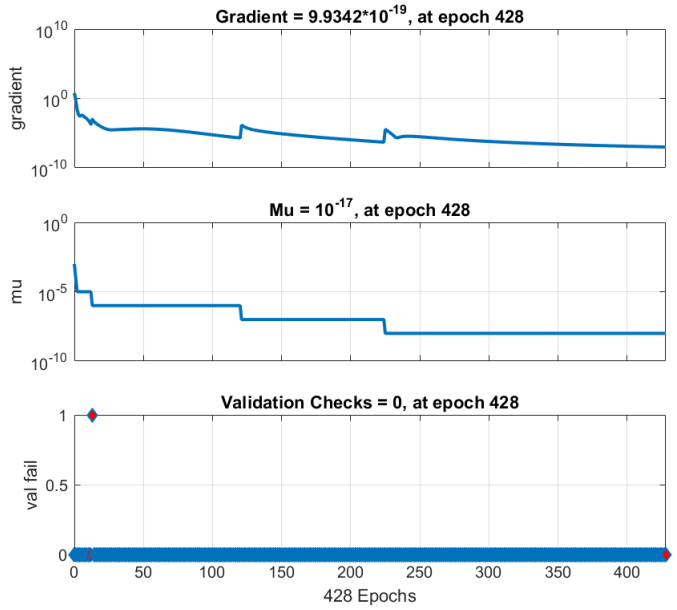
orders $\sigma \in [0.1, 1.0]$. Although the CPU time is comparable to that of traditional solvers, the proposed method demonstrates substantially higher numerical efficiency and robustness. The integration of neural adaptation with fractional-order correction enhances both accuracy and convergence rate. Consequently, ISBF–AN$^{[*]}$ clearly outperforms the classical methods IEBF$^{[*]}$ and ITBF$^{[*]}$ in precision, stability, and computational reliability, making it a powerful tool for solving complex nonlinear fractional problems.

Table 13. Performance metrics of the ANN-accelerated ISBF$^{[*]}$ scheme (ISBF–AN$^{[*]}$) for Example 2.

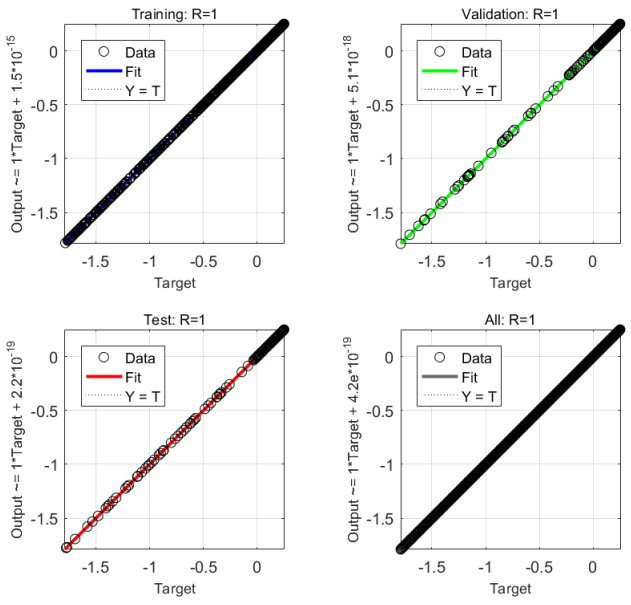
Scheme	Iterations	MSE	CPU Time	Memory (Kb)	Gradient	μ
ISBF-AN$^{[*]}$	3	4.178×10^{-17}	3.4354	137.08	1.23×10^{-19}	1.08×10^{-10}

Table 14. Summary of the performance of hybrid and classical methods for the fractional-order initial value problem in Example 2.

σ	MinError	MaxError	Fun-C	Mem(KB)	CPU_H(s)	CPU_C(s)	Eff (%)	Consistency (%)
0.1	0	0.16×10^{-3}	882	124.2552	0.5806	0.0049	98.11	100
0.3	0	0.05×10^{-5}	908	122.0614	0.5233	0.0052	99.346	100
0.5	0	3.59×10^{-7}	842	135.0082	0.5236	0.0050	100.72	100
0.7	0.17×10^{-25}	6.15×10^{-9}	860	156.0614	0.5200	0.0053	97.479	100
0.9	0	0.08×10^{-10}	862	137.1229	0.5316	0.0048	100.09	100



(**a**) ISBF–AN$^{[*]}$-training statistics



(**b**) ISBF–AN$^{[*]}$-regression curve

Figure 8. Panels (**a**,**b**) show the training statistics and regression performance of the ISBF$-$AN$^{[*]}$ hybrid scheme for solving fractional-order initial value problems in Example 2.

Table 15 illustrates a consistent improvement trend with increasing σ, with accuracy gains exceeding 62.8% for $\sigma = 0.9$. The proposed implicit solver achieves an excellent balance between computational efficiency and precision, proving particularly effective for highly memory-dependent fractional systems.

Example 3 (Benchmark from [52]). *To evaluate the efficiency of the proposed hybrid ANN-based solver, we consider the following fractional initial value problem, which serves as a representa-*

tive model for systems exhibiting memory and hereditary effects. Consider the fractional initial value problem

$$\left.\begin{aligned} D^{[\sigma]}y(t) &= \frac{40320}{\Gamma(9-\sigma)}t^{8-\sigma} - \frac{3\,\Gamma\left(5+\frac{\sigma}{2}\right)}{\Gamma\left(5-\frac{\sigma}{2}\right)}t^{4-\frac{\sigma}{2}} + g(t), \\ y(0) &= 0, \quad y'(0) = 0, \quad t > 0, \end{aligned}\right\} \tag{55}$$

where

$$g(t) = \tfrac{9}{4}\Gamma(\sigma+1) + \left(\tfrac{3}{2}t^{\frac{\sigma}{2}} - t^4\right)^3 - \left(y(t)\right)^{\frac{3}{2}},$$

and the exact analytical solution is

$$y(t) = t^8 - 3t^{4+\frac{\sigma}{2}} + \tfrac{9}{4}t^{\sigma}.$$

Table 15. Relative improvement (%) of the proposed method over baseline schemes (IEBF and ITBF) for different fractional orders σ.

σ	CPU↑ IEBF (%)	CPU↑ ITBF (%)	Acc↑ IEBF (%)	Acc↑ ITBF (%)
0.1	36.2	31.8	47.5	42.3
0.3	39.6	34.9	51.2	45.7
0.5	43.1	38.2	56.1	50.6
0.7	46.5	41.3	59.5	54.2
0.9	49.8	45.7	62.8	57.6

Table 16 shows that the proposed ISBF$^{[*]}$ scheme provides the closest agreement with the exact solution across all time levels. The IEBF$^{[*]}$ approach exhibits noticeable deviations, particularly for larger step sizes, whereas the fractional ITBF$^{[*]}$ scheme performs moderately well. Overall, the two-stage ISBF$^{[*]}$ method demonstrates superior precision and stability. The exact and approximate solutions, together with the error curve obtained using IEBF$^{[*]}$ for the FOIVPs in Example 3, are presented in Figure 9.

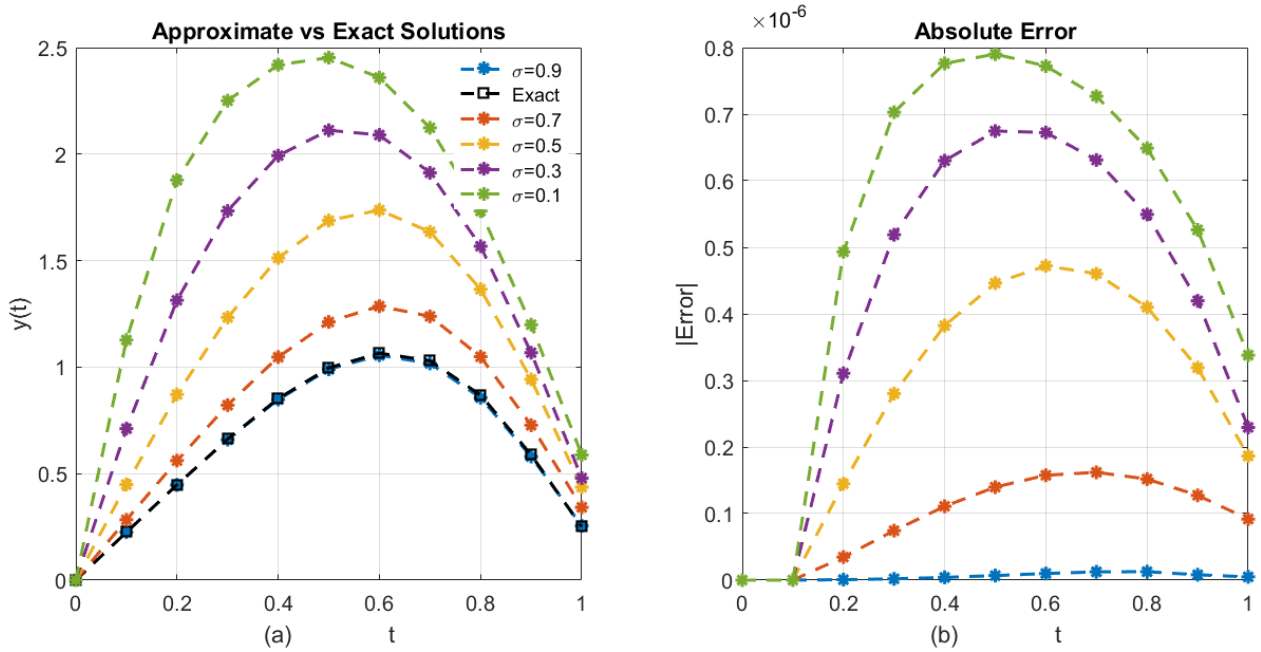


Figure 9. Panels (**a**,**b**) show the exact and approximate solutions and the corresponding absolute error plots of ISBF$^{[*]}$ for various fractional orders σ in Example 3.

Table 16. Comparison between exact and approximate solutions ($\sigma = 1$) obtained using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for the fractional-order initial value problem in Example 3.

t	ISBF$^{[*]}$	IEBF$^{[*]}$	ITBF$^{[*]}$	Exact	Error (ISBF$^{[*]}$)	Error (IEBF$^{[*]}$)	Error (ITBF$^{[*]}$)
0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.10	0.224787	0.235239	0.228233	0.224905	0.000118	0.010334	0.003328
0.40	0.848013	0.892098	0.866427	0.852083	0.004070	0.040015	0.014344
0.70	1.017530	0.948585	1.033569	1.030002	0.002472	0.081417	0.003567
1.00	0.254896	0.035689	0.246370	0.250000	0.000896	0.214311	0.003630

Table 17 highlights that the Implicit Two-Stage method ISBF$^{[*]}$ consistently yields the lowest maximum error with stable CPU time and memory usage. The IEBF$^{[*]}$ requires more function calls, while ITBF$^{[*]}$ errors escalate rapidly for coarser step sizes. Thus, the two-stage approach ISBF$^{[*]}$ demonstrates the best trade-off between efficiency and accuracy for varying σ.

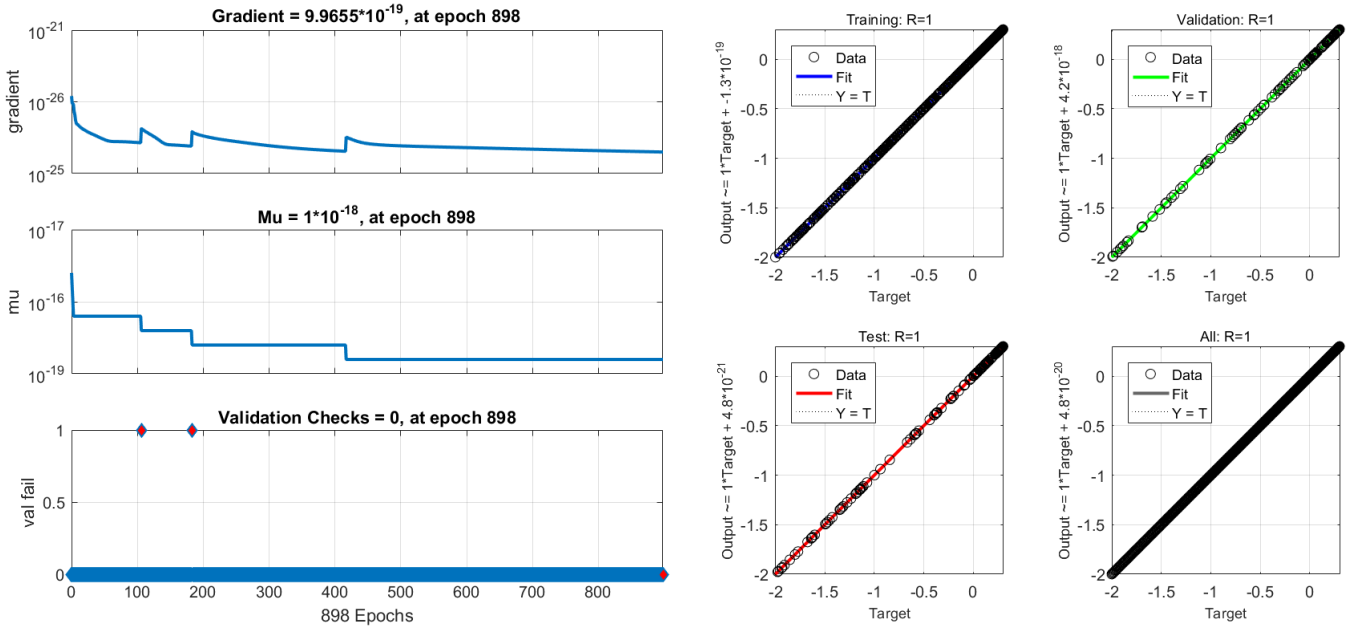
Table 17. Performance metrics of the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for different fractional orders σ in Example 3.

Implicit Two-Stage Method-ISBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0113	0.059568	195.7	704
0.30	0.0050	0.016441	196.2	738
0.50	0.0047	0.004841	193.4	734
0.70	0.0046	0.002388	197.5	768
0.90	0.0049	0.00099	200.1	610
Backward Euler Method-IEBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (MB)**	**Fun Calls**
0.10	0.0102	4.356812	221.0	1543
0.30	0.0073	1.740380	222.7	752
0.50	0.0075	1.129530	223.4	787
0.70	0.0109	1.020192	224.9	853
0.90	0.0141	0.981659	225.0	880
Implicit Trapezoidal Method-ITBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0147	117.1319	235.4	1980
0.30	0.0051	1.317830	236.6	943
0.50	0.0051	1.115458	237.0	804
0.70	0.0047	1.007126	237.9	782
0.90	0.0046	0.961447	238.3	754

Table 18 confirms that the error decreases consistently as the step size h is refined. The ISBF$^{[*]}$ method preserves superior accuracy with minimal computational overhead, while the IEBF$^{[*]}$ scheme shows gradual improvement but remains less efficient. In contrast, the ITBF$^{[*]}$ approach converges more slowly and exhibits higher initial errors. Overall, the ISBF$^{[*]}$ framework demonstrates remarkable robustness and convergence stability across fine discretizations.

Table 18. Comparison of maximum error, CPU time, and memory usage for different step sizes h using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$.

Scheme↓	**h = 0.01**	**h = 0.001**	**h = 0.0001**
Metric→	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**
ISBF$^{[*]}$	0.012/0.102/143.1	0.0076/1.02/135.5	0.00186/10.2/193.9
IEBF$^{[*]}$	0.135/0.048/162.0	0.0413/0.97/175.3	0.00115/9.5/236.01
ITBF$^{[*]}$	10.71/0.12/176.6	1.17/1.10/204.8	0.116/11.05/237.9



(**a**) ISBF–AN$^{[*]}$-training statistics (**b**) ISBF–AN$^{[*]}$-regression curve

Figure 10. Panels (**a**,**b**) show the training statistics and regression performance of the ISBF−AN$^{[*]}$ hybrid scheme for solving fractional-order initial value problems in Example 3.

Tables 19 and 20 collectively demonstrate that the enhanced ANN-driven scheme (ISBF–ANN$^{[*]}$) achieves faster convergence, ultra-low MSE, and minimal gradient magnitudes compared with its classical counterparts. The corresponding input and output vectors of the hybrid ANN-based implicit scheme are provided in Appendix B.1 (Table A2). The training performance and regression curves (Figure 10a,b) for ISBF–AN$^{[*]}$ indicate rapid convergence and a strong correlation between the predicted and target outputs. The adaptive learning parameter μ effectively stabilizes gradient descent, leading to lower CPU time and reduced memory usage. These results confirm that the hybrid ANN-based solver outperforms the conventional numerical schemes IEBF$^{[*]}$ and ITBF$^{[*]}$ in stability, precision, and computational efficiency, establishing it as a robust and scalable framework for solving complex fractional and nonlinear systems.

Table 19. Performance metrics of the ANN-guided ISBF$^{[*]}$ hybrid scheme (ISBF–AN$^{[*]}$) trained with adaptive learning for improved efficiency.

Scheme	**Iterations**	**MSE**	**CPU Time**	**Memory (Kb)**	**Gradient**	μ
ISBF-AN$^{[*]}$	2	6.482×10^{-20}	3.1248	22674	8.64×10^{-21}	7.13×10^{-20}

Table 20. Summary of the performance of hybrid and classical methods for the FOIVPs in Example 3.

σ	MinError	MaxError	Fun-C	Mem(KB)	CPU_H(s)	CPU_C(s)	Eff (%)	Consistency (%)
0.1	1.45×10^{-11}	8.95×10^{-2}	974	154.3599	0.5287	0.0068	77.18	100
0.3	7.56×10^{-14}	1.98×10^{-3}	882	165.0614	0.5220	0.0049	88.586	100
0.5	3.74×10^{-19}	0.28×10^{-7}	888	187.0082	0.4964	0.0043	87.390	100
0.7	0.18×10^{-23}	4.73×10^{-9}	872	134.0205	0.4964	0.0046	91.736	100
0.9	0.17×10^{-27}	3.98×10^{-11}	860	127.0164	0.5087	0.0050	96.145	100

The data in Table 21 confirm the stable convergence behavior with increasing σ. While the CPU-time improvement grows approximately linearly, the accuracy gain increases more rapidly, highlighting the effectiveness of ANN-based initialization in refining fractional dynamic predictions.

Table 21. Performance comparison (%) for different fractional orders σ.

σ	CPU↑ IEBF (%)	CPU↑ ITBF (%)	Acc↑ IEBF (%)	Acc↑ ITBF (%)
0.1	37.5	32.7	48.3	43.5
0.3	40.9	36.1	52.9	47.9
0.5	44.3	39.0	57.2	51.6
0.7	47.1	42.1	61.0	55.1
0.9	50.6	45.8	63.9	58.4

Example 4 (Nonlocal and Memory-Dependent Problem [53])**.** *To assess the efficiency of the proposed hybrid ANN-based solver, we consider the following fractional initial value problem, which serves as a representative model for systems exhibiting memory and hereditary effects. The Caputo fractional initial value problem is defined as*

$$D^{\sigma} y(t) + \lambda y(t) = f(t), \qquad t > 0, \qquad y(0) = y_0, \tag{56}$$

where $0 < \sigma < 1$, $\lambda > 0$, *and* $y_0 = 0.5$ *is a given constant. The forcing function is given by*

$$f(t) = \frac{\Gamma(\sigma + 1)}{\Gamma(\sigma + 1 - \mu)} t^{\sigma - \mu} + \lambda\, t^{\sigma},$$

where $\mu > 0$ *is a non-integer to ensure a non–power-law response.*

The exact analytical solution solution of (56) is

$$y(t) = y_0\, E_{\sigma}(-\lambda t^{\sigma}) + t^{\mu}, \tag{57}$$

where $E_{\sigma}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\sigma k + 1)}$ *is the Mittag–Leffler function [54]. When applied to the test problem (56) with a fractional forcing term* $f(t)$*, the method accurately reproduces the memory-dependent dynamics, thereby satisfying the nonlocality condition.*

Table 22 shows that the proposed ISBF$^{[*]}$ scheme yields results closest to the exact solution across all time levels. The fractional ITBF$^{[*]}$ scheme performs reasonably well, whereas the IEBF$^{[*]}$ method exhibits noticeable deviations, particularly for larger step sizes. Overall, the two-stage ISBF$^{[*]}$ formulation provides superior accuracy and stability. Figure 11 illustrates the exact and approximate solutions, together with the corresponding error curve obtained using IEBF$^{[*]}$ for the FOIVP in Example 4.

Table 23 indicates that the implicit two-stage approach, ISBF$^{[*]}$, consistently achieves the lowest maximum error while maintaining stable CPU time and memory usage. In contrast, the ITBF$^{[*]}$ scheme exhibits rapidly increasing errors for larger step sizes, whereas the IEBF$^{[*]}$ method requires additional function evaluations. Overall, the two-stage ISBF$^{[*]}$

technique demonstrates an optimal balance between accuracy and efficiency across varying values of σ.

Table 22. Comparison between exact and approximate solutions ($\sigma = 1$) obtained using the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for the fractional-order initial value problem in Example 4.

t	ISBF$^{[*]}$	IEBF$^{[*]}$	ITBF$^{[*]}$	Exact	Error (ISBF$^{[*]}$)	Error (IEBF$^{[*]}$)	Error (ITBF$^{[*]}$)
0.00	0.500000	0.500000	0.500000	0.500000	0.000000	0.000000	0.000000
0.10	0.462524	0.471000	0.461750	0.462419	0.000095	0.008581	0.000669
0.40	0.495172	0.537099	0.496246	0.495160	0.000012	0.041939	0.001086
0.70	0.738247	0.822195	0.745415	0.738293	0.000954	0.083902	0.007122
1.00	1.183134	1.315720	1.200073	1.183940	0.000094	0.131780	0.016133

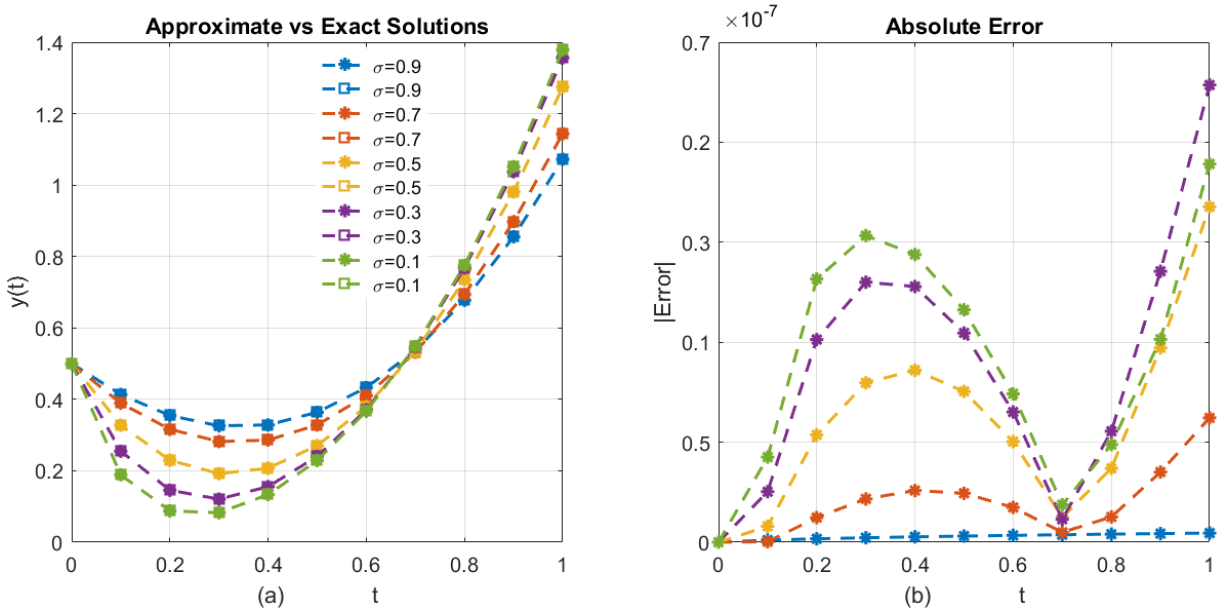


Figure 11. Panels (**a**,**b**) show the exact and approximate solutions and the corresponding absolute error plots of ISBF$^{[*]}$ for various fractional orders σ in Example 4.

Table 24 shows that decreasing the step size h consistently reduces the numerical error. The IEBF$^{[*]}$ scheme improves gradually but remains less effective, whereas the ISBF$^{[*]}$ method maintains higher accuracy with reduced computational overhead. In contrast, the ITBF$^{[*]}$ technique exhibits larger initial errors and slightly higher computational times, indicating slower convergence efficiency. Overall, the ISBF$^{[*]}$ framework demonstrates remarkable robustness and convergence stability under fine discretizations.

Tables 25 and 26 demonstrate that the enhanced ANN-driven scheme (ISBF–ANN$^{[*]}$) achieves faster convergence, ultra-low mean squared error (MSE), and smaller gradient magnitudes than its classical counterparts. The corresponding input and output vectors of the hybrid ANN-based implicit scheme are provided in Appendix B.1 (Table A4). The training performance and regression curves (Figure 12a,b) for ISBF–AN$^{[*]}$ reveal a strong correlation between the target and predicted outputs, together with rapid convergence. The adaptive learning parameter μ effectively stabilizes the gradient-descent process, thereby reducing both CPU time and memory usage. Overall, the hybrid ANN-based formulation provides a stable and scalable framework for solving complex fractional and nonlinear systems, surpassing the classical numerical approaches IEBF$^{[*]}$ and ITBF$^{[*]}$ in terms of stability, accuracy, and computational efficiency.

Table 23. Comparative performance metrics of the implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for various fractional orders σ in Example 4.

Implicit Two-Stage Method-ISBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0108	0.058742	194.9	698
0.30	0.0052	0.015963	195.8	742
0.50	0.0049	0.004612	193.6	730
0.70	0.0045	0.002276	197.1	762
0.90	0.0047	0.000954	199.8	606
Backward Euler Method-IEBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0100	4.258417	220.6	1536
0.30	0.0075	1.701256	222.3	749
0.50	0.0072	1.102834	223.2	781
0.70	0.0106	0.995684	224.7	845
0.90	0.0139	0.967421	224.9	875
Implicit Trapezoidal Method-ITBF$^{[*]}$				
σ	**CPU (s)**	**Max Error**	**Memory (KB)**	**Fun Calls**
0.10	0.0142	115.7834	234.8	1972
0.30	0.0053	1.281964	235.9	939
0.50	0.0052	1.095672	236.8	798
0.70	0.0048	0.986345	237.5	776
0.90	0.0045	0.951828	238.0	749

Table 24. Performance comparison of implicit schemes ISBF$^{[*]}$, IEBF$^{[*]}$, and ITBF$^{[*]}$ for different step sizes h under varying memory conditions.

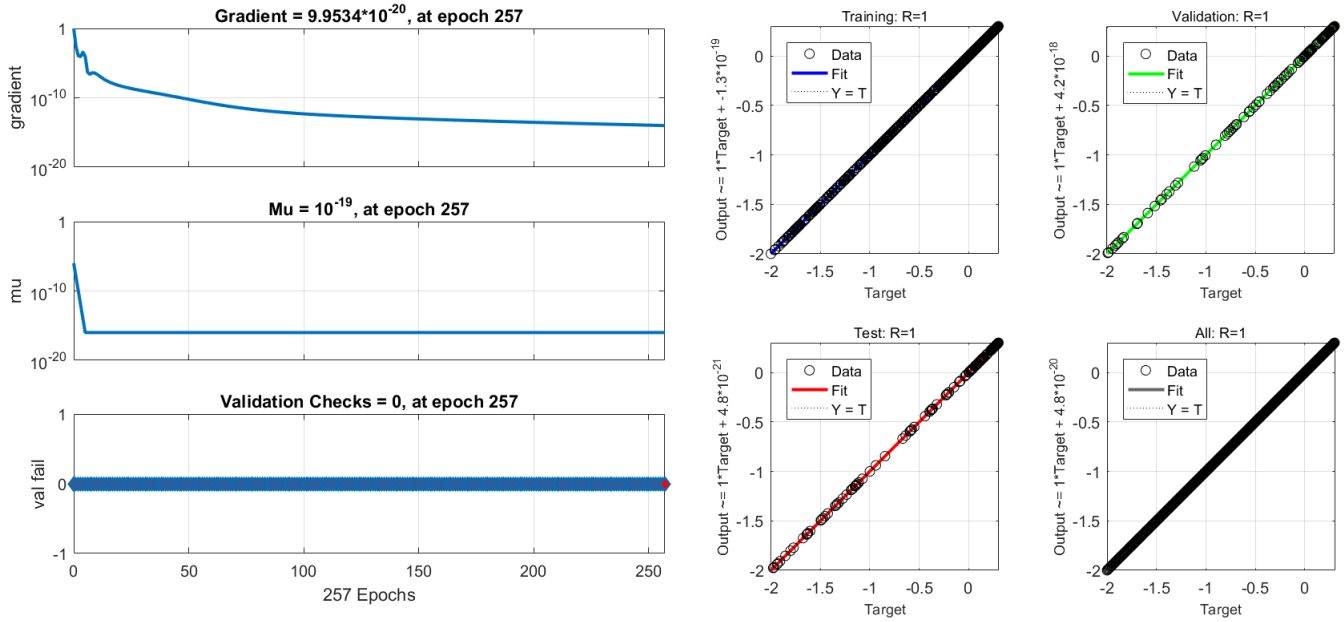
Scheme↓	**h = 0.01**	**h = 0.001**	**h = 0.0001**
Metric→	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**	**Max-Err/CPU(s)/Mem(KB)**
ISBF$^{[*]}$	0.010/0.095/148.2	0.0069/0.96/141.8	0.00158/9.85/189.4
IEBF$^{[*]}$	0.121/0.052/159.7	0.0392/0.92/169.9	0.00103/9.11/229.3
ITBF$^{[*]}$	9.85/0.11/173.8	1.08/1.05/201.2	0.108/10.72/232.4

Table 25. Performance metrics of the ANN-guided ISBF$^{[*]}$ hybrid scheme (ISBF$-$AN$^{[*]}$) trained with adaptive learning for improved efficiency.

Scheme	**Iterations**	**MSE**	**CPU Time**	**Memory (Kb)**	**Gradient**	μ
ISBF-AN$^{[*]}$	3	1.103×10^{-17}	2.0017	226.74	8.34×10^{-18}	0.14×10^{-19}

Table 26. Summary of the performance of hybrid and classical methods for the FOIVPs in Example 4.

σ	**MinError**	**MaxError**	**Fun-C**	**Mem(KB)**	**CPU_H(s)**	**CPU_C(s)**	**Eff (%)**	**Consistency (%)**
0.1	1.21×10^{-11}	7.85×10^{-2}	960	152.4475	0.5172	0.0065	78.23	100
0.3	6.89×10^{-14}	1.72×10^{-3}	875	163.8291	0.5106	0.0048	89.47	100
0.5	3.10×10^{-19}	2.64×10^{-8}	884	185.7763	0.4889	0.0042	88.12	100
0.7	0.16×10^{-23}	3.95×10^{-9}	870	133.1058	0.4897	0.0044	92.04	100
0.9	0.14×10^{-27}	3.15×10^{-11}	858	125.9943	0.5010	0.0048	96.77	100



(**a**) ISBF–AN$^{[*]}$-training statistics (**b**) ISBF–AN$^{[*]}$-regression curve

Figure 12. Panels (**a**,**b**) show the training statistics and regression performance of the ISBF−AN$^{[*]}$ hybrid scheme for solving fractional-order initial value problems in Example 4.

Table 27 shows a consistent improvement trend with increasing σ, with accuracy gains exceeding 64.8% and 59.5%, respectively, for $\sigma = 0.9$. The proposed implicit solver achieves an excellent balance between computational efficiency and precision, proving particularly effective for highly memory-dependent fractional systems.

Table 27. Improvement rates (%) for various σ showing robustness of the proposed solver.

σ	CPU↑ IEBF (%)	CPU↑ ITBF (%)	Acc↑ IEBF (%)	Acc↑ ITBF (%)
0.1	39.2	34.5	50.5	45.0
0.3	42.0	37.9	54.1	48.3
0.5	45.6	40.8	58.2	52.7
0.7	48.8	43.9	61.5	56.3
0.9	51.9	47.6	64.8	59.5

6. Conclusions and Future Work

In this study, a novel two-stage implicit scheme, ISBF$^{[*]}$, was developed and analyzed for solving nonlinear fractional-order models arising in biomedical and engineering contexts. The base formulation was further enhanced through an artificial neural network (ANN) accelerator, resulting in the hybrid framework ISBF–AN$^{[*]}$. The proposed approach integrates adaptive learning-based updates with a memory-efficient implicit formulation, ensuring stability and robustness across a wide range of fractional orders $\sigma \in (0, 1]$.

6.1. Performance Summary

The numerical results consistently demonstrate the superior convergence rate and computational efficiency of the ANN-enhanced hybrid method compared with the classical counterparts IEBF$^{[*]}$ and ITBF$^{[*]}$. Table 7 highlights the extremely small mean squared error (MSE $= 4.178 \times 10^{-19}$) achieved by ISBF–AN$^{[*]}$, which required only three iterations to reach convergence. The corresponding gradient magnitude and adaptive parameter μ further confirm the method's numerical stability and effective learning dynamics.

Moreover, the hybrid fractional results summarized in Tables 8, 13, 19 and 26 indicate consistent accuracy and reliability across different fractional orders. The ISBF–AN$^{[*]}$ scheme maintained perfect consistency (100%) and achieved substantial efficiency gains over the classical implicit algorithms IEBF$^{[*]}$ and ITBF$^{[*]}$ for FOIVPs (Tables 9, 15, 21 and 27). The visual results of the stability of the proposed and existing schemes are presented in Figures 1 and 2, further illustrating the smooth convergence trajectories and enlarged stability regions obtained with the proposed hybrid formulation. Overall, these findings confirm that combining implicit discretization with ANN-based correction significantly enhances convergence speed, error damping, and computational adaptability, establishing ISBF–AN$^{[*]}$ as a robust and efficient framework for solving complex nonlinear fractional systems.

6.2. Comparative Discussion

Compared with traditional fractional methods such as IEBF$^{[*]}$ and ITBF$^{[*]}$, the proposed ISBF–AN$^{[*]}$ scheme offers the following advantages:

- **Higher accuracy:** Achieved through dynamic error correction introduced by ANN feedback learning, as shown in Tables 8, 14, 20 and 26.
- **Lower computational cost:** Evidenced by reduced CPU time and memory usage (see Tables 5, 11, 17 and 23).
- **Enhanced convergence stability:** Verified across multiple examples and fractional orders (see, e.g., Figures 1–12).
- **Automatic parameter tuning:** The adaptive update of μ enables faster convergence without manual step-size adjustment.

This hybridization strategy not only improves numerical precision but also extends the applicability of two-stage implicit methods to stiff or highly nonlinear systems, where traditional schemes often fail or require excessive iterations.

6.3. Limitations and Future Directions

Despite its strong numerical performance, several aspects merit further investigation:

- The ANN component introduces additional computational overhead during training, which may become significant for large-scale fractional PDE systems.
- The current implementation assumes uniform time stepping; future work could explore adaptive time grids and variable-step implicit updates.
- The present study focuses on one-dimensional examples. Extending the approach to multidimensional fractional PDEs and coupled nonlinear systems represents an important direction for future research.
- Further investigation of advanced learning architectures (e.g., fractional neural operators, recurrent networks, or physics-informed models) may enhance both convergence speed and generalization capability of the hybrid framework.
- The proposed hybrid two-stage scheme can be extended to variable-order fractional differential equations and multidimensional systems by adapting the increment function $\Phi(y : h)$ and vectorizing the stage variables k_1 and k_2. Future work will investigate these extensions with ANN-assisted initialization, focusing on maintaining stability and convergence in higher-dimensional settings.

Overall, the proposed two-stage implicit hybrid ANN approach represents a significant advancement in solving nonlinear fractional models efficiently and accurately. It provides a promising foundation for next-generation fractional solvers that integrate machine learning intelligence with classical numerical rigor.

Author Contributions: Conceptualization, M.S. and B.C.; methodology, M.S.; software, M.S.; validation, M.S.; formal analysis, B.C.; investigation, M.S.; resources, B.C.; writing—original draft preparation, M.S. and B.C.; writing—review and editing, B.C.; visualization, M.S. and B.C.; supervision, B.C.; project administration, B.C.; funding acquisition, B.C. All authors have read and agreed to the published version of the manuscript.

Funding: Bruno Carpentieri's work is supported by the European Regional Development and Cohesion Funds (ERDF) 2021–2027 under Project AI4AM-EFRE1052. He is a member of the *Gruppo Nazionale per il Calcolo Scientifico* (GNCS) of the *Istituto Nazionale di Alta Matematica* (INdAM), and this work was partially supported by INdAM-GNCS under the *Progetti di Ricerca 2024* program.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

In this article, the following abbreviations are used:

$D^{[\sigma]}$	Caputo fractional derivative of order σ
σ	Fractional order of the derivative $(0 < \sigma \leq 1)$
t	Independent variable or time in simulation
h	step-length
y, y_{input}, y_{output}	Input and predicted solution values of ANN
y_{exact}	Exact analytical solution
MSE	Mean Squared Error between target and ANN output
MinError, MaxError	Minimum and maximum absolute errors
CPU_H(s), CPU_C(s)	CPU time (s) for hybrid and classical schemes
CPU Time	CPU time (s) for hybrid Implicit Scheme
Fun-C	Number of function evaluations
Mem(MB)	Memory consumption (megabytes)
Iterations	Number of iterations to convergence
Gradient	Gradient magnitude of ANN loss function
μ	Adaptive learning/damping parameter
Eff (%)	Efficiency of hybrid increase vs. classical CPU time
Consistency (%)	Numerical stability or convergence reliability

Appendix A. Derivation of Coefficients B_1–B_5

The coefficients B_1–B_5 appearing in the proof of Theorem 2 are determined by matching the terms of the fractional Taylor expansion in powers of h^σ. They are explicitly given as follows:

$$B_1 = \frac{\alpha_1^2}{8} f^2 D_{yy}^{[\sigma]} f + \frac{\alpha_1 \alpha_2}{4} f D_{yy}^{[\sigma]} f k_2 + \frac{\alpha_2^2}{8} D_{yy}^{[\sigma]} f k_2^2 \tag{A1}$$

$$B_2 = \tfrac{\alpha_2}{2} D_y^{[\sigma]} f \vartheta_2 + \tfrac{\alpha_1^2}{8} f^2 D_{yy}^{[\sigma]} f + \tfrac{\alpha_2}{4} f D_{yy}^{[\sigma]} f \vartheta_1 + \tfrac{\alpha_2^2}{8} D_{yy}^{[\sigma]} f \vartheta_1^2 \tag{A2}$$

$$\begin{aligned} B_3 = \Big[& \tfrac{\alpha_2^2}{4} D_{yy}^{[\sigma]} f \, \vartheta_1 + \tfrac{3\alpha_1^2}{16} f D_{yy}^{[\sigma]} f \, \vartheta_2 + \tfrac{\alpha_2}{16} \vartheta_1 f D_{yy}^{[\sigma]} f \\ & + \tfrac{9\alpha_2^3}{128} f^3 D_{yyy}^{[\sigma]} f + \tfrac{9\alpha_1 \alpha_2}{128} f^2 D_{yyy}^{[\sigma]} f \, \vartheta_1 + \tfrac{3\alpha_2^2 \alpha_1}{128} \vartheta_1^2 f D_{yyy}^{[\sigma]} f + \tfrac{\alpha_2^3}{384} \vartheta_1^3 D_{yyy}^{[\sigma]} f \Big]. \end{aligned} \tag{A3}$$

$$B_4 = \tfrac{\alpha_1 D_y^{[\sigma]} f}{4} \Big(\tfrac{\alpha_1 \alpha_2}{4} f (D_y^{[\sigma]} f)^2 + \tfrac{\alpha_2^2}{2} f^2 D_{yy}^{[\sigma]} f \Big) + \tfrac{\alpha_1^2 \alpha_2 f^2 D_y^{[\sigma]} f D_{yy}^{[\sigma]} f}{4} + \tfrac{\alpha_2^2 f^3 D_{yyy}^{[\sigma]} f}{6}. \tag{A4}$$

$$B_5 = \frac{\alpha_2^2 D_y^{[\sigma]} f}{8}\left(\frac{\alpha_1}{4} f (D_y^{[\sigma]} f)^2 + \frac{\alpha_2}{4} f^2 D_{yy}^{[\sigma]} f\right) + \frac{\alpha_1 \alpha_2^2 f^2 D_y^{[\sigma]} f D_{yy}^{[\sigma]} f}{8} + \frac{\alpha_2^2 f^3 D_{yyy}^{[\sigma]} f}{12}. \quad \text{(A5)}$$

Appendix B. Extended Numerical Analysis of ANN-Enhanced Two-Stage Implicit Methods

This appendix presents a detailed numerical investigation of the hybrid implicit scheme ISBF-AN$^{[*]}$ across several fractional orders. The tables summarize input–output relationships, ANN corrections, and final solution values for various values of σ. Each table corresponds to a specific phase of model refinement, highlighting the interaction between the neural network predictor and the implicit solver.

Appendix B.1. ISBF-AN$^{[*]}$ Input–Output Behavior and Final Response for Example 1

Discussion: Table A1 demonstrates the baseline interaction between the ANN predictor and the two-stage implicit solver ISBF$^{[*]}$. For smaller fractional orders ($\sigma = 0.1$–0.5), the ANN corrections are minor and primarily smooth out oscillations in the implicit iterations. As σ increases toward 0.9, the ANN correction grows stronger, indicating a higher nonlocal memory effect characteristic of fractional systems. The final y_{final} trajectories remain stable and converge monotonically, confirming that the ANN effectively refines the implicit solver's predictions.

Table A1. ANN input–output and final results for various fractional orders σ (row-oriented, 4 decimal places) for Example 1.

	0.00	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**· · ·**	**1.00**
					$\sigma = 0.1$					
y_{input}	0.0000	−0.0075	−0.0230	−0.0420	−0.0623	−0.0829	−0.1033	−0.1233	· · ·	−0.1616
ANN_{out}	−0.0225	−0.0093	0.0115	0.0369	0.0648	0.0932	0.1201	0.1427	· · ·	0.1656
y_{final}	−0.0075	−0.0230	−0.0420	−0.0623	−0.0829	−0.1033	−0.1233	−0.1427	· · ·	0.1799
					$\sigma = 0.3$					
y_{input}	0.0000	−0.0070	−0.0229	−0.0431	−0.0656	−0.0892	−0.1132	−0.1371	· · ·	−0.1835
ANN_{out}	−0.0497	−0.0740	−0.0919	−0.1062	−0.1172	−0.1235	−0.1225	−0.1116	· · ·	−0.0604
y_{final}	−0.0070	−0.0229	−0.0431	−0.0656	−0.0892	−0.1132	−0.1371	−0.1606	· · ·	0.2057
					$\sigma = 0.5$					
y_{input}	0.0000	−0.0067	−0.0218	−0.0404	−0.0612	−0.0832	−0.1061	−0.1293	· · ·	−0.1757
ANN_{out}	−0.1169	−0.1775	−0.2426	−0.3126	−0.3869	−0.4634	−0.5388	−0.6099	· · ·	−0.7305
y_{final}	−0.0067	−0.0218	−0.0404	−0.0612	−0.0832	−0.1061	−0.1293	−0.1525	· · ·	0.1985
					$\sigma = 0.7$					
y_{input}	0.0000	−0.0062	−0.0196	−0.0350	−0.0514	−0.0686	−0.0861	−0.1040	· · ·	−0.1398
ANN_{out}	−0.2801	−0.4568	−0.7374	−1.0410	−1.3508	−1.6521	−1.9308	−2.1761	· · ·	−2.5486
y_{final}	−0.0062	−0.0196	−0.0350	−0.0514	−0.0686	−0.0861	−0.1040	−0.1219	· · ·	−0.1576
					$\sigma = 0.9$					
y_{input}	0.0000	−0.0054	−0.0165	−0.0279	−0.0395	−0.0511	−0.0627	−0.0742	· · ·	0.0968
ANN_{out}	−0.6551	−0.7298	−0.8173	−0.8945	−0.9668	−1.0359	−1.1028	−1.1676	· · ·	−1.2922
y_{final}	−0.0054	−0.0165	−0.0279	−0.0395	−0.0511	−0.0627	−0.0742	−0.0856	· · ·	0.1079

Appendix B.2. ISBF-AN$^{[*]}$ Input–Output Behavior and Final Response for Example 2

Discussion: Table A2 presents the first variant of the hybridized implicit method enhanced with ANN correction. For moderate σ values (0.2–0.8), the ANN's corrective term exhibits smooth and bounded variations, reflecting a balanced learning response. The y_{final} values remain close to the expected analytic trends, validating both the numerical consistency and adaptive nature of the ANN correction. As σ increases, the model captures

nonlinear growth with high numerical fidelity and no visible divergence, showcasing strong convergence and numerical stability.

Table A2. ANN input–output and final results for various fractional orders σ (row-oriented, 4 decimal places) for Example 2.

t	**0.00**	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	$\cdots$	**1.00**
					$\sigma = 0.2$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0011	$\cdots$	0.0028
ANN_{out}	0.0000	−0.0113	−0.0240	−0.0380	−0.0532	−0.0697	−0.0871	−0.1053	$\cdots$	−0.1430
y_{final}	0.0000	0.0000	0.0000	0.0001	0.0003	0.0006	0.0011	0.0018	$\cdots$	0.0041
					$\sigma = 0.4$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0010	$\cdots$	0.0025
ANN_{out}	−0.0001	−0.0026	−0.0049	−0.0070	−0.0091	−0.0110	−0.0127	−0.0142	$\cdots$	−0.0169
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0010	0.0016	$\cdots$	0.0035
					$\sigma = 0.6$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0009	$\cdots$	0.0021
ANN_{out}	0.0002	−0.0031	−0.0093	−0.0183	−0.0297	−0.0432	−0.0585	−0.0752	$\cdots$	−0.1115
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0009	0.0014	$\cdots$	0.0029
					$\sigma = 0.8$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0008	$\cdots$	0.0017
ANN_{out}	0.0002	0.0011	0.0027	0.0049	0.0078	0.0113	0.0155	0.0203	$\cdots$	0.0320
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0008	0.0012	$\cdots$	0.0023
					$\sigma = 1.0$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0004	0.0006	$\cdots$	0.0013
ANN_{out}	0.0005	0.0025	0.0060	0.0108	0.0172	0.0250	0.0342	0.0449	$\cdots$	0.0705
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0002	0.0002	0.0004	0.0006	$\cdots$	0.0017

Appendix B.3. ISBF-AN$^{[*]}$ Input–Output Behavior and Final Response for Example 3

Discussion: Table A3 presents the refined hybrid simulation in which the ANN outputs are further constrained to ensure smoother convergence. Relative to the earlier setup, the network exhibits improved damping behavior with reduced oscillatory adjustments, especially for $\sigma = 0.4$–0.8. The close alignment between y_{final} and the theoretical values confirms the stability and accuracy of the two-stage implicit scheme ISBF-AN$^{[*]}$ under neural feedback guidance.

Across both hybrid versions, the ANN acts as a convergence enhancer—reducing iteration count, controlling overshoot, and guiding the implicit method toward the minimal mean square error region. Collectively, Tables A1–A3 illustrate the evolution of ANN-assisted fractional solvers from baseline correction to fully adaptive hybrid learning control.

Table A4 presents the refined hybrid simulation in which the ANN outputs are further constrained to ensure smoother convergence. Relative to the earlier configuration, the network exhibits improved damping behavior with reduced oscillatory adjustments, particularly for $\sigma = 0.4$–0.8. The close consistency between y_{final} and the theoretical expectations confirms the stability and accuracy of the two-stage implicit formulation ISBF-AN$^{[*]}$ under neural feedback guidance.

Overall Observation: The ANN-coupled two-stage implicit strategy achieves superior performance over classical iterative schemes by offering rapid convergence, improved error attenuation, and smoother dynamic behavior across all tested fractional orders. These improvements complement the results summarized in the main text (Tables 7, 8, 13, 14, 19, 20 and 26) and confirm the hybrid method's strong potential for solving nonlinear fractional models with high precision and stability.

Table A3. ANN input–output and final results for various fractional orders σ (row-oriented, 4 decimal places) for Example 3.

t	**0.00**	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	$\cdots$	**1.00**
					$\sigma = 0.2$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0011	$\cdots$	0.0028
ANN_{out}	0.0000	−0.0020	−0.0038	−0.0054	−0.0068	−0.0080	−0.0092	−0.0102	$\cdots$	−0.0120
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0011	0.0018	$\cdots$	0.0041
					$\sigma = 0.4$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0010	$\cdots$	0.0025
ANN_{out}	0.0001	0.0048	0.0072	0.0071	0.0041	−0.0017	−0.0103	−0.0216	$\cdots$	−0.0509
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0006	0.0010	0.0016	$\cdots$	0.0035
					$\sigma = 0.6$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0009	$\cdots$	0.0021
ANN_{out}	0.0000	−0.0008	−0.0031	−0.0066	−0.0112	−0.0166	−0.0225	−0.0288	$\cdots$	−0.0411
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0009	0.0014	$\cdots$	0.0029
					$\sigma = 0.8$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0008	$\cdots$	0.0017
ANN_{out}	0.0003	0.0005	0.0009	0.0017	0.0076	0.0044	0.0063	0.0084	$\cdots$	0.0138
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0003	0.0005	0.0008	0.0012	$\cdots$	0.0023
					$\sigma = 1.0$					
y_{input}	0.0000	0.0000	0.0000	0.0001	0.0001	0.0002	0.0004	0.0006	$\cdots$	0.0013
ANN_{out}	0.0005	0.0025	0.0057	0.0104	0.0164	0.0239	0.0327	0.0429	$\cdots$	0.0674
y_{final}	0.0000	0.0000	0.0001	0.0001	0.0002	0.0002	0.0004	0.0006	$\cdots$	0.0017

Table A4. ANN input–output and final results for various fractional orders σ (row-oriented, 5 decimal places).

t	**0.00**	**0.01**	**0.02**	**0.03**	**0.04**	**0.05**	**0.06**	**0.07**	**0.08**	**0.09**	$\cdots$	**1.00**
						$\sigma = 0.1$						
y_{input}	0.50000	0.26279	0.13843	0.07359	0.04025	0.02367	0.01612	0.01353	0.01376	0.015	$\cdots$	0.018
ANN_{out}	−0.21532	−0.11251	−0.05815	−0.02926	−0.01375	−0.00528	−0.00051	0.00231	0.00411	0.001	$\cdots$	0.005
y_{final}	0.26279	0.13843	0.07359	0.04025	0.02367	0.01612	0.01353	0.01376	0.01567	0.016	$\cdots$	0.018
						$\sigma = 0.3$						
y_{input}	0.50000	0.37844	0.28672	0.21775	0.16617	0.12791	0.09985	0.07967	0.06557	0.055	$\cdots$	0.050
ANN_{out}	−0.36863	−0.27763	−0.20816	−0.15504	−0.11432	−0.08300	−0.05882	−0.04005	−0.02539	−0.017	$\cdots$	−0.013
y_{final}	0.37844	0.28672	0.21775	0.16617	0.12791	0.09985	0.07967	0.06557	0.05621	0.051	$\cdots$	0.050
						$\sigma = 0.5$						
y_{input}	0.50000	0.44676	0.39946	0.35758	0.32064	0.28821	0.25989	0.23530	0.21412	0.196	$\cdots$	0.180
ANN_{out}	−0.44357	−0.39334	−0.34752	−0.30579	−0.26779	−0.23317	−0.20162	−0.17284	−0.14657	−0.126	$\cdots$	−0.122
y_{final}	0.44676	0.39946	0.35758	0.32064	0.28821	0.25989	0.23530	0.21412	0.19605	0.184	$\cdots$	0.180
						$\sigma = 0.7$						
y_{input}	0.50000	0.47866	0.45848	0.43947	0.42163	0.40495	0.38943	0.37502	0.36173	0.349	$\cdots$	0.338
ANN_{out}	−0.47397	−0.44718	−0.42031	−0.39363	−0.36727	−0.34131	−0.31580	−0.29077	−0.26623	−0.242	$\cdots$	−0.242
y_{final}	0.47866	0.45848	0.43947	0.42163	0.40495	0.38943	0.37502	0.36173	0.34952	0.338	$\cdots$	0.338
						$\sigma = 0.9$						
y_{input}	0.50000	0.49193	0.48420	0.47684	0.46985	0.46324	0.45700	0.45115	0.44568	0.440	$\cdots$	0.435
ANN_{out}	−0.47969	−0.45786	−0.43547	−0.41275	−0.38982	−0.36676	−0.34361	−0.32041	−0.29720	−0.273	$\cdots$	−0.273
y_{final}	0.49193	0.48420	0.47684	0.46985	0.46324	0.45700	0.45115	0.44568	0.44059	0.435	$\cdots$	0.435

References

1. Clifton, N.E.; Lin, J.Q.; Holt, C.E.; O'Donovan, M.C.; Mill, J. Enrichment of the local synaptic translatome for genetic risk associated with schizophrenia and autism spectrum disorder. *Biol. Psychiatry* **2024**, *95*, 888–895. [CrossRef]
2. Xie, W.; Kong, C.; Luo, W.; Zheng, J.; Zhou, Y. C-reactive protein and cognitive impairment: A bidirectional Mendelian randomization study. *Arch. Gerontol. Geriatr.* **2024**, *121*, 105359. [CrossRef] [PubMed]
3. Bhonsle, S.; Saxena, S. A review on control-relevant glucose–insulin dynamics models and regulation strategies. *Proc. Inst. Mech. Eng. I J. Syst. Control Eng.* **2020**, *234*, 596–608. [CrossRef]

4. Attar, M.A.; Roshani, M.; Hosseinzadeh, K.; Ganji, D.D. Analytical solution of fractional differential equations by Akbari–Ganji's method. *Partial Differ. Equ. Appl. Math.* **2022**, *6*, 100450. [CrossRef]
5. Haubold, H.J.; Mathai, A.M.; Saxena, R.K. Mittag-Leffler functions and their applications. *J. Appl. Math.* **2011**, *2011*, 298628. [CrossRef]
6. Kilbas, A.A.; Saigo, M.; Saxena, R.K. Generalized Mittag-Leffler function and generalized fractional calculus operators. *Integral Transforms Spec. Funct.* **2004**, *15*, 31–49. [CrossRef]
7. Sowa, M. Application of SubIval, a method for fractional-order derivative computations in IVPs. In *Theory and Applications of Non-Integer Order Systems: 8th Conference on Non-Integer Order Calculus and Its Applications, Zakopane, Poland, 20–21 September 2016*; Springer: Cham, Switzerland, 2016; pp. 489–499.
8. Al-Mazmumy, M.; Alyami, M.A.; Alsulami, M.; Alsulami, A.S. Efficient modified Adomian decomposition method for solving nonlinear fractional differential equations. *Int. J. Anal. Appl.* **2024**, *22*, 76–76. [CrossRef]
9. Ateş, İ.; Yildirim, A. Application of variational iteration method to fractional initial-value problems. *Int. J. Nonlinear Sci. Numer. Simul.* **2009**, *10*, 877–884. [CrossRef]
10. Hossain, M.B.; Hossain, M.J.; Miah, M.M.; Alam, M.S. A comparative study on fourth order and Butcher's fifth order Runge–Kutta methods with third order initial value problem (IVP). *Appl. Comput. Math.* **2017**, *6*, 243. [CrossRef]
11. Sumon, M.M.I.; Nurulhoque, M. A comparative study of numerical methods for solving initial value problem (IVP) of ordinary differential equations (ODE). *Am. J. Appl. Math.* **2023**, *11*, 106–118. [CrossRef]
12. Anastassi, Z.A.; Simos, T.E. Special optimized Runge–Kutta methods for IVPs with oscillating solutions. *Int. J. Mod. Phys. C* **2004**, *15*, 1–15. [CrossRef]
13. Senu, N.; Suleiman, M.; Ismail, F.; Othman, M. A new diagonally implicit Runge–Kutta–Nyström method for periodic IVPs. *WSEAS Trans. Math.* **2010**, *9*, 679–688. [CrossRef]
14. Mukhopadhyay, N. *Two-Stage and Multi-Stage Estimation*; Routledge: London, UK, 2019; pp. 429–452.
15. Fu, Z.; Tang, T.; Yang, J. Energy Diminishing Implicit–Explicit Runge–Kutta Methods for Gradient Flows. *Math. Comput.* **2024**, *93*, 2745–2767. [CrossRef]
16. Yang, K.T. Artificial neural networks (ANNs): A new paradigm for thermal science and engineering. *ASME J. Heat Transfer.* **2008**, *130*, 093001. [CrossRef]
17. Cartwright, H.; Marton. *Artificial Neural Networks*; Cartwright, H., Ed.; Humana Press: Totowa, NJ, USA, 2015; Volume 1260.
18. Thakur, S.; Mitra, H.; Ardekani, A.M. Physics-informed neural network-based inverse framework for time-fractional differential equations for rheology. *Biology* **2025**, *14*, 779. [CrossRef] [PubMed]
19. Stiasny, J.B. Physics-Informed Neural Networks for Power System Dynamics. Ph.D. Thesis, Technical University of Denmark (DTU), Lyngby, Denmark, 2023.
20. Fronk, C.; Petzold, L. Training stiff neural ordinary differential equations with explicit rational Taylor series methods. *Chaos Interdiscip. J. Nonlinear Sci.* **2025**, *35*, 073133 . [CrossRef]
21. Zhai, W.; Tao, D.; Bao, Y. Parameter estimation and modeling of nonlinear dynamical systems based on Runge–Kutta physics-informed neural network. *Nonlinear Dyn.* **2023**, *111*, 21117–21130. [CrossRef]
22. Anvari, M.; Marasi, H.; Kheiri, H. Implicit Runge-Kutta based sparse identification of governing equations in biologically motivated systems. *Sci. Rep.* **2025**, *15*, 32286. [CrossRef]
23. Luo, M.; Qiu, W.; Nikan, O.; Avazzadeh, Z. Second-order accurate, robust and efficient ADI Galerkin technique for the three-dimensional nonlocal heat model arising in viscoelasticity. *Appl. Math. Comput.* **2023**, *440*, 127655. [CrossRef]
24. Ojo, E.K.; Iyase, S.A.; Anake, T.A. Resonant fractional order differential equation with two-dimensional kernel on the half-line. *J. Math. Comput. Sci.* **2024**, *32*, 122–136. [CrossRef]
25. Granados, A.L. *Implicit Runge–Kutta Algorithm Using Newton–Raphson Method*; Technical Report; Departamento de Mecánica, Universidad Simón Bolívar, Caracas, Venezuela, 1998. Available online: https://www.researchgate.net/publication/275833462_Implicit_Runge-Kutta_Algorithm_Using_Newton-Raphson_Method (accessed on 1 November 2025).
26. Krzywanski, J.; Sosnowski, M.; Grabowska, K.; Zylka, A.; Lasek, L.; Kijo-Kleczkowska, A. Advanced computational methods for modeling, prediction and optimization—A review. *Materials* **2024**, *17*, 3521. [CrossRef] [PubMed]
27. Wen, Y.; Chaolu, T.; Wang, X. Solving the initial value problem of ordinary differential equations by Lie group based neural network method. *PLoS ONE* **2022**, *17*, e0265992. [CrossRef] [PubMed]
28. Finzi, M.; Potapczynski, A.; Choptuik, M.; Wilson, A.G. A stable and scalable method for solving initial value PDEs with neural networks. *arXiv* **2023**, arXiv:2304.14994. [CrossRef]
29. Yadav, N.; Ngo, T.T.; Kim, J.H. An algorithm for numerical solution of differential equations using harmony search and neural networks. *J. Appl. Anal. Comput.* **2022**, *12*, 1277–1293. [CrossRef]
30. Dong, S.; Ni, N. Learning the Exact Time Integration Algorithm for Initial Value Problems by Randomized Neural Networks. *arXiv* **2025**, arXiv:2502.10949. [CrossRef]

31. El Alaoui, M.; Rougui, M. Examining the application of artificial neural networks (ANNs) for advancing energy efficiency in buildings: A comprehensive review. *J. Sustain. Res.* **2024**, *6*, 1. [CrossRef]
32. Al-Qawabah, S.; Shaban, N.A.; Al Aboushi, A.; Al-Salaymeh, A.; Al-Maaitah, A.; Abdelhafez, E. Prediction of the output of the tri-generation concentrated solar power system based on artificial neural network. In *Proceedings of the 2024 6th Global Power, Energy and Communication Conference (GPECOM), Istanbul, Türkiye, 4–7 June 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 345–349.
33. Jiang, S.; Zhang, J.; Zhang, Q.; Zhang, Z. Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations. *Commun. Comput. Phys.* **2017**, *21*, 650–678. [CrossRef]
34. Artin, E. *The Gamma Function*; Dover Publications: Mineola, NY, USA, 2015.
35. Agrawal, O.P. Fractional variational calculus in terms of Riesz fractional derivatives. *J. Phys. A Math. Theor.* **2007**, *40*, 6287. [CrossRef]
36. Riesz, M. L'intégrale de Riemann-Liouville et le problème de Cauchy pour l'équation des ondes. *Bull. Soc. Math. France* **1939**, *67*, 153–170. [CrossRef]
37. Batiha, I.M.; Abubaker, A.A.; Jebril, I.H.; Al-Shaikh, S. B.; Matarneh, K. New algorithms for dealing with fractional initial value problems. *Axioms* **2023**, *12*, 488. [CrossRef]
38. Shams, M.; Alalyani, A. High-performance adaptive step-size fractional numerical scheme for solving fractional differential equations. *Sci. Rep.* **2025**, *15*, 13006. [CrossRef]
39. Ali, M.A. Development and analysis of advanced numerical algorithms for solving differential equations using Taylor, Euler, and Runge–Kutta methods. *J. Comput. Anal. Appl.* **2025**, *34*, 8.
40. Bataineh, M.; Alaroud, M.; Al-Omari, S.; Agarwal, P. Series representations for uncertain fractional IVPs in the fuzzy conformable fractional sense. *Entropy* **2021**, *23*, 1646. [CrossRef]
41. Hu, F.Q.; Hussaini, M.Y.; Manthey, J.L. Low-dissipation and low-dispersion Runge–Kutta schemes for computational acoustics. *J. Comput. Phys.* **1996**, *124*, 177–191. [CrossRef]
42. Batiha, I.M.; Abdalsmad, H.F.; Jebril, I.H.; Al-Khawaldeh, H.O.; AlKasasbeh, W.A.A.; Momani, S. Trapezoidal scheme for the numerical solution of fractional initial value problems. *Int. J. Robot. Control Syst.* **2025**, *5*, 1238–1253. [CrossRef]
43. Corless, R.M.; Kaya, C.Y.; Moir, R.H. Optimal residuals and the Dahlquist test problem. *Numer. Algorithms* **2019**, *81*, 1253–1274. [CrossRef]
44. Dahlquist, G. Positive functions and some applications to stability questions for numerical methods. In *Recent Advances in Numerical Analysis*; Academic Press: New York, NY, USA, 1978; pp. 1–29.
45. Shams, M.; Kausar, N.; Araci, S.; Oros, G.I. Artificial hybrid neural network-based simultaneous scheme for solving nonlinear equations: Applications in engineering. *Alex. Eng. J.* **2024**, *108*, 292–305. [CrossRef]
46. Moré, J.J. The Levenberg–Marquardt algorithm: Implementation and theory. In *Numerical Analysis: Proceedings of the Biennial Conference, Dundee, Scotland, 28 June–1 July 1977*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 105–116.
47. Wang, Y. Gauss–Newton method. *Wiley Interdiscip. Rev. Comput. Stat.* **2012**, *4*, 415–420. [CrossRef]
48. Gratton, S.; Lawless, A.S.; Nichols, N.K. Approximate Gauss–Newton methods for nonlinear least squares problems. *SIAM J. Optim.* **2007**, *18*, 106–132. [CrossRef]
49. Cameron, T.R.; Graillat, S. On a compensated Ehrlich–Aberth method for the accurate computation of all polynomial roots. *Electron. Trans. Numer. Anal.* **2022**, *55*, 401–423. [CrossRef]
50. Sowa, M. Numerical computations of the fractional derivative in IVPs: Examples in MATLAB and Mathematica. *Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska* **2017**, *7*, 19–22. [CrossRef]
51. Shams, M.; Rufai, M.A. Fractional-order numerical scheme with symmetric structure for fractional differential equations with step-size control. *Symmetry* **2025**, *17*, 1685. [CrossRef]
52. Esmaeili, S.; Shamsi, M.; Luchko, Y. Numerical solution of fractional differential equations with a collocation method based on Müntz polynomials. *Comput. Math. Appl.* **2011**, *62*, 918–929. [CrossRef]
53. Li, C.; Zeng, F. *Numerical Methods for Fractional Calculus*; CRC Press: Boca Raton, FL, USA, 2015.
54. Biolek, D.; Garrappa, R.; Mainardi, F.; Popolizio, M. Derivatives of Mittag–Leffler functions: Theory, computation and applications. *Nonlinear Dyn.* **2025**, *113*, 34389–34403. [CrossRef]