RESEARCH ARTICLE

A Third-Order Fully Implicit Two-Point Superclass of Fractional Block Backward Differentiation Formula for Stiff Systems of Fractional-Order Differential Equations

Buhari Alhassan[1,*], **Hamisu Musa**[2], **Najamuddeen Bala**[3], **Abdulrahman Adamu**[1], **Sharhabilu Umar Wasagu**[4], **Kabiru Abubakar Kantsi**[1], **Abubakar Jibrin**[1], **Hamza Yusuf**[5], **Asmau Abdullahi Mani**[2] **and Muhammad Abdullahi**[5]

[1] Department of Mathematics and Statistics, Al-Qalam University Katsina, Katsina State, Nigeria
[2] Department of Mathematics, Umaru Musa Yar'adua University, Katsina, Katsina State, Nigeria
[3] Department of Statistics, Federal Polytechnic Kaura-Namoda, Zamfara State, Nigeria
[4] Department of Mathematics, Federal College of Education, Katsina, Katsina State, Nigeria
[5] Department of Mathematics, Federal University Dutsin-Ma, Katsina State, Nigeria

Abstract

This paper constructs a third-order fully implicit two-point superclass fractional block backward differentiation formula (2SFBBDF3) for stiff systems of fractional-order differential equations with the Caputo derivative. The proposed scheme computes the solution at two consecutive grid points per step and incorporates a stability-control parameter ρ, chosen arbitrarily from the stiff stability interval $-1 < \rho < 1$. Setting $\rho = 0$, it recovers the existing 2FBBDF3 method, while $\rho = -3/4$ is adopted throughout this study. The method coefficients are derived from Taylor-type expansions of the solution and of the Caputo derivative. For $\alpha = 0.75$, 0.85, 0.95 and 1 the method is shown to be of order three, consistent, zero-stable and hence convergent, and its regions of absolute stability are obtained from the fractional Dahlquist test equation. These regions contain the left half-plane for $\alpha = 0.75$-0.95, whereas for $\alpha = 1$ the boundary slightly enters the left half-plane near the imaginary axis. Numerical experiments, restricted to the classical limit $\alpha = 1$, compare the method with 2FBBDF3 and 2DIFBBDF2 on three stiff systems with identical step sizes. 2SFBBDF3 gives the smallest maximum error at every step size without larger CPU times, although the observed convergence rates are below the theoretical order. Genuinely fractional experiments are left for future work.

Keywords: fractional differential equations, Caputo fractional derivative, fractional linear multistep method, fractional block backward differentiation formula, superclass, stability-control parameter.

Submitted: 11 June 2026
Revised: 04 September 2026
Accepted: 17 October 2026
Published: 23 November 2026

Vol. 2, **No.** 4, 2026.
10.62762/JMIA.2026.761121

***Corresponding author:**
✉ Buhari Alhassan
buharialhassan@auk.edu.ng

Citation
Alhassan, B., Musa, H., Bala, N., Adamu, A., Wasagu, S. U., Kantsi, K. A., Jibrin, A., Yusuf, H., Mani, A. A., & Abdullahi, M. (2026). A Third-Order Fully Implicit Two-Point Superclass of Fractional Block Backward Differentiation Formula for Stiff Systems of Fractional-Order Differential Equations. *Journal of Mathematics and Interdisciplinary Applications*, 2(4), 244–261.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

1 Introduction

Fractional calculus generalizes classical calculus by allowing derivatives and integrals of arbitrary real or complex order instead of only positive integer order. The mathematical theory is several centuries old, but its use in modeling has grown quickly over the last decades, mainly because fractional-order operators give a natural description of systems with memory, hereditary behavior, anomalous transport and other nonlocal effects [1–4]. Behavior of this kind is observed in viscoelastic materials, diffusion processes, control systems, electrochemistry, biological tissue and many engineering materials, and fractional models are now used routinely in these areas; broad surveys of such applications are given in [5], with bioengineering and viscoelastic wave applications treated in [6] and [7], respectively. As a consequence, fractional differential equations (FDEs) have become an important research topic in applied and computational mathematics.

Several non-equivalent definitions of a fractional derivative are in use, and the Riemann-Liouville and Caputo derivatives are the two most common. For initial-value problems the Caputo derivative is usually preferred: the initial conditions are then imposed on the unknown function and its integer-order derivatives, which have a clear physical meaning, whereas the Riemann-Liouville form requires initial values of fractional integrals [2, 8]. For $0 < \alpha < 1$, the Caputo derivative of a sufficiently smooth function $v(t)$ is

$$ {}_{t_0}^{C}D_t^{\alpha} v(t) = \frac{1}{\Gamma(1-\alpha)} \int_{t_0}^{t} \frac{v'(s)}{(t-s)^{\alpha}} \, \mathrm{d}s, \qquad (1) $$

where $\Gamma(\cdot)$ is the Gamma function. A Caputo initial-value problem then reads

$$ {}_{t_0}^{C}D_t^{\alpha} v(t) = f\big(t, v(t)\big), \qquad t \geq t_0, \qquad v(t_0) = v_0. \quad (2) $$

Throughout the paper v may be a scalar or a vector-valued function, and f is assumed to be smooth enough for the expansions used in Section 2.

The analytical theory of (2), including existence and uniqueness of solutions, their regularity near the initial point and their stability, has been studied extensively [9–11]. Closed-form solutions, however, are known only for a restricted class of FDEs, mostly linear equations with constant coefficients whose solutions can be written in terms of Mittag-Leffler-type functions. For nonlinear equations and for the systems that arise in applications, numerical methods are usually the only practical tool, and this explains the continuing interest in reliable and efficient numerical schemes for FDEs.

A major difficulty in the numerical treatment of FDEs is the nonlocal nature of the fractional derivative. An integer-order derivative can be approximated from a few neighboring values, but the Caputo derivative at time t depends on the solution over the whole interval $[t_0, t]$. A step of a numerical method therefore involves, in principle, all previously computed values, and both the arithmetic cost and the memory requirement grow with the number of steps [12, 13]. Ford and Simpson [13] studied how this cost can be reduced without losing too much accuracy, and Diethelm et al. [12] compared several algorithms for fractional calculus with respect to accuracy and computational effort. A new method for FDEs therefore has to be judged not only by its accuracy and stability but also by the way it treats this memory.

Many classes of numerical methods have been proposed for FDEs, among them finite-difference and product-integration schemes, spectral and collocation methods, predictor-corrector and fractional Adams methods, fractional linear multistep methods and convolution quadrature [12, 14–17]; a recent survey with accompanying software is given by Garrappa [17]. Odibat [18] derived approximations of fractional integrals and of Caputo derivatives that can be used as building blocks for such schemes, and Scherer et al. [19] analyzed the stability, convergence and error behavior of the Grünwald-Letnikov approach. Most of this work aims at methods that are accurate and stable while keeping the cost of the memory term under control.

Among these approaches, fractional linear multistep methods (FLMMs) hold a special position, because they connect the well-developed theory of multistep methods for ordinary differential equations (ODEs) with fractional problems. Lubich [20] introduced FLMMs for Abel-Volterra integral equations of the second kind and then developed a general convolution-quadrature framework for fractional calculus [21, 22]. In this framework the generating polynomial of a classical linear multistep method defines the convolution weights that approximate the fractional operator, so that the order and stability of the underlying ODE method carry over to the fractional scheme in a controlled way. Galeone and Garrappa [23] later studied multistep methods for FDEs in detail and showed that the order and stability conditions depend on the fractional order;

they cannot simply be copied from the integer-order case. Garrappa [24] constructed stability-preserving high-order methods for multi-term FDEs, and Aceto et al. [25] developed fractional convolution quadrature based on generalized Adams methods.

Fractional Adams-type methods form another well-studied family. Diethelm et al. [14] proposed a predictor-corrector approach for Caputo-type FDEs that applies to linear and nonlinear problems alike and is now one of the most widely used methods in practice. In a subsequent paper [15] they gave a detailed error analysis of this fractional Adams method and established its convergence order under suitable smoothness assumptions. Galeone and Garrappa investigated fractional Adams-Moulton methods [26] and the linear stability of explicit methods for FDEs [27], and Garrappa [16] studied explicit Adams multistep methods. Much of what is known about the accuracy and stability of Adams-type schemes for Caputo problems comes from these papers.

Backward differentiation formulas (BDFs) are a second important class of implicit linear multistep methods (LMMs). For stiff ODEs they are among the most widely used integrators, because they have large stability regions and their implicit equations can be solved efficiently by Newton-type iterations [28, 29]. Extending them to stiff FDEs is therefore a natural step. Lubich's convolution-quadrature framework provides one such extension [20–22]; other fractional BDF formulations have been built from interpolation, collocation or Taylor expansion. Bonab and Javidi [30], for example, constructed higher-order methods from a third-order fractional BDF and proved convergence and stability results for them. Their work shows that BDF-based schemes can give accurate solutions of FDEs with good stability properties, and it suggests looking for BDF variants with additional free parameters that can be used to adjust the stability.

A further development is the combination of fractional BDFs with the block idea. Unlike conventional one-point multistep methods, a block method produces several new approximations in each step. This reduces the number of steps needed to cover a given interval and allows the values inside a block to be computed in parallel. Biala and Jator [31] derived block BDFs for FDEs by interpolation and collocation and analyzed their stability through the associated stability functions; their results show that block BDFs are a feasible approach to fractional initial-value problems. The same authors also developed block implicit Adams methods for FDEs [32], which shows that the block structure can be combined with different multistep families.

Stability is a central issue in the design of methods for FDEs. In classical numerical analysis the stability of a method describes how it propagates perturbations and how well it reproduces the qualitative behavior of the underlying equation. For FDEs the analysis is more involved, because the discrete scheme inherits the memory of the fractional operator; stability results for fractional multistep, multi-term and Grünwald-Letnikov schemes are given in [19, 23, 24]. The standard test problem is

$$ {}_{t_0}^{C}D_t^{\alpha} v(t) = \lambda v(t), \qquad \lambda \in \mathbb{C}, \qquad v(t_0) = v_0, \tag{3} $$

where λ is a complex parameter (for a system, an eigenvalue of the Jacobian matrix). Its exact solution is $v(t) = v_0 E_\alpha\big(\lambda (t - t_0)^\alpha\big)$, where

$$ E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \qquad \mathrm{Re}(\alpha) > 0, \tag{4} $$

is the Mittag-Leffler function, which plays the same role for (3) as the exponential function does for the Dahlquist equation $y' = \lambda y$ [2, 33].

A stability analysis of a fractional method therefore has to take into account both the fractional order and the structure of the discretized operator, and the resulting stability region tells us for which values of $h^\alpha \lambda$ the method produces bounded approximations.

For fractional BDF-type and block methods, stability results have been reported by Biala and Jator [31] and by Bonab and Javidi [30]. More recently, Noor et al. [34] derived a two-point fractional block BDF by combining a fractional linear multistep formulation with a linear difference operator. They investigated its stability for $0 < \alpha < 1$, reported A-stability, and tested the method on linear, nonlinear and system FDEs. Their two-point formulation is the starting point of the present work. Taken together, the literature provides a solid basis through FLMMs [20–23], fractional Adams methods [14–16, 26, 27], fractional BDF schemes [30] and fractional block methods [31, 32, 34]. Less attention has been paid to fractional block BDFs that contain a free parameter for adjusting their stability.

One way to obtain such flexibility is to add a stability-control parameter to a fractional block BDF. For ODEs this idea leads to the superclass of block BDFs [35], in which a parameter ρ multiplies

an additional derivative term and different values of ρ give methods with different stability regions. Introducing a parameter changes every coefficient of the formula, so the consistency, zero-stability, convergence and stability of the resulting fractional scheme have to be checked directly; they cannot be inferred from the known properties of classical BDFs. The parameter also enlarges the space in which a method can be chosen: instead of a single formula for a given number of points and order, one obtains a one-parameter family, and the parameter can be fixed by requiring a large stability region, a small error constant or a compromise between the two.

In this paper we add the parameter ρ to the two-point fractional block BDF and obtain a superclass of fractional block BDFs for Caputo-type FDEs. The coefficients are derived from Taylor-type expansions of the solution and of the Caputo derivative within a fractional block multistep discretization. We do not assume an algebraic order in advance. Instead, the order is obtained from the local truncation error, because order conditions for fractional methods differ from those of their integer-order counterparts and should not be inferred from them.

The main contributions are the derivation of the parameterized two-point scheme, an analysis of its order, consistency, zero-stability, convergence and absolute stability for several fractional orders, and a numerical comparison with two existing fractional block BDFs. The paper is organized as follows. Section 2 derives the method, and Section 3 gives its order, error constants and consistency. Section 4 studies zero-stability and absolute stability, and Section 5 discusses convergence. Section 6 describes the implementation, Sections 7 and 8 present and discuss the numerical results, and Section 9 concludes.

2 Derivation of the 2SFBBDF3 Method

This section derives the two-point superclass fractional block backward differentiation formula. The method uses a uniform mesh $t_j = t_0 + jh$ with a fixed step size h. In each step the values v_{n-1} and v_n, which are already known, are used to compute the two new values v_{n+1} and v_{n+2} simultaneously; the next block then starts from v_{n+1} and v_{n+2}. The grid points involved in the construction are shown in Figure 1.

The construction follows the general fractional linear multistep framework of Galeone and Garrappa [23] and the classical superclass of block BDF for stiff ODEs [35]. The proposed parameterized block scheme

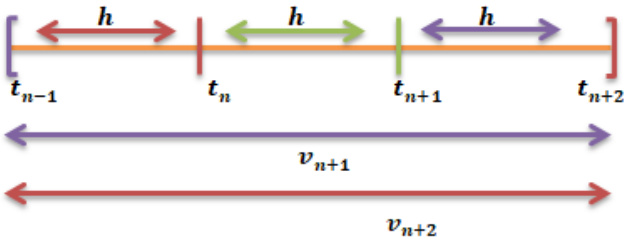


Figure 1. Interpolation points of the 2SFBBDF3 method.

is

$$\sum_{j=0}^{3} \sigma_{j,i}\, v_{n+j-1} = h^{\alpha}\gamma_{i,i}\big(f_{n+i} - \rho f_{n+i-1}\big), \qquad -1 < \rho < 1, \quad i = 1, 2, \tag{5}$$

where v_{n+j-1} is the approximation at t_{n+j-1}, $f_{n+i} = f(t_{n+i}, v_{n+i})$, $0 < \alpha \leq 1$ is the fractional order, ρ is the stability-control parameter, and $\sigma_{j,i}$, $\gamma_{i,i}$ are real coefficients to be determined. The factor h^{α} comes from the discretization of the fractional derivative. Setting $\rho = 0$ gives the 2FBBDF3 of [36],

$$\sum_{j=0}^{3} \sigma_{j,i}\, v_{n+j-1} = h^{\alpha}\gamma_{i,i} f_{n+i}, \qquad i = 1, 2, \tag{6}$$

so that (5) is a one-parameter superclass of (6). The extra term $-\rho f_{n+i-1}$ on the right-hand side is the only structural difference between the two schemes. It uses a derivative value that is already known (for $i = 1$) or that is computed in the same block (for $i = 2$), so the cost of a step increases only slightly. Each admissible value of ρ gives a different member of the family, and the parameter can be chosen to improve the stability of the scheme.

Definition 1. The linear difference operator L_h associated with (5) is

$$L_h[v(t), t, \alpha] = \sum_{j=0}^{3} \sigma_{j,i} v_{n+j-1} - h^{\alpha}\gamma_{i,i}\big({}_{t_0}^{C}D_t^{\alpha} v_{n+i} - \rho\, {}_{t_0}^{C}D_t^{\alpha} v_{n+i-1}\big). \tag{7}$$

To determine the coefficients, the solution values appearing in (7) are expanded as Taylor series about t_n, the last known point of the previous block:

$$v_{n-1} = \sum_{i=0}^{\infty} \frac{(-h)^i}{i!} v_n^{(i)}, \qquad v_{n+1} = \sum_{i=0}^{\infty} \frac{h^i}{i!} v_n^{(i)}, \qquad v_{n+2} = \sum_{i=0}^{\infty} \frac{(2h)^i}{i!} v_n^{(i)}, \tag{8}$$

and the Caputo derivatives at t_{n+1} and t_{n+2} are represented as

$$\begin{aligned} {}_{t_0}^{C}D_t^{\alpha} v_{n+1} &= \sum_{i=1}^{\infty} \frac{h^{i-\alpha}}{\Gamma(i+1-\alpha)} v_n^{(i)}, \\ {}_{t_0}^{C}D_t^{\alpha} v_{n+2} &= \sum_{i=1}^{\infty} \frac{(2h)^{i-\alpha}}{\Gamma(i+1-\alpha)} v_n^{(i)}. \end{aligned} \tag{9}$$

The expansions (9) follow from the formula ${}_{t_0}^{C}D_t^{\alpha}(t-t_n)^i = \Gamma(i+1)(t-t_n)^{i-\alpha}/\Gamma(i+1-\alpha)$, applied term by term to the Taylor polynomial of v about t_n with t_n taken as the lower terminal of the derivative. They reduce to the ordinary Taylor expansions of $v'(t_{n+1})$ and $v'(t_{n+2})$ when $\alpha = 1$, which is why the scheme becomes a classical block BDF in this limit. For $0 < \alpha < 1$ the expansions describe only the local part of the derivative; the contribution of the history on $[t_0, t_n]$ is not represented in (9), and this has to be kept in mind when the results of the following sections are interpreted. Substituting these expansions into (7) and collecting terms with the same power of h gives a linear system for the unknown coefficients. For the first point, $i = 1$ is substituted into (7) together with (8)-(9). Collecting the coefficients of $v_n, v_n', v_n'', \ldots$ gives

$$L_h[v(t), t, \alpha] = \Psi_{0,1} v_n + \sum_{k=1}^{m} h^k \Psi_{k,1} v_n^{(k)} + h^{m+1} R_{m+1}, \tag{10}$$

$m = 1, 2, 3, \ldots$, where R_{m+1} is the remainder after truncation at the m-th derivative, and

$$\left.\begin{aligned} \Psi_{0,1} &= \sigma_{0,1} + \sigma_{1,1} + \sigma_{2,1} + \sigma_{3,1} = 0, \\ \Psi_{1,1} &= -\sigma_{0,1} + \sigma_{2,1} + 2\sigma_{3,1} - \frac{1-\rho}{\Gamma(2-\alpha)}\gamma_{1,1} = 0, \\ \Psi_{2,1} &= \tfrac{1}{2}\sigma_{0,1} + \tfrac{1}{2}\sigma_{2,1} + 2\sigma_{3,1} - \frac{1}{\Gamma(3-\alpha)}\gamma_{1,1} = 0, \\ \Psi_{3,1} &= -\tfrac{1}{6}\sigma_{0,1} + \tfrac{1}{6}\sigma_{2,1} + \tfrac{4}{3}\sigma_{3,1} - \frac{1}{\Gamma(4-\alpha)}\gamma_{1,1} = 0. \end{aligned}\right\} \tag{11}$$

The coefficient of v_{n+1} is normalized to one ($\sigma_{2,1} = 1$), which leaves four unknowns, $\sigma_{0,1}$, $\sigma_{1,1}$, $\sigma_{3,1}$ and $\gamma_{1,1}$, and four equations. Requiring in addition $\Psi_{4,1} = 0$ would over-determine the system, so the first non-vanishing term $\Psi_{4,1}$ determines the error constant (Section 3). Solving (11) gives the coefficients listed in Table 1, with $D_1 = \alpha^2\rho - \alpha^2 - 5\alpha\rho + 6\alpha + 6\rho - 6$.

Substituting these values into (7) gives the first-point formula

Table 1. Coefficients of the first-point formula.

Coefficient	Expression
$\sigma_{0,1}$	$-\dfrac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 2\alpha + 6\rho}{3D_1}$
$\sigma_{1,1}$	$-\dfrac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 9\alpha + 6\rho - 12}{2D_1}$
$\sigma_{2,1}$	1
$\sigma_{3,1}$	$-\dfrac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 5\alpha + 6\rho}{6D_1}$
$\gamma_{1,1}$	$-\dfrac{\Gamma(4-\alpha)}{D_1}$

$$\begin{aligned} v_{n+1} = {} & \frac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 2\alpha + 6\rho}{3D_1} v_{n-1} \\ & + \frac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 9\alpha + 6\rho - 12}{2D_1} v_n \\ & + \frac{\alpha^2\rho - \alpha^2 - 5\alpha\rho + 5\alpha + 6\rho}{6D_1} v_{n+2} \\ & - \frac{\Gamma(4-\alpha)}{D_1} h^{\alpha} f_{n+1} + \frac{\rho\,\Gamma(4-\alpha)}{D_1} h^{\alpha} f_n. \end{aligned} \tag{12}$$

For the second point, $i = 2$ is substituted into (7) and the same procedure gives

$$L_h[v(t), t, \alpha] = \Psi_{0,2} v_n + \sum_{k=1}^{m} h^k \Psi_{k,2} v_n^{(k)} + h^{m+1} R_{m+1}, \tag{13}$$

$m = 1, 2, 3, \ldots$, where the coefficients $\Psi_{k,2}$ are

$$\left.\begin{aligned} \Psi_{0,2} &= \sigma_{0,2} + \sigma_{1,2} + \sigma_{2,2} + \sigma_{3,2} = 0, \\ \Psi_{1,2} &= -\sigma_{0,2} + \sigma_{2,2} + 2\sigma_{3,2} - \frac{2^{1-\alpha} - \rho}{\Gamma(2-\alpha)}\gamma_{2,2} = 0, \\ \Psi_{2,2} &= \tfrac{1}{2}\sigma_{0,2} + \tfrac{1}{2}\sigma_{2,2} + 2\sigma_{3,2} - \frac{2^{2-\alpha} - \rho}{\Gamma(3-\alpha)}\gamma_{2,2} = 0, \\ \Psi_{3,2} &= -\tfrac{1}{6}\sigma_{0,2} + \tfrac{1}{6}\sigma_{2,2} + \tfrac{4}{3}\sigma_{3,2} - \frac{2^{3-\alpha} - \rho}{\Gamma(4-\alpha)}\gamma_{2,2} = 0. \end{aligned}\right\} \tag{14}$$

In contrast to the first point, the ρ-term now involves f_{n+1}, so it contributes the factor $1^{k-\alpha} = 1$ to every equation. With $\sigma_{3,2} = 1$, the solution of (14) is given in Table 2, where $D_2 = \alpha^2\rho\,2^{\alpha} - 5\alpha\rho\,2^{\alpha} - 2\alpha^2 + 10\alpha + 36$ and $G_2 = -2^{1-\alpha}\alpha^2 + \alpha^2\rho + 10\cdot 2^{-\alpha}\alpha - 5\alpha\rho + 36\cdot 2^{-\alpha}$.

Substituting into (7) gives the second-point formula

$$\begin{aligned} v_{n+2} = {} & -\sigma_{0,2} v_{n-1} - \sigma_{1,2} v_n - \sigma_{2,2} v_{n+1} \\ & + \frac{6\Gamma(4-\alpha)}{G_2} h^{\alpha} f_{n+2} - \frac{6\rho\,\Gamma(4-\alpha)}{G_2} h^{\alpha} f_{n+1}, \end{aligned} \tag{15}$$

Table 2. Coefficients of the second-point formula.

Coefficient	Expression
$\sigma_{0,2}$	$\dfrac{\alpha\left(2^{1+\alpha}\alpha\rho - 4\cdot 2^{\alpha}\rho - 4\alpha - 4\right)}{D_2}$
$\sigma_{1,2}$	$\dfrac{3\left(\alpha^2\rho 2^{\alpha} - 9\alpha\rho 2^{\alpha} + 12\cdot 2^{\alpha}\rho - 2\alpha^2 + 26\alpha - 12\right)}{D_2}$
$\sigma_{2,2}$	$-\dfrac{6\left(\alpha^2\rho 2^{\alpha} - 6\alpha\rho 2^{\alpha} + 6\cdot 2^{\alpha}\rho - 2\alpha^2 + 14\alpha\right)}{D_2}$
$\sigma_{3,2}$	1
$\gamma_{2,2}$	$\dfrac{6\,\Gamma(4-\alpha)}{G_2}$

with $\sigma_{j,2}$ from Table 2. Together, (12) and (15) form the general two-point superclass. The two formulas are coupled: the first contains v_{n+2} and the second contains v_{n+1}, so the pair has to be solved as one implicit system in each step.

Each choice of ρ in the interval $(-1, 1)$ required for stiff stability gives a member of the family, and $\rho = 0$ reduces (12) and (15) to the 2FBBDF3 of [36]. The value of ρ is a design choice. For ODEs, superclass block BDFs constructed in this way have been reported to be A-stable with smaller error constants than the conventional BBDF [35], and the choice of the parameter for stiff stability is discussed in [37, 38]. The stability properties that guide such a choice for block BDFs, namely A(α)-stability, stiff stability and optimal stability regions, are analyzed in [39–41], and a variable step-size formulation of the block BDF for stiff problems is given in [42]. Following these stability arguments, we take $\rho = -3/4$. Substituting $\rho = -3/4$ into (12) and (15) gives the 2SFBBDF3 used in this study:

$$\left.\begin{aligned} v_{n+1} &= \delta_1 v_{n-1} + \delta_2 v_n + \delta_3 v_{n+2} \\ &\quad + \delta_4 h^{\alpha} f_{n+1} + \delta_5 h^{\alpha} f_n, \\ v_{n+2} &= \tau_1 v_{n-1} + \tau_2 v_n + \tau_3 v_{n+1} \\ &\quad + \tau_4 h^{\alpha} f_{n+2} + \tau_5 h^{\alpha} f_{n+1}, \end{aligned}\right\} \tag{16}$$

where, with $W_\alpha = 7\alpha^2 - 39\alpha + 42$, $B_\alpha = 3\alpha^2 2^{\alpha} - 15\alpha 2^{\alpha} + 8\alpha^2 - 40\alpha - 144$ and $Q_\alpha = 8\cdot 2^{-\alpha}\alpha^2 - 40\cdot 2^{-\alpha}\alpha + 3\alpha^2 - 144\cdot 2^{-\alpha} - 15\alpha$ $(= 2^{-\alpha}B_\alpha)$,

$$\begin{aligned} \delta_1 &= \frac{7\alpha^2 - 23\alpha + 18}{3W_\alpha}, \quad \delta_2 = \frac{7\alpha^2 - 51\alpha + 66}{2W_\alpha}, \\ \delta_3 &= \frac{7\alpha^2 - 35\alpha + 18}{6W_\alpha}, \\ \delta_4 &= \frac{4\Gamma(4-\alpha)}{W_\alpha}, \quad \delta_5 = \frac{3\Gamma(4-\alpha)}{W_\alpha}, \\ \tau_1 &= -\frac{4\alpha N_1}{B_\alpha}, \quad \tau_2 = -\frac{3N_2}{B_\alpha}, \quad \tau_3 = \frac{6N_3}{B_\alpha}, \\ \tau_4 &= -\frac{24\Gamma(4-\alpha)}{Q_\alpha}, \quad \tau_5 = -\frac{18\Gamma(4-\alpha)}{Q_\alpha}, \end{aligned} \tag{17}$$

with

$$\begin{aligned} N_1 &= 3\cdot 2^{\alpha-1}\alpha - 3\cdot 2^{\alpha} + 4\alpha + 4, \\ N_2 &= 3\alpha^2 2^{\alpha} - 27\alpha 2^{\alpha} + 8\alpha^2 + 36\cdot 2^{\alpha} - 104\alpha + 48, \\ N_3 &= 3\alpha^2 2^{\alpha} - 18\alpha 2^{\alpha} + 8\alpha^2 + 18\cdot 2^{\alpha} - 56\alpha. \end{aligned}$$

For $\alpha = 1$ the coefficients reduce to $\delta_1 = 1/15$, $\delta_2 = 11/10$, $\delta_3 = -1/6$, $\delta_4 = 4/5$, $\delta_5 = 3/5$ and $\tau_1 = 1/10$, $\tau_2 = -9/25$, $\tau_3 = 63/50$, $\tau_4 = 12/25$, $\tau_5 = 9/25$, which gives a classical third-order block method for ODEs. The following sections study the order, consistency, zero-stability and absolute stability of (16) as functions of α, so that the analysis can be repeated for any fractional order of interest.

3 Order, Error Constant and Consistency

The order and the error constant describe how accurately a method approximates the differential equation locally, and consistency is the minimal requirement for convergence. In this section the order of 2SFBBDF3 is obtained from the order conditions, the error constants of the two points of the block are computed for several values of α, and consistency is deduced from the behavior of the local truncation

error. These results are used again in the convergence analysis of Section 5.

Definition 2 (Order and error constant). The block method (16) is of order p if $\Psi_0 = \Psi_1 = \cdots = \Psi_p = 0$ and $\Psi_{p+1} \neq 0$; Ψ_{p+1} is then the error constant [36]. For each point,

$$\begin{aligned}\Psi_0(m,\alpha) &= \sum_{i=0}^{m} \sigma_i,\\ \Psi_k(m,\alpha) &= \sum_{i=0}^{m} \frac{(i-1)^k}{k!}\sigma_i \\ &\quad - \frac{1}{\Gamma(1+k-\alpha)} \sum_{i=0}^{m} (i-1)^{k-\alpha}\gamma_i,\end{aligned} \tag{18}$$

with $m = 3$ and $k = 1, 2, 3, \ldots$.

For the first point, the term $-\rho f_n$ is located at t_n, where the series (9) contains no contribution for $k \geq 2$; consistently with (11), its contribution is retained only in the condition $k = 1$, where $(i-1)^{k-\alpha}$ is evaluated as 1 (it is the exact value $0^0 = 1$ when $\alpha = 1$). This is a convention of the construction for $\alpha < 1$, and the order stated below for $\alpha < 1$ holds under it. In (18), σ_i and γ_i denote the coefficients of the point under consideration, and the factor $(i-1)$ is the position of the grid point t_{n+i-1} relative to t_n in units of h. The quantity Ψ_k is the coefficient of $h^k v_n^{(k)}$ in the expansion of the local truncation error, so the first index k with $\Psi_k \neq 0$ determines both the order and the leading error term. Substituting the coefficients of (16) into (18) gives, for the two points of the block,

$$\begin{aligned}\Psi_k(3,\alpha) &= \begin{bmatrix}0\\0\end{bmatrix}, \quad k = 0,1,2,3,\\ \Psi_4(3,\alpha) &= \begin{bmatrix}\Psi_4^{(1)}\\ \Psi_4^{(2)}\end{bmatrix} \neq \begin{bmatrix}0\\0\end{bmatrix},\end{aligned} \tag{19}$$

where $\Psi_4^{(1)}$ and $\Psi_4^{(2)}$ are the error constants of the first and second points. Their values for several α are listed in Table 3.

Table 3 shows that $\Psi_4^{(1)}$ and $\Psi_4^{(2)}$ are non-zero for all orders considered. Since $\Psi_0 = \Psi_1 = \Psi_2 = \Psi_3 = 0$ and $\Psi_4 \neq 0$, the 2SFBBDF3 method is of order three for $\alpha = 0.75, 0.85, 0.95$ and 1.00. The error constant of the second point varies moderately with α (from -0.059 to -0.090), whereas that of the first point changes sign between $\alpha = 0.75$ and $\alpha = 0.85$ and is very small near $\alpha = 0.85$, so for these orders the leading error term of the first point is not a reliable indicator of its accuracy and the higher-order terms matter. In every case the error constant of the second point is larger in magnitude than that of the first point, which is expected because t_{n+2} lies further from the expansion point t_n than t_{n+1}. The values for $\alpha < 1$ in Table 3 are computed from (18) with the convention stated above, and they depend on that convention.

Definition 3 (Consistency). An FLMM is consistent if its local truncation error tends to zero as $h \to 0$ [43]. If T_{n+i} denotes the local truncation error at the i-th point, consistency means $\lim_{h\to 0} T_{n+i} = 0$. A method of order $p \geq 1$ satisfies $T_{n+i} = O(h^{p+1})$ and is therefore consistent.

Lemma 1. The 2SFBBDF3 method is consistent for the fractional orders considered.

Proof. Let $T_{n+1}^{(1)}$ and $T_{n+2}^{(2)}$ be the local truncation errors of the two points. By Definition 2 and the order conditions obtained from (18), all terms up to order three in the truncation-error expansion vanish, so

$$T_{n+1}^{(1)} = \Psi_4^{(1)} h^4 + O(h^5), \qquad T_{n+2}^{(2)} = \Psi_4^{(2)} h^4 + O(h^5).$$

The constants $\Psi_4^{(1)}$ and $\Psi_4^{(2)}$ are finite for $\alpha = 0.75$, 0.85, 0.95 and 1.00 (Table 3). Hence $\lim_{h\to 0} T_{n+1}^{(1)} = \lim_{h\to 0} T_{n+2}^{(2)} = 0$, and the method is consistent. □

Lemma 1 is stated for the four values of α in Table 3. The coefficients (17) depend continuously on α, and the denominators W_α and B_α do not vanish on $(0, 1]$, so the same argument applies to any other order in this interval for which the error constants are finite.

4 Stability Analysis

This section studies the stability of 2SFBBDF3, which determines whether the method is suitable for FDEs with stiff or rapidly varying solutions. Stability ensures that perturbations introduced during the computation, for example by rounding or by the starting values, do not grow without bound as the integration proceeds. We apply the method to the test equation (3), derive the corresponding stability polynomial, and examine the stability region for several fractional orders with $\rho = -3/4$.

Definition 4 (Root condition). The first characteristic polynomial of an FLMM satisfies the root condition if all its roots r_i, $i = 1, \ldots, k$, satisfy $|r_i| \leq 1$ and every root with $|r_i| = 1$ is simple [44]. The same condition is used for block BDFs in [45].

Definition 5 (Zero-stability). An FLMM is zero-stable if its first characteristic polynomial

Table 3. Error constants of 2SFBBDF3 for different fractional orders α.

Error constant	$\alpha = 0.75$	$\alpha = 0.85$	$\alpha = 0.95$	$\alpha = 1.00$
$\Psi_4^{(1)}$	-0.0072746	0.00032599	0.010316	$\frac{1}{60} \approx 0.016667$
$\Psi_4^{(2)}$	-0.059280	-0.070803	-0.083332	$-\frac{9}{100}$

satisfies the root condition. This ensures that perturbations of the starting values do not grow without bound as $h \to 0$ [44].

Definition 6 (Absolute stability). For a given $H = h^\alpha \lambda$, an FLMM is absolutely stable if all roots r_i of its stability polynomial satisfy $|r_i| < 1$, $i = 1, \ldots, k$; the set of such H is the region of absolute stability [44].

Definition 7 (A-stability). An FLMM is A-stable if its region of absolute stability contains the whole left half-plane $\mathrm{Re}(H) < 0$ [43].

Applying (16) to (3) gives the linear difference system

$$\left.\begin{aligned} v_{n+1} &= \delta_1 v_{n-1} + \delta_2 v_n + \delta_3 v_{n+2} \\ &\quad + \delta_4 h^\alpha \lambda v_{n+1} + \delta_5 h^\alpha \lambda v_n, \\ v_{n+2} &= \tau_1 v_{n-1} + \tau_2 v_n + \tau_3 v_{n+1} \\ &\quad + \tau_4 h^\alpha \lambda v_{n+2} + \tau_5 h^\alpha \lambda v_{n+1}. \end{aligned}\right\} \tag{20}$$

With $H = h^\alpha \lambda$, (20) can be written as

$$A_0 V_m = A_1 V_{m-1},$$
$$A_0 = \begin{bmatrix} 1 - \delta_4 H & -\delta_3 \\ -\tau_3 - \tau_5 H & 1 - \tau_4 H \end{bmatrix}, \quad A_1 = \begin{bmatrix} \delta_1 & \delta_2 + \delta_5 H \\ \tau_1 & \tau_2 \end{bmatrix}, \tag{21}$$

where $V_m = (v_{n+1}, v_{n+2})^\mathsf{T}$ and $V_{m-1} = (v_{n-1}, v_n)^\mathsf{T}$. The matrix A_0 contains the coefficients of the two new values of the block and A_1 those of the back values, so the block method can be viewed as a one-step recursion acting on vectors of two values. The stability polynomial is

$$\pi(r, H) = \det\left(rA_0 - A_1\right) = 0. \tag{22}$$

The roots r of $\pi(r, H)$ are the amplification factors of the recursion (21). Setting $H = 0$ in (22) gives the first characteristic polynomial, whose roots decide zero-stability. These roots were computed for selected values of α and are listed in Table 4.

For each α, $\pi(r, 0)$ has a simple root $r = 1$ and a second root of modulus less than one (Table 4). The root condition holds, and 2SFBBDF3 is zero-stable for the orders considered. The second root lies between -0.11 and -0.05 and moves away from zero as α increases, but it remains far from the unit circle, so zero-stability is not marginal for any of the four orders.

The boundary of the stability region is obtained by substituting $r = e^{i\theta}$, $0 \le \theta \le 2\pi$, into (22) and solving for H. The regions for $\alpha = 0.75, 0.85, 0.95$ and 1.00, plotted with Maple 18, are shown in Figures 2-5. In each figure the boundary locus is a closed curve that passes through the origin and lies essentially in the right half-plane; inside the curve at least one root has modulus greater than one, and outside it the method is absolutely stable. The right-hand crossing of the positive real axis and the smallest real part reached by each curve give the instability intervals listed in Table 5; for $\alpha = 1.00$ the left end -0.111 is the smallest real part of the boundary, which is attained away from the real axis, and the curve crosses the real axis itself at 0 and 16.

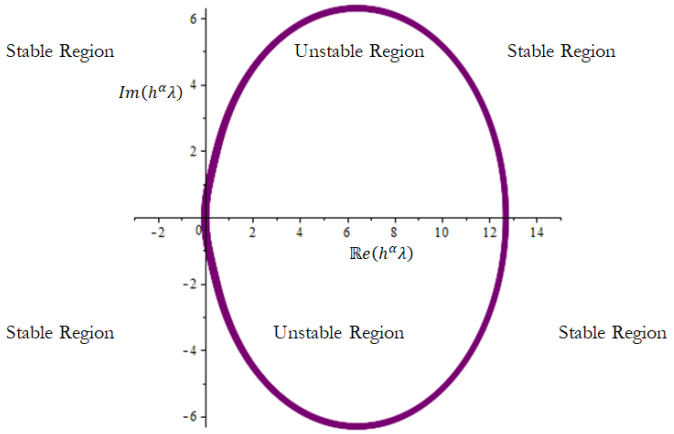


Figure 2. Stability region of 2SFBBDF3 for $\alpha = 0.75$.

Table 5 shows that the instability interval on the positive real axis widens as α increases, its right end moving from about 12.7 for $\alpha = 0.75$ to 16 for $\alpha = 1$. The left ends of the intervals for $\alpha = 0.75$, 0.85 and 0.95 are of order 10^{-9}-10^{-10} and are zero to within the accuracy of the computation, which is consistent with the simple root $r = 1$ of $\pi(r, 0)$. The colors in the third column refer to the curves in Figure 6.

Figure 6 superimposes the boundaries for the four orders. For each α the unstable region lies inside the closed curve and the stable region outside it, and the boundaries grow as α increases, the largest being the

Table 4. First characteristic polynomials $\pi(r,0)$ and their roots for different fractional orders α.

α	$\pi(r,0)$	Roots
0.75	$\frac{16723}{16115}r^2 - \frac{35337}{35894}r - \frac{4173}{78371}$	$r_1 = 1, r_2 = -0.05131083944$
0.85	$\frac{41329}{38157}r^2 - \frac{16371}{16409}r - \frac{13114}{153477}$	$r_1 = 1, r_2 = -0.07888804683$
0.95	$\frac{19139}{16543}r^2 - \frac{68867}{66258}r - \frac{7818}{66509}$	$r_1 = 1, r_2 = -0.1016038771$
1.00	$\frac{121}{100}r^2 - \frac{269}{250}r - \frac{67}{500}$	$r_1 = 1, r_2 = -0.1107438017$

Table 5. Instability intervals of 2SFBBDF3 along the real axis (right end: crossing of the positive real axis; left end: smallest real part of the boundary) and stability properties.

α	**Interval of instability**	**Curve color in Figure 6**	**Stability property**
0.75	$(4.751909 \times 10^{-10},\ 12.701)$	Purple	A-stable
0.85	$(2.831971 \times 10^{-9},\ 14.054)$	Green	A-stable
0.95	$(-1.049156 \times 10^{-9},\ 15.365)$	Blue	A-stable
1.00	$(-0.111,\ 16)$	Orange	A(θ)-stable, $\theta \approx 86.8°$

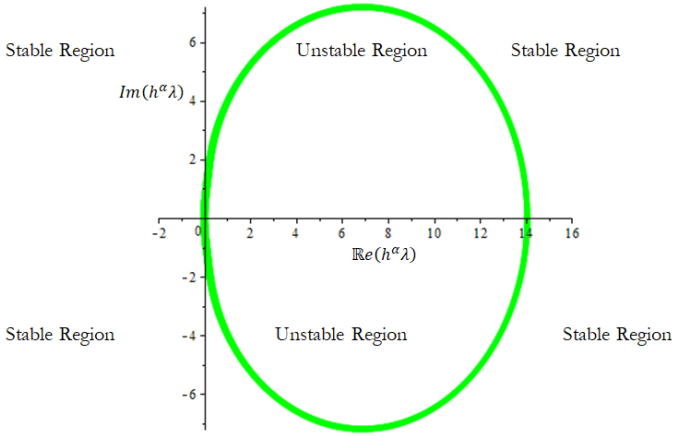


Figure 3. Stability region of 2SFBBDF3 for $\alpha = 0.85$.

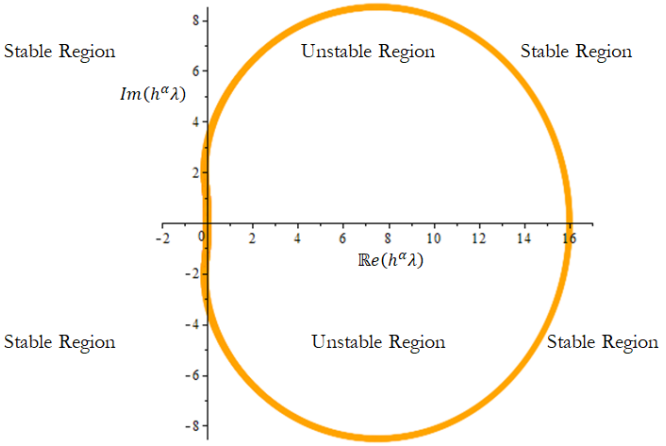


Figure 5. Stability region of 2SFBBDF3 for $\alpha = 1.00$.

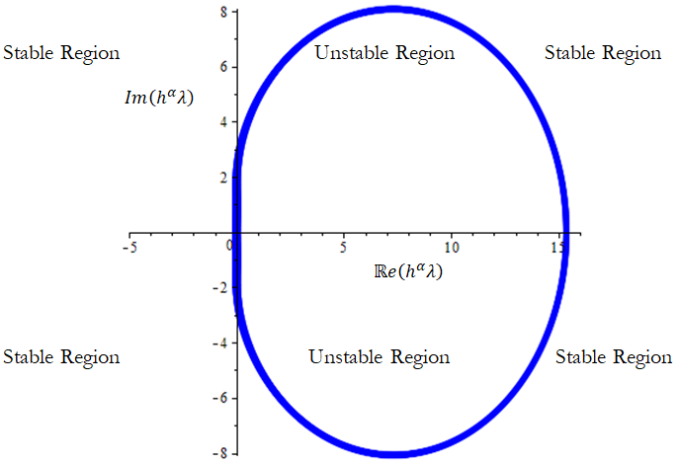


Figure 4. Stability region of 2SFBBDF3 for $\alpha = 0.95$.

one for $\alpha = 1.00$. For $\alpha = 0.75, 0.85$ and 0.95 the whole left half-plane $\mathrm{Re}(h^\alpha\lambda) < 0$ lies in the stable region, so the method is A-stable for these orders. For $\alpha = 1.00$ the boundary crosses slightly into the left half-plane near the imaginary axis (Table 5); the method is then A(θ)-stable with θ close to $90°$ rather than A-stable. In all four cases the negative real axis is contained in the stability region. For stiff problems the dominant eigenvalues usually lie on or near the negative real axis, where all four variants are stable. The loss of strict A-stability for $\alpha = 1$ is therefore of little consequence for the test problems of Section 7, whose eigenvalues are real and negative, but it should be kept in mind for problems with eigenvalues close to the imaginary axis.

5 Convergence Analysis

Convergence means that the numerical solution approaches the exact solution at every grid point as the step size tends to zero. For linear multistep methods it follows from consistency together with zero-stability. In this section we state the result in the form used

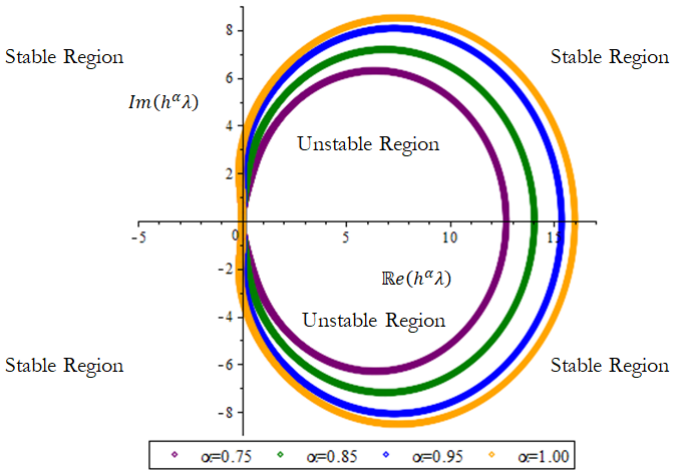


Figure 6. Superimposed boundaries of the absolute stability regions of 2SFBBDF3 for $\alpha = 0.75, 0.85, 0.95$ and 1.00.

in [46, 47] and apply it to 2SFBBDF3.

Assumption 1 (Lipschitz condition) [46, 47]. There is a constant $L > 0$ such that

$$\left|f(t,v) - f(t,v^*)\right| \le L\,|v - v^*| \tag{23}$$

for all $(t,v),(t,v^*)$ in $\mathcal{R} = \{(t,v) \in \mathbb{R}^2 : a \le t \le b,\ -\infty < v < \infty\}$, where a and b are real constants.

Theorem 1 [46, 47]. Suppose an FLMM is consistent and zero-stable and is applied to problem (2) with f satisfying Assumption 1. Then there exist constants $K > 0$ and $0 < \mu < 1$, depending on α but not on h, such that

$$|v(t_n) - v_n| \le K\,h^{\alpha-1}h^{\mu}, \tag{24}$$

and consequently $\lim_{h\to 0} v_n = v(t_n)$.

Proof for 2SFBBDF3. The argument compares the exact solution, which satisfies the method up to the local truncation error, with the numerical solution, which satisfies it exactly, and bounds the difference with the Lipschitz condition. Write the exact solution in the form of (16) with the local truncation errors added:

$$\left.\begin{aligned} v(t_{n+1}) &= \delta_1 v(t_{n-1}) + \delta_2 v(t_n) + \delta_3 v(t_{n+2}) \\ &\quad + \delta_4 h^{\alpha} f(t_{n+1}, v(t_{n+1})) \\ &\quad + \delta_5 h^{\alpha} f(t_n, v(t_n)) \\ &\quad + \Psi_4^{(1)} h^4 v^{(4)}(\xi_1), \\ v(t_{n+2}) &= \tau_1 v(t_{n-1}) + \tau_2 v(t_n) + \tau_3 v(t_{n+1}) \\ &\quad + \tau_4 h^{\alpha} f(t_{n+2}, v(t_{n+2})) \\ &\quad + \tau_5 h^{\alpha} f(t_{n+1}, v(t_{n+1})) \\ &\quad + \Psi_4^{(2)} h^4 v^{(4)}(\xi_2), \end{aligned}\right\} \tag{25}$$

while the numerical solution v_{n+j} satisfies (16) exactly. Subtracting (16) from (25), writing $d_k = v(t_k) - v_k$, applying (23) and setting $M_4 = \max_{a\le t\le b} |v^{(4)}(t)|$ gives

$$\left.\begin{aligned} \left(1 - |\delta_4| h^{\alpha} L\right)|d_{n+1}| &\le |\delta_1||d_{n-1}| \\ &\quad + \left(|\delta_2| + |\delta_5| h^{\alpha} L\right)|d_n| \\ &\quad + |\delta_3||d_{n+2}| + |\Psi_4^{(1)}| h^4 M_4, \\ \left(1 - |\tau_4| h^{\alpha} L\right)|d_{n+2}| &\le |\tau_1||d_{n-1}| + |\tau_2||d_n| \\ &\quad + \left(|\tau_3| + |\tau_5| h^{\alpha} L\right)|d_{n+1}| \\ &\quad + |\Psi_4^{(2)}| h^4 M_4. \end{aligned}\right\} \tag{26}$$

The inequalities (26) show how the error is built up from one block to the next: the local truncation errors enter with the factor h^4, and the Lipschitz terms are multiplied by h^{α}, so for small h the coefficients on the left-hand side remain close to one. The method is consistent (Lemma 1) and zero-stable (Table 4). By Theorem 1 the global error tends to zero as $h \to 0$, so $\lim_{h\to 0} v_n = v(t_n)$ and 2SFBBDF3 is convergent. □

6 Implementation

The numerical implementation of 2SFBBDF3 is carried out in the classical limiting case $\alpha = 1$. The consistency, zero-stability, convergence and absolute stability results of the previous sections were obtained for several fractional orders, but the experiments reported here are restricted to $\alpha = 1$. In this case the exact solutions of the test problems are known in closed form, and the results can be compared directly with those of existing block BDFs. The restriction does not affect the fractional-order analysis; it does mean that the experiments test the method as an integrator for stiff ODEs, a point we return to in Section 8.

For a fair comparison, all methods are applied to the same problems with the same constant step sizes. Accuracy is measured by the maximum absolute error, obtained by comparing the exact and numerical solutions at every grid point and taking the largest difference over all points and components. Efficiency is measured by the CPU time and the total number of steps. Because 2SFBBDF3 is implicit, a nonlinear algebraic system has to be solved in each step, and Newton's method is used for this purpose. Let $v_i(t_n)$ and $v_i^{(n)}$ be the exact and approximate values of the i-th component at t_n. The absolute error and the maximum error are

$$\begin{aligned} E_i^{(n)} &= \left|v_i(t_n) - v_i^{(n)}\right|, \\ \mathrm{MAXE} &= \max_{1\le n\le M_s} \max_{1\le i\le N} \left|v_i(t_n) - v_i^{(n)}\right|, \end{aligned} \tag{27}$$

where M_s is the number of steps and N the number of equations. In residual form, the 2SFBBDF3 equations are

$$\left.\begin{aligned} F_1 &= v_{n+1} - \delta_3 v_{n+2} - \delta_4 h^\alpha f_{n+1} \\ &\quad - \delta_5 h^\alpha f_n - \Omega_1 = 0, \\ F_2 &= v_{n+2} - \tau_3 v_{n+1} - \tau_4 h^\alpha f_{n+2} \\ &\quad - \tau_5 h^\alpha f_{n+1} - \Omega_2 = 0, \end{aligned}\right\} \tag{28}$$

with the back values

$$\Omega_1 = \delta_1 v_{n-1} + \delta_2 v_n, \qquad \Omega_2 = \tau_1 v_{n-1} + \tau_2 v_n. \tag{29}$$

The back values depend only on the previous block and are computed once per step. Let $v_{n+j}^{(k)}$, $j = 1, 2$, be the k-th Newton iterate and $e_{n+j}^{(k+1)} = v_{n+j}^{(k+1)} - v_{n+j}^{(k)}$. Instead of forming the inverse Jacobian, each iteration solves the linear system

$$J^{(k)} \begin{bmatrix} e_{n+1}^{(k+1)} \\ e_{n+2}^{(k+1)} \end{bmatrix} = - \begin{bmatrix} F_1^{(k)} \\ F_2^{(k)} \end{bmatrix}, \tag{30}$$

where $F_j^{(k)} = F_j\big(v_{n+1}^{(k)}, v_{n+2}^{(k)}\big)$ and

$$J^{(k)} = \begin{bmatrix} I - \delta_4 h^\alpha J_{n+1} & -\delta_3 I \\ -\tau_3 I - \tau_5 h^\alpha J_{n+1} & I - \tau_4 h^\alpha J_{n+2} \end{bmatrix}, \tag{31}$$

with $J_{n+j} = \partial f / \partial v$ evaluated at $(t_{n+j}, v_{n+j}^{(k)})$ and I is the identity matrix. For a system of N equations, $J^{(k)}$ is a $2N \times 2N$ matrix, and solving (30) as a linear system is cheaper and more stable than forming the inverse Jacobian explicitly. The Newton iteration is repeated until a prescribed convergence criterion is satisfied. The algorithm was implemented in the C programming language and used to compute the numerical solutions, maximum errors and CPU times reported in Section 7. One step of the method can be summarized as follows [48].

Because each step produces two solution values, the number of steps needed to integrate over an interval of length $b - a$ with step size h is $(b - a)/(2h)$. This is the quantity reported as TS in Tables 6-8; for example, Problem 1 on $[0, 10]$ with $h = 10^{-2}$ requires 500 steps.

Algorithm 1: One step of 2SFBBDF3

Data: back values v_{n-1}, v_n; step size h; tolerance
Result: new block values v_{n+1}, v_{n+2}
compute the back values Ω_1, Ω_2 from (29);
choose initial guesses $v_{n+1}^{(0)}$, $v_{n+2}^{(0)}$;
repeat
  evaluate $F_1^{(k)}$, $F_2^{(k)}$ from (28) and $J^{(k)}$ from (31);
  solve (30) for $e_{n+1}^{(k+1)}$, $e_{n+2}^{(k+1)}$;
  update $v_{n+j}^{(k+1)} = v_{n+j}^{(k)} + e_{n+j}^{(k+1)}$, $j = 1, 2$;
until *the correction $e^{(k+1)}$ is smaller than the tolerance*;
accept v_{n+1}, v_{n+2}, update the maximum error, and move to the next block;

7 Numerical Experiments

To assess the accuracy and efficiency of 2SFBBDF3, we apply it to three stiff systems that are frequently used as benchmarks for stiff solvers. Two of them are linear with constant coefficients, and the third is a nonlinear chemical-kinetics model whose stiffness is controlled by a parameter. The systems are written with Caputo derivatives, but the numerical results are computed for $\alpha_1 = \alpha_2 = 1$, for which the exact solutions are known.

Problem 1. The linear stiff system used by Noor and Yatim [36]:

$$\begin{aligned} D^{\alpha_1} v_1(t) &= -v_1(t) + 95 v_2(t), \\ D^{\alpha_2} v_2(t) &= -v_1(t) - 97 v_2(t), \end{aligned}$$

with $v_1(0) = 1$, $v_2(0) = 1$ and $0 \le t \le 10$. For $\alpha_1 = \alpha_2 = 1$ the exact solution is

$$\begin{aligned} v_1(t) &= \tfrac{1}{47}\big(95 e^{-2t} - 48 e^{-96t}\big), \\ v_2(t) &= \tfrac{1}{47}\big(48 e^{-96t} - e^{-2t}\big). \end{aligned}$$

The eigenvalues of the coefficient matrix are $\lambda_1 = -2$ and $\lambda_2 = -96$, so the stiffness ratio $|\lambda_2|/|\lambda_1| = 48$ confirms that the problem is stiff. The fast component e^{-96t} decays within a short initial layer, after which the solution is governed by the slow component e^{-2t}; a method with a small stability region would need a very small step size over the whole interval. The fast component falls below 10^{-6} already at $t \approx 0.14$, so for most of the interval $[0, 10]$ the step size is limited by stability rather than by accuracy when an explicit method is used.

Problem 2. The linear system with an oscillatory solution, of the kind used as a stiff-system test problem in [49]:

$$\begin{aligned} D^{\alpha_1} v_1(t) &= -3 v_1(t) + 2 v_2(t) + 3\cos t - 3\sin t, \\ D^{\alpha_2} v_2(t) &= 2 v_1(t) - 3 v_2(t) - \cos t + 3\sin t, \end{aligned}$$

with $v_1(0) = 1$, $v_2(0) = 0$ and $0 \le t \le 20$. For $\alpha_1 = \alpha_2 = 1$ the exact solution is $v_1(t) = \cos t$, $v_2(t) = \sin t$, and the eigenvalues of the coefficient matrix are $\lambda_1 = -1$ and $\lambda_2 = -5$. The solution oscillates with unit frequency and does not decay, and the stiffness ratio is only 5, so this problem mainly tests the accuracy of the method for a smooth, persistent solution.

Problem 3. The Kaps chemical-kinetics problem, a standard stiff-system test of the type treated in [49], written in fractional form,

$$D^{\alpha_1} v_1(t) = -(\epsilon^{-1} + 2) v_1(t) + \epsilon^{-1} v_2(t)^2,$$
$$D^{\alpha_2} v_2(t) = v_1(t) - v_2(t)\big(1 + v_2(t)\big),$$

with $v_1(0) = v_2(0) = 1$ and $0 \le t \le 20$, where D^{α} denotes the Caputo derivative of order $0 < \alpha \le 1$ and $\epsilon > 0$ is the stiffness parameter. For $\alpha = 1$ the exact solution is $v_1(t) = e^{-2t}$, $v_2(t) = e^{-t}$. The Jacobian evaluated at the equilibrium $(0, 0)$ has eigenvalues $\lambda_1 = -1$ and $\lambda_2 = -(\epsilon^{-1} + 2)$, so the stiffness grows as ϵ decreases. For $\epsilon = 10^{-3}$, the value used in Table 8, $\lambda_2 \approx -1002$ and the stiffness ratio is about 10^3. The problem is therefore a demanding test of both the stability and the accuracy of the method. It is widely used for testing stiff solvers because its exact solution is the same for every value of ϵ, so that the effect of increasing stiffness can be studied without changing the reference solution.

The proposed 2SFBBDF3 is compared with the fully implicit third-order two-point fractional block BDF (2FBBDF3) [36] and the diagonally implicit second-order two-point fractional block BDF (2DIFBBDF2) [43]. The first reference method is the member $\rho = 0$ of the family (5), so the comparison with it shows directly the effect of the parameter ρ at the same order and with the same stencil. The second is a diagonally implicit scheme, whose stage equations can be solved one after the other, and which was reported in [43] to be A-stable for fractional orders between 0.7 and 1. It therefore represents a cheaper but lower-order alternative. Using one reference method of the same order and one of lower order makes it possible to separate the effect of the parameter from the effect of the order of the scheme. To compare the schemes, both accuracy and computational cost are considered. For each problem we record the maximum absolute error (MAXE) and the CPU time for a sequence of constant step sizes, together with the total number of steps (TS) as an additional measure of cost. All experiments use $\alpha = 1$ for all three methods, so that the comparison is made under identical conditions. The three test problems are standard benchmarks in the numerical analysis literature. The results for 2FBBDF3 and 2DIFBBDF2 were recomputed in this study rather than copied from the original publications; the parameter settings, constant step sizes and other computational settings were chosen to agree with those reported in [36, 43]. The proposed 2SFBBDF3 was then applied to the same problems in the same program, with the same step sizes and the same computing environment, so that differences caused by implementation details, parameter choices or hardware are kept as small as possible. The results are given in Tables 6-8. For each step size the tables list, for the three methods, the number of steps, the maximum absolute error defined in (27), and the CPU time in seconds.

Figures 7-9 plot $\log_{10}(\mathrm{MAXE})$ against $\log_{10} h$ for the three problems. They present the data of Tables 6-8 in graphical form and make it easier to see how quickly the errors of the three methods decrease as the step size is reduced. In Figures 7 and 9 the vertical scale is dominated by the very large errors of the reference methods at $h = 10^{-2}$.

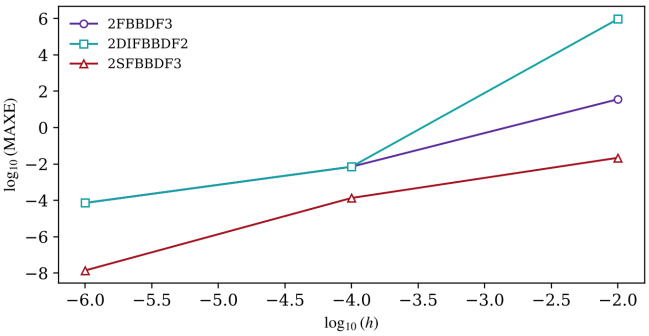


Figure 7. Accuracy curves for Problem 1.

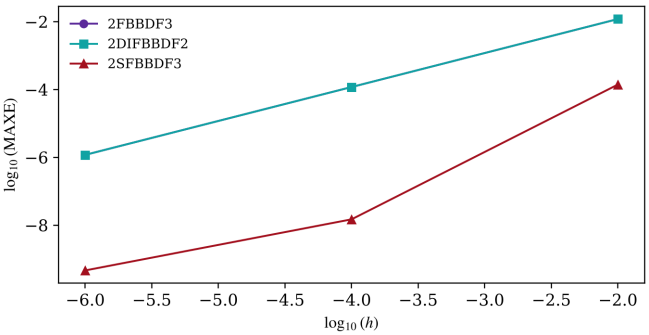


Figure 8. Accuracy curves for Problem 2.

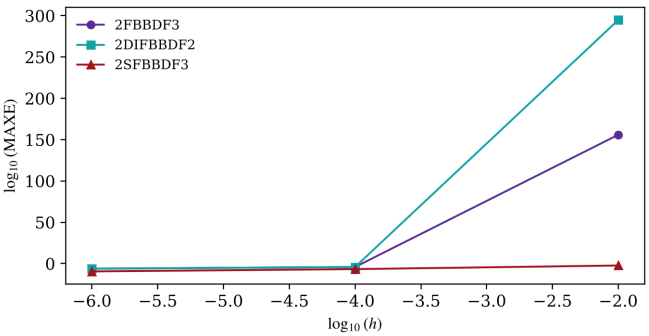


Figure 9. Accuracy curves for Problem 3.

Table 6. Comparison of 2SFBBDF3 with existing methods for Problem 1 ($\alpha = 1$).

h	Method	TS	MAXE	CPU time (s)
10^{-2}	2FBBDF3	500	3.62349×10^{1}	2.45900×10^{-4}
	2DIFBBDF2	500	9.37034×10^{5}	2.26100×10^{-4}
	2SFBBDF3	500	2.18712×10^{-2}	4.12500×10^{-5}
10^{-4}	2FBBDF3	50000	7.04927×10^{-3}	2.61400×10^{-3}
	2DIFBBDF2	50000	7.04562×10^{-3}	3.02900×10^{-3}
	2SFBBDF3	50000	1.34815×10^{-4}	4.76400×10^{-4}
10^{-6}	2FBBDF3	5000000	7.21175×10^{-5}	3.26900×10^{-2}
	2DIFBBDF2	5000000	7.21171×10^{-5}	5.72800×10^{-3}
	2SFBBDF3	5000000	1.39191×10^{-8}	5.31600×10^{-3}

Table 7. Comparison of 2SFBBDF3 with existing methods for Problem 2 ($\alpha = 1$).

h	Method	TS	MAXE	CPU time (s)
10^{-2}	2FBBDF3	1000	1.18959×10^{-2}	4.62400×10^{-3}
	2DIFBBDF2	1000	1.18964×10^{-2}	4.29100×10^{-3}
	2SFBBDF3	1000	1.39285×10^{-4}	4.12600×10^{-4}
10^{-4}	2FBBDF3	100000	1.17682×10^{-4}	4.82600×10^{-3}
	2DIFBBDF2	100000	1.17682×10^{-4}	3.99400×10^{-3}
	2SFBBDF3	100000	1.47844×10^{-8}	3.66100×10^{-3}
10^{-6}	2FBBDF3	10000000	1.17676×10^{-6}	5.74200×10^{-3}
	2DIFBBDF2	10000000	1.17673×10^{-6}	4.34100×10^{-3}
	2SFBBDF3	10000000	4.60420×10^{-10}	3.48310×10^{-3}

Table 8. Comparison of 2SFBBDF3 with existing methods for Problem 3 ($\alpha = 1, \epsilon = 10^{-3}$).

h	Method	TS	MAXE	CPU time (s)
10^{-2}	2FBBDF3	1000	7.71189×10^{155}	3.52100×10^{-4}
	2DIFBBDF2	1000	5.92077×10^{294}	3.02600×10^{-4}
	2SFBBDF3	1000	4.73428×10^{-3}	3.14600×10^{-5}
10^{-4}	2FBBDF3	100000	7.37885×10^{-5}	4.21600×10^{-3}
	2DIFBBDF2	100000	7.37870×10^{-5}	3.73900×10^{-3}
	2SFBBDF3	100000	2.06260×10^{-7}	3.51400×10^{-4}
10^{-6}	2FBBDF3	10000000	7.38360×10^{-7}	4.47200×10^{-3}
	2DIFBBDF2	10000000	7.38382×10^{-7}	4.10800×10^{-3}
	2SFBBDF3	10000000	3.23417×10^{-10}	3.91300×10^{-3}

8 Discussion of Numerical Results

The numerical results demonstrate the effectiveness of the proposed 2SFBBDF3 in comparison with the existing 2FBBDF3 and 2DIFBBDF2 methods. All the methods are implemented under identical computational conditions and step-size settings, providing a fair basis for assessing their accuracy and computational efficiency. The results generally show that the proposed 2SFBBDF3 produces more accurate numerical approximations than the competing methods, with its maximum absolute error decreasing substantially as the step size is refined. This improvement becomes particularly evident at finer discretizations, indicating the strong approximation capability of the proposed two-point superclass formulation.

The results further reveal that the proposed method maintains favorable computational efficiency while achieving improved accuracy. The CPU-time measurements indicate that the proposed 2SFBBDF3

generally requires less computational effort than the existing methods under corresponding step-size conditions. Since all methods are evaluated using the same step sizes and consequently the same number of time steps, the observed computational advantage is not due to the use of fewer integration steps. Rather, it reflects the efficiency of the proposed formulation and its ability to obtain highly accurate numerical solutions without imposing excessive computational costs.

Another important observation is the consistent reduction in the maximum absolute error as the step size decreases. This behavior provides numerical evidence of the convergent nature of the proposed scheme and is consistent with the theoretical analysis establishing its consistency, zero-stability, convergence, and absolute stability properties. Although the numerical results show a clear reduction of the error with the step size, they do not by themselves confirm third-order accuracy, and the observed error reductions should be interpreted as practical evidence of its approximation capability rather than as an independent proof of the theoretical order of convergence. The numerical experiments therefore complement the theoretical analysis by demonstrating that the favorable properties established analytically translate into effective computational performance.

Overall, the numerical results indicate that the proposed 2SFBBDF3 offers a favorable balance between accuracy and computational efficiency. Its ability to produce substantially smaller numerical errors under refined step sizes while maintaining competitive computational costs demonstrates the practical advantage of the proposed formulation over the existing methods considered. These results further support the suitability of the proposed scheme for the numerical treatment of stiff differential systems and highlight its potential as an efficient and reliable computational technique.

Two limitations of the present experiments should be stated clearly. First, all results were obtained for $\alpha = 1$, where the Caputo derivative reduces to the ordinary derivative, so they show the behavior of the method as an integrator for stiff ODEs and do not test the treatment of the fractional memory. Second, only constant step sizes and a single value of the parameter, $\rho = -3/4$, were used. A study of other values of ρ and of variable step sizes would show how much of the improvement is due to the parameter itself. Third, the errors of 2FBBDF3 and 2DIFBBDF2 in Tables 6-8 decrease only linearly in h and are almost identical for the two methods, which points to an error floor set by the starting values rather than by the methods; the procedure that supplies the starting values and the timing procedure should be fixed and reported identically for all methods before the quantitative differences in the tables are interpreted.

9 Conclusion

This paper has presented a third-order fully implicit two-point superclass fractional block backward differentiation formula (2SFBBDF3) for stiff systems of fractional-order differential equations with the Caputo derivative. The method is built on a two-point block framework, so that the solution is computed at two consecutive grid points in each step, and it contains a stability-control parameter ρ. For $\rho = 0$ it reduces to the existing 2FBBDF3, and the value $\rho = -3/4$ was used throughout.

The theoretical properties of the method were studied for the fractional orders $\alpha = 0.75, 0.85, 0.95$ and 1. The method is of order three, consistent and zero-stable for all four orders, and its convergence follows from these two properties. The regions of absolute stability, obtained from the fractional Dahlquist test equation, contain the whole left half-plane for $\alpha = 0.75$-0.95. For $\alpha = 1$ the boundary crosses slightly into the left half-plane near the imaginary axis, and the method is A(θ)-stable with θ close to 90°; the negative real axis is contained in the stability region in all cases.

The numerical experiments were carried out in the classical limiting case $\alpha = 1$ on two linear and one nonlinear stiff system, and the method was compared with 2FBBDF3 and 2DIFBBDF2 under identical problem conditions and step sizes. 2SFBBDF3 gave the smallest maximum error at every step size, and for the coarsest step it remained accurate on the two stiff problems where the reference methods failed. Its recorded CPU times were not larger than those of the reference methods. The observed convergence rates, however, were lower than the theoretical order, and this point needs further study. Because 2FBBDF3 is the member $\rho = 0$ of the same family, the comparison with it is the natural test of the stability-control parameter. The present experiments do not, however, isolate the effect of ρ: the starting values and the Newton settings are not varied, the errors of the reference methods decrease only as $O(h)$ while the theoretical order is three, and the measured CPU times are very small. The attribution of the accuracy gain to ρ therefore remains to be confirmed with identical,

accurate starting values for all methods.

Future work will apply the method directly to genuinely fractional problems with $0 < \alpha < 1$, treat the memory term of the Caputo derivative explicitly, and extend the tests to larger nonlinear stiff systems, fractional chemical-kinetics models and other fractional-order applications.

Data Availability Statement

The data used to support the findings of this study are available from the corresponding author upon request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable. This study does not involve human participants or animals.

References

[1] Podlubny, I. (1998). Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications (Vol. 198). elsevier.

[2] Diethelm, K., & Ford, N. J. (2002). Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications, 265*(2), 229-248. [CrossRef]

[3] Kilbas, A. A., Srivastava, H. M., & Trujillo, J. J. (2006). *Theory and Applications of Fractional Differential Equations*. Elsevier, Amsterdam. https://sutlib2.sut.ac.th/sut_contents/H103746.pdf

[4] Samko, S. G., Kilbas, A. A., & Marichev, O. I. (1993). *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach, Amsterdam. https://scispace.com/pdf/fractional-integrals-and-derivatives-theory-and-applications-3mo1ouv89m.pdf

[5] Sun, H., Zhang, Y., Baleanu, D., Chen, W., & Chen, Y. (2018). A new collection of real world applications of fractional calculus in science and engineering. *Communications in Nonlinear Science and Numerical Simulation, 64*, 213-231. [CrossRef]

[6] Magin, R. L. (2006). *Fractional Calculus in Bioengineering*. Begell House, Redding. https://www.begellhouse.com/books/0f9d45b5270a0d80.html

[7] Mainardi, F. (2010). *Fractional Calculus and Waves in Linear Viscoelasticity*. Imperial College Press, London. [CrossRef]

[8] Caputo, M. (1967). Linear models of dissipation whose Q is almost frequency independent—II. *Geophysical Journal International, 13*(5), 529-539. [CrossRef]

[9] Diethelm, K., & Ford, N. J. (2004). Multi-order fractional differential equations and their numerical solution. *Applied Mathematics and Computation, 154*(3), 621-640. [CrossRef]

[10] Diethelm, K. (2010). *The Analysis of Fractional Differential Equations*. Springer, Berlin. [CrossRef]

[11] Li, C. P., & Zhang, F. R. (2011). A survey on the stability of fractional differential equations: Dedicated to Prof. YS Chen on the Occasion of his 80th Birthday. *The European Physical Journal Special Topics, 193*(1), 27-47. [CrossRef]

[12] Diethelm, K., Ford, N. J., Freed, A. D., & Luchko, Y. (2005). Algorithms for the fractional calculus: a selection of numerical methods. *Computer Methods in Applied Mechanics and Engineering, 194*(6-8), 743-773. [CrossRef]

[13] Ford, N. J., & Simpson, A. C. (2001). The numerical solution of fractional differential equations: speed versus accuracy. *Numerical Algorithms, 26*(4), 333-346. [CrossRef]

[14] Diethelm, K., Ford, N. J., & Freed, A. D. (2002). A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dynamics, 29*(1), 3-22. [CrossRef]

[15] Diethelm, K., Ford, N. J., & Freed, A. D. (2004). Detailed error analysis for a fractional Adams method. *Numerical Algorithms, 36*(1), 31-52. [CrossRef]

[16] Garrappa, R. (2009). On some explicit Adams multistep methods for fractional differential equations. *Journal of Computational and Applied Mathematics, 229*(2), 392-399. [CrossRef]

[17] Garrappa, R. (2018). Numerical solution of fractional differential equations: A survey and a software tutorial. *Mathematics, 6*(2), 16. [CrossRef]

[18] Odibat, Z. (2006). Approximations of fractional integrals and Caputo fractional derivatives. *Applied Mathematics and Computation, 178*(2), 527-533. [CrossRef]

[19] Scherer, R., Kalla, S. L., Tang, Y., & Huang, J. (2011). The Grünwald–Letnikov method for fractional differential equations. *Computers & Mathematics with Applications, 62*(3), 902-917. [CrossRef]

[20] Lubich, C. (1985). Fractional linear multistep methods for Abel-Volterra integral equations of the second kind. *Mathematics of Computation, 45*(172), 463-469.

[CrossRef]

[21] Lubich, C. (1986). Discretized fractional calculus. *SIAM Journal on Mathematical Analysis, 17*(3), 704-719. [CrossRef]

[22] Lubich, C. (1986). A stability analysis of convolution quadratures for Abel-Volterra integral equations. *IMA Journal of Numerical Analysis, 6*(1), 87-101. [CrossRef]

[23] Galeone, L., & Garrappa, R. (2006). On multistep methods for differential equations of fractional order. *Mediterranean Journal of Mathematics, 3*(3), 565-580. [CrossRef]

[24] Garrappa, R. (2012). Stability-preserving high-order methods for multiterm fractional differential equations. *International Journal of Bifurcation and Chaos, 22*(04), 1250073. [CrossRef]

[25] Aceto, L., Magherini, C., & Novati, P. (2014). Fractional convolution quadrature based on generalized Adams methods. *Calcolo, 51*(3), 441-463. [CrossRef]

[26] Galeone, L., & Garrappa, R. (2008). Fractional Adams–Moulton methods. *Mathematics and Computers in Simulation, 79*(4), 1358-1367. [CrossRef]

[27] Galeone, L., & Garrappa, R. (2009). Explicit methods for fractional differential equations and their stability properties. *Journal of Computational and Applied Mathematics, 228*(2), 548-560. [CrossRef]

[28] Gear, C. W. (1971). *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice Hall, Englewood Cliffs, NJ.

[29] Hairer, E., & Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems* (2nd rev. ed.). Springer Series in Computational Mathematics, Vol. 14. Springer, Berlin. [CrossRef]

[30] Bonab, Z. F., & Javidi, M. (2020). Higher order methods for fractional differential equation based on fractional backward differentiation formula of order three. *Mathematics and Computers in Simulation, 172*, 71-89. [CrossRef]

[31] Biala, T. A., & Jator, S. N. (2015). Block backward differentiation formulas for fractional differential equations. *International Journal of Engineering Mathematics, 2015*(1), 650425. [CrossRef]

[32] Biala, T. A., & Jator, S. N. (2015). Block implicit Adams methods for fractional differential equations. *Chaos, Solitons & Fractals, 81*, 365-377. [CrossRef]

[33] Gorenflo, R., Kilbas, A. A., Mainardi, F., & Rogosin, S. V. (2014). *Mittag-Leffler Functions, Related Topics and Applications*. Springer, Berlin, Heidelberg. [CrossRef]

[34] Noor, N. M., Yatim, S. A. M., & Ibrahim, Z. B. (2024). Fractional block method for the solution of fractional order differential equations. *Malaysian Journal of Mathematical Sciences, 18*(1), 185-208. [CrossRef]

[35] Suleiman, M. B., Musa, H., Ismail, F., Senu, N., & Ibrahim, Z. B. (2014). A new superclass of block backward differentiation formula for stiff ordinary differential equations. *Asian-European Journal of Mathematics, 7*(01), 1350034. [CrossRef]

[36] Noor, N. M., & Yatim, S. A. M. (2024, October). Stability Analysis of the Fractional Block Method for Solving FODEs. In *UiTM International Conference on Mathematical Sciences* (pp. 15-27). Singapore: Springer Nature Singapore. [CrossRef]

[37] Bala, N., Musa, H., & Alhassan, B. (2026). A Third-Order Variable Step Size Superclass of Block Backward Differentiation Formula for Efficient Solution of Highly Stiff Differential Systems. *Journal of Nonlinear Dynamics and Applications, 2*(3), 143-158. [CrossRef]

[38] Alhassan, B., Saadu, A. S., & Musa, H. (2026). Time Dependent Algorithm using Modified Three-Point Superclass of Block Backward Differentiation Formula for Solving Stiff Ordinary Differential Equations. *ICCK Journal of Applied Mathematics, 2*(1), 87-110. [CrossRef]

[39] Ibrahim, Z. B., Noor, N. M., & Othman, K. I. (2019). Fixed coefficient A(α) stable block backward differentiation formulas for stiff ordinary differential equations. *Symmetry, 11*(7), 846. [CrossRef]

[40] Ijam, H. M., Ibrahim, Z. B., & Zawawi, I. S. M. (2024). Stiffly stable diagonally implicit block backward differentiation formula with adaptive step size strategy for stiff ordinary differential equations. *MATEMATIKA, 40*(1), 27-47. [CrossRef]

[41] Ijam, H. M., & Ibrahim, Z. B. (2019). Diagonally implicit block backward differentiation formula with optimal stability properties for stiff ordinary differential equations. *Symmetry, 11*(11), 1342. [CrossRef]

[42] Suleiman, M. B., Musa, H., Ismail, F., & Senu, N. (2013). A new variable step size block backward differentiation formula for solving stiff initial value problems. *International Journal of Computer Mathematics, 90*(11), 2391-2408. [CrossRef]

[43] Yiung, Y. L., & Yatim, S. A. M. (2025). Two-Point Diagonally Implicit Fractional Block Backward Differentiation Formula for Solving Fractional Differential Equations. *Malaysian Journal of Fundamental and Applied Sciences, 21*(4), 2356-2373. [CrossRef]

[44] Lambert, J. D. (1973). *Computational Methods in Ordinary Differential Equations*. John Wiley & Sons, Chichester.

[45] Zawawi, I. S. M., Ibrahim, Z. B., Ismail, F., & Majid, Z. A. (2012). Diagonally implicit block backward differentiation formulas for solving ordinary differential equations. *International Journal of Mathematics and Mathematical Sciences, 2012*, 767328. [CrossRef]

[46] Zabidi, N. A., Abdul Majid, Z., Kilicman, A., & Rabiei, F. (2020). Numerical solutions of fractional differential equations by using fractional explicit Adams method. *Mathematics, 8*(10), 1675. [CrossRef]

[47] Noor, N. M., Yatim, S. A. M., & Zawawi, I. S. M. (2024). The design of the new fractional block method in the solution of the three-compartment pharmacokinetics model. *Engineering Letters, 32*(2), 252-261. https://www.engineeringletters.com/issues_v32/issue_2/EL_32_2_06.pdf

[48] Ibrahim, Z. B., Othman, K. I., & Suleiman, M. (2007). Implicit r-point block backward differentiation formula for solving first-order stiff ODEs. *Applied Mathematics and Computation, 186*(1), 558-565. [CrossRef]

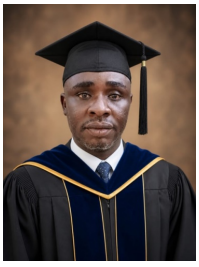
[49] Aminikhah, H., & Hemmatnezhad, M. (2011). An effective modification of the homotopy perturbation method for stiff systems of ordinary differential equations. *Applied Mathematics Letters, 24*(9), 1502-1508. [CrossRef]



Buhari Alhassan is a Lecturer in the Department of Mathematics and Statistics, Al-Qalam University Katsina, Nigeria. He obtained the NCE in Mathematics/Chemistry from Isah Kaita College of Education, Dutsin-Ma, and the B.Sc., M.Sc. and Ph.D. degrees in Mathematics (Numerical Analysis) from Umaru Musa Yar'adua University, Katsina, Nigeria. His research interests are numerical analysis and scientific computing, in particular the development and implementation of numerical methods for ordinary and fractional differential equations. He has published several papers in peer-reviewed journals and is actively involved in teaching and research. (Email: buharialhassan@auk.edu.ng)



Hamisu Musa is a Professor of Numerical Analysis in the Department of Mathematics, Umaru Musa Yar'adua University, Katsina, Nigeria. He obtained the B.Sc. and M.Sc. degrees in Mathematics from Bayero University, Kano, Nigeria, and the Ph.D. degree in Numerical Analysis from Universiti Putra Malaysia. He has an extensive record of postgraduate supervision: he has guided numerous Ph.D. candidates to completion, including the first and third authors of this work, continues to supervise several doctoral studies, and has supervised more than twenty M.Sc. students in Mathematics. His research focuses on the development of advanced numerical methods, particularly for stiff initial value problems. (Email: hamisu.musa@umyu.edu.ng)



Najamuddeen Bala obtained the B.Sc. degree in Mathematics from Bayero University Kano, Nigeria, and the M.Sc. and Ph.D. degrees in Mathematics, specializing in Numerical Analysis, from Umaru Musa Yar'adua University, Katsina State, Nigeria. He is a Lecturer in the Department of Statistics, Federal Polytechnic Kaura-Namoda, Zamfara State, Nigeria. His research centers on the development, analysis and implementation of efficient numerical techniques for stiff ordinary differential equations. He has authored and co-authored several articles in peer-reviewed journals and is engaged in teaching, student supervision and the development of numerical algorithms for applied problems. (Email: Phdmth190179@students.umyu.edu.ng)



Abdulrahman Adamu is a Ph.D. candidate in the Department of Mathematics, Umaru Musa Yar'adua University, Katsina, Nigeria, and a staff member of the Department of Mathematics and Statistics, Al-Qalam University Katsina. His research interests include numerical analysis and computational mathematics, with a particular focus on the development of numerical methods for solving stiff initial value problems (IVPs). (Email: adamuaman@auk.edu.ng)



Sharhabilu Umar Wasagu is an M.Sc. candidate in the Department of Mathematics, Federal University Dutsin-Ma, Katsina State, Nigeria, and a staff member of the Department of Mathematics, Federal College of Education, Katsina, Nigeria. His research interests are in numerical analysis, with a particular focus on adaptive variable step size methods for solving stiff initial value problems (IVPs). (Email: sharhabilu.umar@fcekatsina.edu.ng)



Kabiru Abubakar Kantsi holds the B.Sc. degree in Mathematics from Kano State University of Science and Technology, Wudil, and the M.Sc. degree in Mathematics from Al-Qalam University Katsina, Nigeria, where he is with the Department of Mathematics and Statistics. His research interests include applied mathematics and numerical analysis, with emphasis on numerical methods for stiff ordinary differential equations. He is engaged in research on the development of computational techniques. (Email: kakantsi88@gmail.com)



Abubakar Jibrin holds the B.Sc. degree in Mathematics with First Class Honours from Kebbi State University of Science and Technology, Aliero, Nigeria, and the M.Sc. degree in Mathematics from Bayero University, Kano, Nigeria. His research interests include pure and applied mathematics, particularly algebra and semigroup theory. He is an academic staff member of the Department of Mathematics and Statistics, Al-Qalam University Katsina, Katsina State, Nigeria. (Email: abubakarjibrin@auk.edu.ng)



Hamza Yusuf obtained the B.Sc. and M.Sc. degrees in Mathematics, specializing in Numerical Analysis, from Umaru Musa Yar'adua University, Katsina State, Nigeria. He is pursuing the Ph.D. degree in Mathematics at Bayero University, Kano, Nigeria, with a research focus on numerical optimization techniques for unconstrained optimization problems. He is a Lecturer in the Department of Mathematics, Federal University Dutsin-Ma, Katsina State, Nigeria. (Email: yhamza@fudutsinma.edu.ng)



Asmau Abdullahi Mani is a Ph.D. candidate in the Department of Mathematics, Umaru Musa Yar'adua University, Katsina, Nigeria, where she is also a staff member. Her research interests include numerical analysis and computational mathematics, with a particular focus on the development of numerical methods for solving stiff initial value problems (IVPs). (Email: asmau.amani@umyu.edu.ng)



Muhammad Abdullahi obtained the B.Sc. and M.Sc. degrees in Mathematics from Umaru Musa Yar'adua University, Katsina State, Nigeria, and is pursuing the Ph.D. degree in Mathematics, specializing in Numerical Analysis, at the Federal University Dutsin-Ma, Katsina State, Nigeria, where he is a Lecturer in the Department of Mathematics. His research interests lie in numerical analysis, particularly the development, analysis and implementation of efficient numerical methods for stiff ordinary differential equations. He has authored and co-authored several articles in peer-reviewed journals and is involved in teaching, student supervision and the development of numerical algorithms. (Email: maunwala@gmail.com)