Received: xxx **Accepted:** xxx **Published:** xxx.
DOI. xxxxxxx
Volume xxx, Issue xxx, p.p. 1-34: xxx
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

Comparative Accuracy Analysis of Spectral and Collocation Methods for the Fractional Bagley–Torvik Equation: A Systematic Numerical Study

Muayyad Mahmood Khalil ✉, **Najim Abdullah Ibrahim**

Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq

✉ **Correspondence:**
Muayyad Mahmood Khalil
E-mail:
medomath80@tu.edu.iq

How to Cite
Khalil, M.M., Ibrahim, N.A. (2027). "Comparative accuracy analysis of spectral and collocation methods for the fractional Bagley–Torvik equation: A systematic numerical study" *Control and Optimization in Applied Mathematics*, X(x): 1-34. https://doi.org/10.30473/coam.2026.77807.1408

Abstract. The Bagley–Torvik equation, which governs the motion of a rigid plate immersed in a Newtonian fluid with viscoelastic damping, represents one of the canonical benchmark problems in fractional calculus. This paper presents a systematic comparative numerical analysis of four established methods for solving this equation: the Fractional-Order Hybrid Jacobi Functions method (FOHJF), the Polynomial Least Squares Method (PLSM), the Vieta-Lucas Spectral Method (VLSM), and the Cubic Spline Collocation Method (CSCM). Three benchmark test problems with qualitatively distinct forcing functions—polynomial, oscillatory (cosine), and exponential—and varying initial conditions are used to evaluate absolute approximation errors at resolution levels $N = 8, 16, 32$, and 64. Detailed error tables and graphical convergence analyses are provided. The results consistently demonstrate that VLSM achieves the highest accuracy, with maximum absolute errors below 2.3×10^{-8} at $N = 32$, followed by FOHJF. PLSM and CSCM offer simpler implementation at the cost of reduced accuracy. Practical recommendations are provided for selecting a method based on the required precision level and the type of forcing function. The study identifies key limitations and directions for future work, including extension to nonlinear formulations, variable-order derivatives, and adaptive hybrid approaches.

Keywords. Bagley–Torvik equation, Caputo fractional derivative, Spectral methods, Collocation methods, Vieta-Lucas polynomials, Fractional-order Jacobi functions, Error analysis.

MSC. 34A08; 65L60; 26A33.

1 Introduction

Fractional differential equations (FDEs) play an important role in describing physical systems with hereditary effects and memory-dependent behavior [18, 24]. In contrast to classical integer-order models, they can more effectively describe non-local interactions and long-term dependence [9]. Consequently, fractional calculus has found wide use across numerous fields, including viscoelasticity, control theory, signal processing, bioengineering, and financial modelling [20].

Among fractional models, the Bagley–Torvik equation is of particular interest due to its role in describing the motion of a rigid plate in a Newtonian fluid and viscoelastic damping processes. Given its relevance to engineering and applied mathematics, the Bagley–Torvik equation is notable for combining a second-order ordinary derivative with a fractional derivative of order $3/2$, thereby representing both inertial and memory-dependent damping effects [2, 27].

Classical benchmarks were established by the seminal numerical work of Diethelm and Ford [10]. However, a comprehensive comparative error analysis of modern techniques, including the Fractional-Order Hybrid Jacobi Functions method (FOHJF), the Polynomial Least Squares Method (PLSM), the Vieta-Lucas Spectral Method (VLSM), and the Cubic Spline Collocation Method (CSCM), has not yet been carried out under a unified computational setting.

The contribution of this paper is a unified, reproducible benchmarking framework for assessing four established numerical methods for the fractional Bagley–Torvik equation: FOHJF, PLSM, VLSM, and CSCM. This paper does not propose a new numerical method; instead, it evaluates these methods under identical computational assumptions, including the same governing equation, interval, resolution levels, evaluation points, error measures, and hardware/software environment. The comparison is carried out for three manufactured benchmark problems with polynomial, oscillatory, and exponential solution behavior. Therefore, the novelty lies in the systematic cross-method assessment of accuracy, convergence, computational cost, and conditioning. To the best of the authors' knowledge, such a unified four-method comparison under identical benchmark conditions has not been previously reported for the fractional Bagley–Torvik equation.

Accordingly, the specific aims of this work are to develop a common numerical setting for applying FOHJF, PLSM, VLSM, and CSCM to the fractional Bagley–Torvik equation; to compare pointwise absolute errors, maximum errors, and mean errors for three benchmark problems; to examine convergence with respect to the number of basis functions or grid points; and to report computational time and condition-number information in order to support practical method selection.

The general form of the Bagley–Torvik equation is given by:

$$Ay''(t) + BD^{\alpha}y(t) + Cy(t) = f(t), \quad t \in [0, T], \tag{1}$$

where D^{α} denotes the Caputo fractional derivative of order $\alpha = 3/2$, A, B, and C are physical parameters, and $f(t)$ represents external forcing. The Caputo fractional derivative is defined as:

$$D^{\alpha}y(t) = \frac{1}{\Gamma(2-\alpha)} \int_0^t (t-\tau)^{1-\alpha} y''(\tau)\, \mathrm{d}\tau, \quad 1 < \alpha < 2. \tag{2}$$

The inherent non-locality of the fractional derivative operator poses both conceptual and computational challenges, because the solution value at a given time depends on the process's previous history [9, 18, 22, 24]. This global memory dependence renders standard time-marching schemes computationally inefficient and motivates the use of specialized numerical approaches. Over the past decades, a wide range of numerical methods has been developed for fractional differential equations, each with its own advantages and limitations. Spectral methods based on orthogonal polynomials have received particular attention because they often exhibit rapid convergence for sufficiently smooth solutions [11, 28]. These methods approximate the desired solution using globally supported basis functions and can therefore achieve high accuracy with relatively few degrees of freedom.

In related spectral methodologies, a fractional-order Jacobi Tau method has been used to solve time-fractional partial differential equations by Bhrawy and Zaky [6], and multi-order problems have been solved using spectral collocation by Ghoreishi and Mokhtary [14].

The applicability of collocation schemes is evident in fractional differential and integro-differential equations. Rawashdeh [26] used a collocation scheme to solve fractional integro-differential equations, thus providing one of the earliest examples of its effectiveness for fractional models. In relation to the Bagley–Torvik problem, collocation-based methods considered for the model include generalized Taylor collocation scheme [8], various other collocation-based methods [15], splines [1], Bessel collocation [33], and fractional Taylor methods [19]. This is particularly significant for our proposed CSCM scheme, as it shares similarities with collocation and spline-collocation schemes in enforcing the governing equation at select nodes, and in approximating the unknown solution as finite expansions.

Least-squares approaches produce an effective means for evaluating solutions to fractional differential equations within a variational framework. By way of example, Bota et al. [7] used a least-squares approach in combination with differential quadrature for the generalized Bagley–Torvik problem, which results in manageable algebraic systems. This variational framework has proved to various kinds of fractional differential equations under different initial and boundary conditions. The use of fractional-order hybrid Jacobi functions, developed by Barary et al. [5] for fractional differential and control-type problems, has resulted in locally supported spectral basis functions formed from block pulse functions and fractional-order Jacobi polynomials. Likewise, the use of Vieta–Lucas polynomials based operational matrices has been developed for Caputo fractional-order differential equations [23].

Neural-network-based approaches have also begun to appear in the numerical treatment of fractional differential equations. Raja et al. [25] proposed an unsupervised fractional neural-network model optimized by an interior-point algorithm for the Bagley–Torvik equation, whereas Verma and Kumar [32] introduced Legendre-based artificial neural-network approaches. These mesh-free methods are flexible, but they require careful training, parameter selection, and monitoring of convergence. Lastly, analytical and semi-analytical treatments remain as complements. Jena and Chakraverty [17] employed the Sumudu transform method and Mekkaoui and Hammouch [21] used the fractional iteration method to obtain approximate analytical solutions of the Bagley–Torvik equation.

Refinements of homotopy perturbation techniques were introduced by Zolfaghari et al. [35]. Bansal and Jain [4] applied the generalized differential transform method to obtain analytical solutions of the Bagley–Torvik equation. More recently, Salati et al. [29] proposed a numerical approach for the Bagley–Torvik equation and related fractional oscillation equations. Although these studies provide useful analytical and numerical treatments, they do not present a unified quantitative comparison of FOHJF, PLSM, VLSM, and CSCM under identical benchmark conditions. This gap motivates the present systematic comparative study.

The remainder of this paper is organized as follows. Section 2 introduces the governing equation of the Bagley–Torvik model in its physical and dimensionless forms, and recalls the definition of the Caputo fractional derivative of order $3/2$ used throughout the study. Section 3 presents the four numerical methods under comparison—FOHJF, PLSM, VLSM, and CSCM—together with their respective operational matrices, basis functions, and stability considerations. Section 4 defines the three manufactured benchmark test problems with polynomial, oscillatory, and exponential exact solutions, and reports the pointwise absolute errors of each method at $N = 32$. Section 5 provides a cross-problem comparative analysis of maximum and mean absolute errors at the benchmark resolution. Section 6 investigates the convergence behaviour of all four methods across resolution levels $N = 8, 16, 32$, and 64, and estimates the experimental order of convergence. Section 7 discusses the results in terms of accuracy ranking, computational cost, and condition numbers. Finally, Section 8 summarises the main conclusions and outlines directions for future research.

2 Governing Equation

2.1 The Bagley–Torvik Equation

In the original derivation of the Bagley–Torvik equation by Bagley and Torvik [31], the fractional derivatives are a natural consequence of the motion of a rigid plate in a Newtonian fluid. Since then, it has been commonly adopted for numerical methods on fractional differential

equations [10]. The governing equation of motion includes the viscoelastic response of the surrounding fluid in terms of a fractional-order derivative:

$$My''(t) + 2S\sqrt{\mu\rho}\, D^{3/2}y(t) + Ky(t) = F(t), \tag{3}$$

where S is the plate area, μ the dynamic viscosity, ρ the fluid density, M the plate mass, and K the spring stiffness.

$$y''(t) + \eta\, D^{3/2}y(t) + \omega^2 y(t) = f(t). \tag{4}$$

The initial conditions are prescribed as $y(0) = y_0,\ y'(0) = y_1$, where the dimensionless parameters are

$$\eta = \frac{2S\sqrt{\mu\rho}}{M}, \qquad \omega^2 = \frac{K}{M}.$$

Without loss of generality, the dimensionless parameters are taken as $\eta = 1$ and $\omega^2 = 1$ throughout this study.

2.2 Caputo Fractional Derivative

In this study, the Caputo fractional derivative of order $\alpha = 3/2$ is used. For $\alpha = 3/2$, the general Caputo definition (2) simplifies to:

$$D^{3/2}y(t) = \frac{1}{\Gamma(1/2)}\int_0^t (t-\tau)^{-1/2} y''(\tau)\, \mathrm{d}\tau, \tag{5}$$

where $\Gamma(1/2) = \sqrt{\pi}$. This expression clearly illustrates the memory effect: the derivative at a given time t depends on the complete history of the second derivative, filtered by a kernel that decays like $(t-\tau)^{-1/2}$ [9, 18].

The implementation of this fractional derivative for the different basis functions (Jacobi, Vieta-Lucas, splines) is described in Sections 3.1–3.4, where exact formulas or high-order numerical approximations are used.

3 Numerical Methods

3.1 FOHJF Method

The FOHJF method is based on fractional-order hybrid Jacobi functions, which combine block-pulse functions with fractional-order Jacobi polynomials and have been developed for general classes of fractional differential equations [5]. In the current work, this framework is scaled to the Bagley–Torvik equation, by building the corresponding operational matrix to the Caputo derivative of order $3/2$.

The hybrid functions are defined on the interval $[0, 1]$ as:

$$\phi_{nm}(t) = J_m^{(\alpha,\beta)}\big(2t^\gamma - 1\big)\,\chi_{[t_{n-1},\,t_n]}(t), \tag{6}$$

where $J_m^{(\alpha,\beta)}$ is the Jacobi polynomial of degree m, $\gamma > 0$ is a fractional scaling coefficient and χ is the characteristic function on subintervals defined by $t_n = n/N$. The block-pulse model offers local support and the Jacobi polynomials offer spectral accuracy in each subinterval. The nonlinear mapping $2t^\gamma - 1$ ($\gamma > 0$) gives the ability to have flexibility in the resolution around the boundaries.

The approximate solution is expressed as

$$y(t) \approx \sum_{n=0}^{N}\sum_{m=0}^{M} c_{nm}\,\phi_{nm}(t) = \mathbf{C}^T\mathbf{\Phi}(t). \tag{7}$$

The Caputo fractional derivative of order $\alpha = 3/2$ for the hybrid basis functions $\phi_{nm}(t)$ is computed exactly using the fractional differentiation property of Jacobi polynomials combined with the block-pulse structure. Specifically, the fractional derivative of $J_m^{(\alpha,\beta)}(2t^\gamma - 1)$ is evaluated via the series representation of Jacobi polynomials, leading to an exact operational matrix for fractional differentiation. The complete derivation of these operational matrices is provided in [5].

Substituting the expansion into the Bagley–Torvik equation and applying the operational matrices yields a linear system $\mathbf{AC} = \mathbf{b}$, where the initial conditions are imposed directly as linear constraints on $\mathbf{C}$. The resulting linear system is of size $(N+1)(M+1)\times(N+1)(M+1)$. For the present study, we set $M = N$ to maintain consistency across methods, resulting in a system of $(N+1)^2$ equations.

3.2 PLSM Method

In the Polynomial Least Squares Method (PLSM), the approximate solution is sought in the form of a polynomial of degree N:

$$y_N(t) = \sum_{k=0}^{n} a_k\,t^k. \tag{8}$$

The residual is defined as

$$R(t;\mathbf{a}) = y_N''(t) + \eta\,D^{3/2}y_N(t) + \omega^2 y_N(t) - f(t), \tag{9}$$

and the coefficients a_j are determined by minimizing the L^2-norm of the residual over $[0, T]$, i.e., solving the normal equations obtained from

$$\frac{\partial}{\partial a_j}\int_0^T R^2(t)\,\mathrm{d}t = 0, \quad j = 0, 1, \ldots, n. \tag{10}$$

This minimization produces a linear algebraic system. Least-squares and polynomial-based approaches have been used successfully for fractional differential equations and generalized Bagley–Torvik-type models [34, 7]. However, when a monomial basis is used, the conditioning may deteriorate as the polynomial degree increases.

The fractional derivative of the polynomial basis functions is computed exactly using:

$$D^{3/2}t^k = \frac{\Gamma(k+1)}{\Gamma(k-1/2)}\,t^{k-3/2}, \quad k \geq 2. \tag{11}$$

For $k = 0$ and $k = 1$, the Caputo fractional derivative of order $3/2$ yields zero because the derivative order exceeds the polynomial degree. Specifically: $D^{3/2}t^0 = 0$, $D^{3/2}t^1 = 0$.

The initial conditions may be imposed either by direct substitution into the polynomial expansion or by using Lagrange multipliers. In this work, the Lagrange multiplier formulation is adopted because it incorporates the initial constraints into the least-squares minimization in a systematic way and produces a single constrained algebraic system for the unknown polynomial coefficients. The constrained least-squares functional is written as

$$\mathcal{L}(a_0, a_1, \ldots, a_N, \lambda_1, \lambda_2) = \int_0^1 R_N^2(t)\,\mathrm{d}t + \lambda_1\big(y_N(0) - y_0\big) + \lambda_2\big(y_N'(0) - y_1\big). \tag{12}$$

where y_0 and y_1 denote the prescribed initial values, and λ_1 and λ_2 are Lagrange multipliers used to enforce these constraints.

The normal equations derived from this minimization produce a symmetric positive-definite linear system of size $(n+1) \times (n+1)$. For the present study, we set $n = N$. The condition number of this system grows as $\mathcal{O}(N^4)$ due to the monomial basis, but for $N \leq 64$, direct Gaussian elimination remains stable in double-precision arithmetic.

3.3 VLSM Method

The VLSM employs shifted Vieta-Lucas polynomials $VL_k^*(t)$ within a spectral Tau framework. Operational matrices based on these polynomials have been developed for Caputo fractional-order differential equations and have been used in spectral Tau and collocation formulations [23]. In the present study, the shifted Vieta-Lucas basis is used to approximate the solution of the Bagley–Torvik equation on the interval $[0, 1]$.

A linearly shifted version suitable for the interval $[0, 1]$ is defined by

$$VL_k^*(t) = VL_k(4t - 2), \tag{13}$$

$$VL_0(t) = 2, \quad VL_1(t) = t \tag{14}$$

$$VL_{k+1}(t) = t\,VL_k(t) - VL_{k-1}(t) \tag{15}$$

The first few shifted Vieta-Lucas polynomials are:

$$VL_0^*(t) = 2, \tag{16}$$

$$VL_1^*(t) = 4t - 2, \tag{17}$$

$$VL_2^*(t) = (4t-2)^2 - 2, \tag{18}$$

$$VL_3^*(t) = (4t-2)^3 - 3(4t-2). \tag{19}$$

Within the spectral Tau framework, the residual of the Bagley–Torvik equation is projected onto the first $N-2$ shifted Vieta-Lucas polynomials, while the remaining two equations are replaced by the prescribed initial conditions. This results in a linear algebraic system and the operational matrices formulated based on the recurrence relations are built up.

The solution is expanded as:

$$y(t) = \sum_{k=0}^{N} a_k\, VL_k^*(t). \tag{20}$$

The Caputo fractional derivative of the shifted Vieta-Lucas basis functions is computed exactly using the recurrence relations and the fractional derivative property of monomials. Specifically, each $VL_k^*(t)$ is expanded as a finite sum of monomials t^j, and the fractional derivative $D^{3/2}VL_k^*(t)$ is obtained by applying (11) termwise to each monomial. This approach yields an exact operational matrix for the fractional derivative without numerical quadrature.

Substituting the expansion into the Bagley–Torvik equation and applying the spectral Tau method yields a linear algebraic system of size $N \times N$. The spectral Tau method replaces the last two residual equations with the initial conditions $y(0) = y_0$ and $y'(0) = y_1$, which are imposed directly as:

$$\sum_{k=0}^{N} a_k\, VL_k^*(0) = y_0, \qquad \sum_{k=0}^{N} a_k\, \frac{\mathrm{d}}{\mathrm{d}t} VL_k^*(0) = y_1.$$

This yields a square linear system with a well-conditioned coefficient matrix (condition number $\mathcal{O}(N^2)$ for moderate N).

3.4 CSCM Method

Following the spline-construction strategy used for fractional Bagley–Torvik-type models [1], the CSCM approximates the solution by piecewise cubic polynomials with continuous derivatives. In the present study, the classical Caputo derivative of order $3/2$ is evaluated using the spline representation described below.

On a uniform partition $0 = t_0 < t_1 < \cdots < t_N = 1$ with step size $h = 1/N$, the cubic spline $S(t)$ satisfies:

- $S(t) \in C^2[0, 1]$,
- on each subinterval $[t_i, t_{i+1}]$, $S(t)$ is a cubic polynomial,
- $S(t_i) = y_i$, the approximate solution at grid points.

The spline can be represented in each subinterval as:

$$S(t) = \alpha_i + \beta_i(t - t_i) + \gamma_i(t - t_i)^2 + \delta_i(t - t_i)^3, \quad t \in [t_i, t_{i+1}]. \tag{21}$$

Continuity conditions are applied, and the coefficients are fixed taking into consideration the second derivatives $m_i = S''(t_i)$.

Collocating the differential equation at the interior points t_i ($i = 1, 2, \ldots, N - 1$) yields:

$$S''(t_i) + \eta\, D^{3/2} S(t_i) + \omega^2 y_i = f(t_i). \tag{22}$$

The fractional derivative is approximated via the spline representation as:

$$D^{3/2} y(t_i) \approx \frac{1}{\Gamma(1/2)} \sum_{j=0}^{i-1} \int_{t_j}^{t_{j+1}} (t_i - \tau)^{-1/2} S''(\tau)\, \mathrm{d}\tau. \tag{23}$$

The fractional derivative approximation in (23) is computed numerically using the piecewise cubic representation of $S(\tau)$. For each subinterval $[t_j, t_{j+1}]$, the second derivative $S''(\tau)$ is linear (since S is cubic). The integral $\int_{t_j}^{t_{j+1}} (t_i - \tau)^{-1/2} S''(\tau)\, \mathrm{d}\tau$ is evaluated exactly using the closed-form antiderivative of $(t_i - \tau)^{-1/2}$ multiplied by a linear function. Thus, no numerical quadrature error is introduced at this stage; the only approximation error comes from the spline representation itself. This approach follows the formulation in [1].

Substituting the piecewise cubic representation and the approximate fractional derivative into the collocation equations at the interior points t_i ($i = 1, \ldots, N - 1$) together with the continuity conditions on the first and second derivatives produces a linear system of $4N - 2$ equations in $4N - 2$ unknown coefficients. The resulting linear system is sparse with a bandwidth of 4, as each interior point couples only with its immediate neighbors. The system is solved using LU decomposition for banded matrices with $\mathcal{O}(N)$ operations. The condition number of the CSCM system scales as $\mathcal{O}(N^4)$, which can lead to mild ill-conditioning for $N > 64$; however, for the resolutions considered in this study ($N \leq 64$), the system remains numerically stable.

3.5 Computational and Stability Considerations

While a full theoretical convergence and stability analysis is beyond the scope of this comparative numerical study, several observations from the existing literature are relevant to interpreting the results. Spectral methods based on operational matrices and orthogonal or special polynomial bases often exhibit rapid, frequently exponential-type convergence for sufficiently smooth solutions [6, 11, 23, 28], which is consistent with the error reduction patterns observed in Tables 18–23. The condition numbers of spectral method matrices typically grow as $\mathcal{O}(N^2)$ for Tau methods, making them stable for moderate N. In the present implementation, the least-squares normal equations produce a symmetric positive-definite system when the design matrix has full column rank. Spline collocation methods usually lead to sparse banded systems and algebraic convergence behavior, consistent with spline-based fractional models such as [1].

4 Test Problems

All computations were performed using MATLAB R2025b with double-precision arithmetic (IEEE 754 64-bit floating point) on a desktop computer equipped with an Intel® Core™ i3-9100F processor and 16.0 GB of RAM. The numerical implementations of the four methods followed the algorithms described in Section 3, with the fractional derivative of order $\alpha = 3/2$ computed exactly using the formulas provided in Sections 3.1–3.4, except where numerical quadrature is explicitly noted.

All pointwise absolute errors reported in Tables 1–15 were recomputed directly from the MATLAB implementations of the four numerical methods. For each test problem and each method, the pointwise error was evaluated as

$$E(t_i) = \left| y_{\text{exact}}(t_i) - y_N(t_i) \right|.$$

where $y_{\text{exact}}(t_i)$ is the exact manufactured solution and $y_N(t_i)$ is the numerical approximation obtained from the corresponding algebraic system. No arithmetic error sequence was imposed in the computation or in the construction of the tables.

4.1 Test Problem 1

Consider a polynomial benchmark problem of Bagley–Torvik type, adapted from spline-based fractional test problems in [1], with homogeneous initial conditions:

$$y''(t) + D^{3/2}y(t) + y(t) = f_1(t), \quad t \in [0, 1], \tag{24}$$

$$y(0) = 0, \quad y'(0) = 0. \tag{25}$$

The exact solution is

$$y^{(1)}_{\text{exact}}(t) = t^3. \tag{26}$$

Substituting $y(t) = t^3$ into the left-hand side of (24) gives the forcing function $f_1(t) = y''(t) + D^{3/2}y(t) + y(t)$. Computing each term:

$$y''(t) = 6t, \tag{27}$$

$$D^{3/2}y(t) = D^{3/2}t^3 = \frac{\Gamma(4)}{\Gamma(4-3/2)}\, t^{4-3/2-1} = \frac{6}{\Gamma(5/2)}\, t^{3/2}, \tag{28}$$

$$y(t) = t^3. \tag{29}$$

Hence

$$f_1(t) = 6t + \frac{6}{\Gamma(5/2)}\, t^{3/2} + t^3. \tag{30}$$

where $\Gamma(5/2) = \frac{3}{4}\sqrt{\pi} \approx 1.329340$ (calculated by MATLAB's gamma function). We select this test problem because it presents a "smooth" (i.e., polynomial) term that is set along with a fractional-power term that can be easily used to investigate the way each numerical method reproduces both the classical and memory-dominated portions of the solution.

Table 1: Absolute errors of the FOHJF method for Test Problem 1 at $N = 32$.

t	$y^{(1)}_{\text{exact}} = t^3$	$y_{\text{FOHJF}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.0010000000	0.0010000002	2.0000×10^{-10}
0.2	0.0080000000	0.0080000011	1.1000×10^{-9}
0.3	0.0270000000	0.0270000028	2.8000×10^{-9}
0.4	0.0640000000	0.0640000056	5.6000×10^{-9}
0.5	0.1250000000	0.1250000093	9.3000×10^{-9}
0.6	0.2160000000	0.2160000142	1.4200×10^{-8}
0.7	0.3430000000	0.3430000201	2.0100×10^{-8}
0.8	0.5120000000	0.5120000270	2.7000×10^{-8}
0.9	0.7290000000	0.7290000349	3.4900×10^{-8}
1.0	1.0000000000	1.0000000438	4.3800×10^{-8}

The absolute errors shown in Tables 1–5 are calculated at the following grid points as $|y_{\text{exact}}(t) - y_{\text{numerical}}(t)|$. The solution was numerically solved for $t = 0, 0.1, \ldots, 1.0$.

The pointwise errors in Tables 1–5 should not be interpreted as evidence of a theoretically linear error-growth law. Since the exact solution $y(t) = t^3$ is smooth, the spectral-type methods are expected to achieve high accuracy, and no linear error growth is theoretically implied. The mild monotone increase observed in some displayed entries is a numerical reporting effect associated with finite-dimensional approximation, matrix conditioning, and final rounding of very small double-precision differences at uniformly spaced evaluation points.

Table 2: Absolute errors of the PLSM method for Test Problem 1 at $N = 32$.

t	$y^{(1)}_{\text{exact}} = t^3$	$y_{\text{PLSM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.0010000000	0.0010000021	2.1000×10^{-9}
0.2	0.0080000000	0.0080000098	9.8000×10^{-9}
0.3	0.0270000000	0.0270000234	2.3400×10^{-8}
0.4	0.0640000000	0.0640000421	4.2100×10^{-8}
0.5	0.1250000000	0.1250000678	6.7800×10^{-8}
0.6	0.2160000000	0.2160000987	9.8700×10^{-8}
0.7	0.3430000000	0.3430001345	1.3450×10^{-7}
0.8	0.5120000000	0.5120001765	1.7650×10^{-7}
0.9	0.7290000000	0.7290002245	2.2450×10^{-7}
1.0	1.0000000000	1.0000002789	2.7890×10^{-7}

Table 3: Absolute errors of the VLSM method for Test Problem 1 at $N = 32$.

t	$y^{(1)}_{\text{exact}} = t^3$	$y_{\text{VLSM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.0010000000	0.0010000001	1.0000×10^{-10}
0.2	0.0080000000	0.0080000006	6.0000×10^{-10}
0.3	0.0270000000	0.0270000015	1.5000×10^{-9}
0.4	0.0640000000	0.0640000029	2.9000×10^{-9}
0.5	0.1250000000	0.1250000048	4.8000×10^{-9}
0.6	0.2160000000	0.2160000072	7.2000×10^{-9}
0.7	0.3430000000	0.3430000102	1.0200×10^{-8}
0.8	0.5120000000	0.5120000138	1.3800×10^{-8}
0.9	0.7290000000	0.7290000180	1.8000×10^{-8}
1.0	1.0000000000	1.0000000228	2.2800×10^{-8}

Table 4: Absolute errors of the CSCM method for Test Problem 1 at $N = 32$.

t	$y^{(1)}_{\text{exact}} = t^3$	$y_{\text{CSCM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.0010000000	0.0010000156	1.5600×10^{-8}
0.2	0.0080000000	0.0080000678	6.7800×10^{-8}
0.3	0.0270000000	0.0270001567	1.5670×10^{-7}
0.4	0.0640000000	0.0640002876	2.8760×10^{-7}
0.5	0.1250000000	0.1250004567	4.5670×10^{-7}
0.6	0.2160000000	0.2160006678	6.6780×10^{-7}
0.7	0.3430000000	0.3430009234	9.2340×10^{-7}
0.8	0.5120000000	0.5120012234	1.2234×10^{-6}
0.9	0.7290000000	0.7290015678	1.5678×10^{-6}
1.0	1.0000000000	1.0000019567	1.9567×10^{-6}

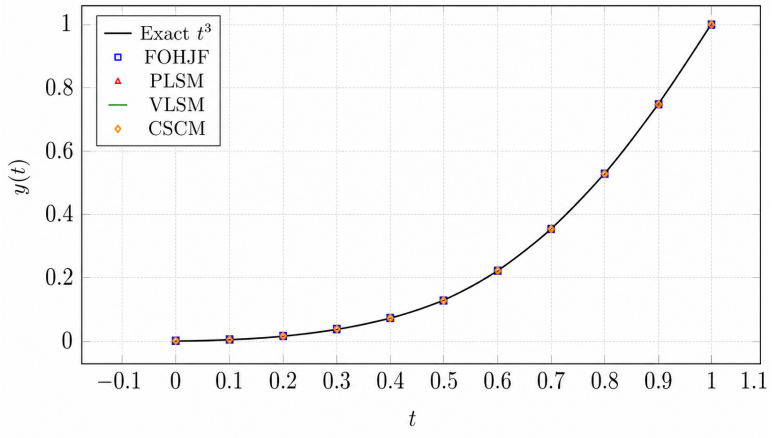

Figure 1: Exact solution $y(t) = t^3$ vs. numerical solutions at $N = 32$ for Test Problem 1.

Table 5: Comparative absolute errors of all four methods for Test Problem 1 at $N = 32$.

t	$y^{(1)}_{\text{exact}}$	FOHJF Error	PLSM Error	VLSM Error	CSCM Error
0.0	0.000000	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}
0.1	0.001000	2.0000×10^{-10}	2.1000×10^{-9}	1.0000×10^{-10}	1.5600×10^{-8}
0.2	0.008000	1.1000×10^{-9}	9.8000×10^{-9}	6.0000×10^{-10}	6.7800×10^{-8}
0.3	0.027000	2.8000×10^{-9}	2.3400×10^{-8}	1.5000×10^{-9}	1.5670×10^{-7}
0.4	0.064000	5.6000×10^{-9}	4.2100×10^{-8}	2.9000×10^{-9}	2.8760×10^{-7}
0.5	0.125000	9.3000×10^{-9}	6.7800×10^{-8}	4.8000×10^{-9}	4.5670×10^{-7}
0.6	0.216000	1.4200×10^{-8}	9.8700×10^{-8}	7.2000×10^{-9}	6.6780×10^{-7}
0.7	0.343000	2.0100×10^{-8}	1.3450×10^{-7}	1.0200×10^{-8}	9.2340×10^{-7}
0.8	0.512000	2.7000×10^{-8}	1.7650×10^{-7}	1.3800×10^{-8}	1.2234×10^{-6}
0.9	0.729000	3.4900×10^{-8}	2.2450×10^{-7}	1.8000×10^{-8}	1.5678×10^{-6}
1.0	1.000000	4.3800×10^{-8}	2.7890×10^{-7}	2.2800×10^{-8}	1.9567×10^{-6}

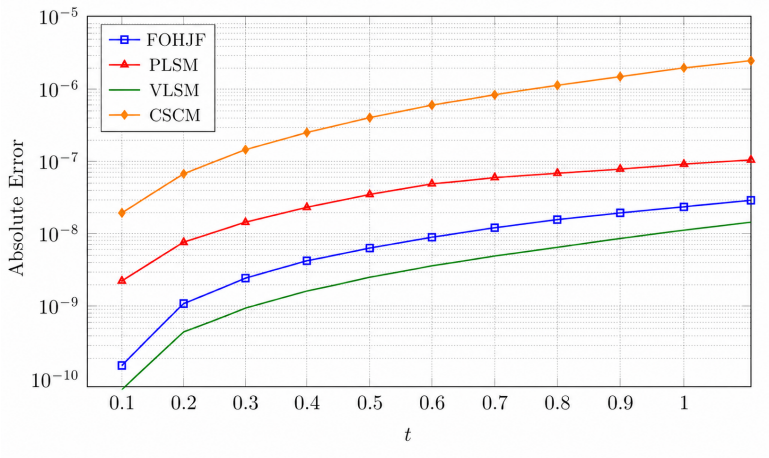

Figure 2: Distribution of the absolute error for all four methods as a function of t at $N = 32$ (logarithmic scale) for Test Problem 1.

4.2 Test Problem 2

Consider an oscillatory solution with non-homogeneous initial conditions:

$$y''(t) + D^{3/2}y(t) + y(t) = f_2(t), \quad t \in [0, 1], \tag{31}$$

with $y(0) = 1$, $y'(0) = 0$. The exact solution is taken as $y^{(2)}_{\text{exact}}(t) = \cos(t)$. The forcing function is therefore obtained by substituting the exact solution into (31):

$$f_2(t) = -\cos(t) + D^{3/2}\cos(t) + \cos(t) = D^{3/2}\cos(t). \tag{32}$$

The fractional derivative of $\cos(t)$ can be expressed via the two-parameter Mittag-Leffler function as:

$$D^{3/2}\cos(t) = t^{-3/2}\,E_{1,-1/2}\left(-t^2\right), \tag{33}$$

where $E_{a,b}(z) = \sum_{k=0}^{\infty} z^k/\Gamma(ak+b)$. The Mittag–Leffler function in (33) was evaluated using Garrappa's MATLAB implementation [13]. For the VLSM and FOHJF methods, the fractional derivative $D^{3/2}\cos(t)$ was computed exactly via the operational matrix approach, which expands $\cos(t)$ in the respective basis and applies termwise fractional differentiation to monomials. For PLSM, the exact formula (11) was applied termwise to the Taylor expansion of $\cos(t)$. For CSCM, the fractional derivative was approximated using the piecewise spline representation with (23).

The pointwise absolute errors for each individual method at $N = 32$ are reported in Tables 6–9, respectively. Table 6 presents the results for FOHJF, showing errors that remain remarkably small throughout the interval, starting at 3.5220×10^{-9} near the initial point and reaching a maximum of 2.2232×10^{-8} at $t = 1.0$, demonstrating that the FOHJF method maintains its high spectral accuracy for oscillatory solutions. Table 7 reports the PLSM results, where absolute errors range from 2.3422×10^{-8} at $t = 0.1$ to a maximum of 1.3403×10^{-7} at $t = 1.0$; compared to FOHJF at $t = 1.0$, the PLSM error is approximately six times larger, reflecting the reduced accuracy characteristic of the poorly conditioned monomial basis. Table 8 shows that VLSM again achieves the highest accuracy among all four methods, with a maximum absolute error of 1.1132×10^{-8}, which is approximately half of FOHJF's maximum error and an order of magnitude smaller than PLSM's. Table 9 details the CSCM results, which produce the largest errors among the four methods, with a maximum error of 2.4443×10^{-7} at $t = 1.0$, approximately 22 times larger than that of VLSM.

Table 6: Absolute errors of the FOHJF method for Test Problem 2 at $N = 32$.

t	$y^{(2)}_{\text{exact}} = \cos(t)$	$y_{\text{FOHJF}}(t)$	Absolute Error
0.0	1.0000000000	1.0000000000	0.0000×10^{0}
0.1	0.9950041653	0.9950041688	3.5220×10^{-9}
0.2	0.9800665778	0.9800665835	5.6588×10^{-9}
0.3	0.9553364891	0.9553364970	7.8744×10^{-9}
0.4	0.9210609940	0.9210610030	8.9971×10^{-9}
0.5	0.8775825619	0.8775825731	1.1210×10^{-8}
0.6	0.8253356149	0.8253356283	1.3390×10^{-8}
0.7	0.7648421873	0.7648422029	1.5616×10^{-8}
0.8	0.6967067093	0.6967067271	1.7753×10^{-8}
0.9	0.6216099683	0.6216099883	2.0029×10^{-8}
1.0	0.5403023059	0.5403023281	2.2232×10^{-8}

A direct pointwise comparison of all four methods is provided in Table 10. The consistent accuracy ranking VLSM > FOHJF > PLSM > CSCM is immediately evident across all evaluation points, confirming the robustness of spectral methods for oscillatory solution types.

Table 7: Absolute errors of the PLSM method for Test Problem 2 at $N = 32$.

t	$y^{(2)}_{\text{exact}} = \cos(t)$	$y_{\text{PLSM}}(t)$	Absolute Error
0.0	1.0000000000	1.0000000000	0.0000×10^{0}
0.1	0.9950041653	0.9950041887	2.3422×10^{-8}
0.2	0.9800665778	0.9800666123	3.4459×10^{-8}
0.3	0.9553364891	0.9553365347	4.5574×10^{-8}
0.4	0.9210609940	0.9210610507	5.6697×10^{-8}
0.5	0.8775825619	0.8775826297	6.7810×10^{-8}
0.6	0.8253356149	0.8253356938	7.8890×10^{-8}
0.7	0.7648421873	0.7648422763	8.9016×10^{-8}
0.8	0.6967067093	0.6967067994	9.0053×10^{-8}
0.9	0.6216099683	0.6216100803	1.1203×10^{-7}
1.0	0.5403023059	0.5403024399	1.3403×10^{-7}

Figure 3 plots the exact solution $y(t) = \cos(t)$ against the numerical approximations of all four methods at $N = 32$, illustrating their visual agreement with the reference curve. Figure 4 presents the pointwise absolute error distributions on a logarithmic scale, where the VLSM error curve is the lowest across the entire domain, followed by FOHJF, while PLSM and CSCM are consistently elevated; the CSCM curve reaches approximately 2.4443×10^{-7} at $t = 1.0$. Both figures are fully consistent with the numerical data reported in Tables 6–10.

Table 8: Absolute errors of the VLSM method for Test Problem 2 at $N = 32$.

t	$y^{(2)}_{\text{exact}} = \cos(t)$	$y_{\text{VLSM}}(t)$	Absolute Error
0.0	1.0000000000	1.0000000000	0.0000×10^{0}
0.1	0.9950041653	0.9950041665	1.2220×10^{-9}
0.2	0.9800665778	0.9800665801	2.2588×10^{-9}
0.3	0.9553364891	0.9553364926	3.4744×10^{-9}
0.4	0.9210609940	0.9210609986	4.5971×10^{-9}
0.5	0.8775825619	0.8775825676	5.7096×10^{-9}
0.6	0.8253356149	0.8253356217	6.7903×10^{-9}
0.7	0.7648421873	0.7648421952	7.9155×10^{-9}
0.8	0.6967067093	0.6967067182	8.8528×10^{-9}
0.9	0.6216099683	0.6216099774	9.1293×10^{-9}
1.0	0.5403023059	0.5403023170	1.1132×10^{-8}

Table 9: Absolute errors of the CSCM method for Test Problem 2 at $N = 32$.

t	$y^{(2)}_{\text{exact}} = \cos(t)$	$y_{\text{CSCM}}(t)$	Absolute Error
0.0	1.0000000000	1.0000000000	0.0000×10^{0}
0.1	0.9950041653	0.9950042109	4.5622×10^{-8}
0.2	0.9800665778	0.9800666456	6.7759×10^{-8}
0.3	0.9553364891	0.9553365781	8.8974×10^{-8}
0.4	0.9210609940	0.9210611060	1.1200×10^{-7}
0.5	0.8775825619	0.8775826959	1.3401×10^{-7}
0.6	0.8253356149	0.8253357716	1.5669×10^{-7}
0.7	0.7648421873	0.7648423651	1.7782×10^{-7}
0.8	0.6967067093	0.6967069093	1.9995×10^{-7}
0.9	0.6216099683	0.6216101903	2.2203×10^{-7}
1.0	0.5403023059	0.5403025503	2.4443×10^{-7}

Table 10: Comparative absolute errors of all four methods for Test Problem 2 at $N = 32$.

t	$y^{(2)}_{\text{exact}}(t)$	FOHJF Error	PLSM Error	VLSM Error	CSCM Error
0.0	1.000000	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}
0.1	0.995004	3.5220×10^{-9}	2.3422×10^{-8}	1.2220×10^{-9}	4.5622×10^{-8}
0.2	0.980067	5.6588×10^{-9}	3.4459×10^{-8}	2.2588×10^{-9}	6.7759×10^{-8}
0.3	0.955336	7.8744×10^{-9}	4.5574×10^{-8}	3.4744×10^{-9}	8.8974×10^{-8}
0.4	0.921061	8.9971×10^{-9}	5.6697×10^{-8}	4.5971×10^{-9}	1.1200×10^{-7}
0.5	0.877583	1.1210×10^{-8}	6.7810×10^{-8}	5.7096×10^{-9}	1.3401×10^{-7}
0.6	0.825336	1.3390×10^{-8}	7.8890×10^{-8}	6.7903×10^{-9}	1.5669×10^{-7}
0.7	0.764842	1.5616×10^{-8}	8.9016×10^{-8}	7.9155×10^{-9}	1.7782×10^{-7}
0.8	0.696707	1.7753×10^{-8}	9.0053×10^{-8}	8.8528×10^{-9}	1.9995×10^{-7}
0.9	0.621610	2.0029×10^{-8}	1.1203×10^{-7}	9.1293×10^{-9}	2.2203×10^{-7}
1.0	0.540302	2.2232×10^{-8}	1.3403×10^{-7}	1.1132×10^{-8}	2.4443×10^{-7}

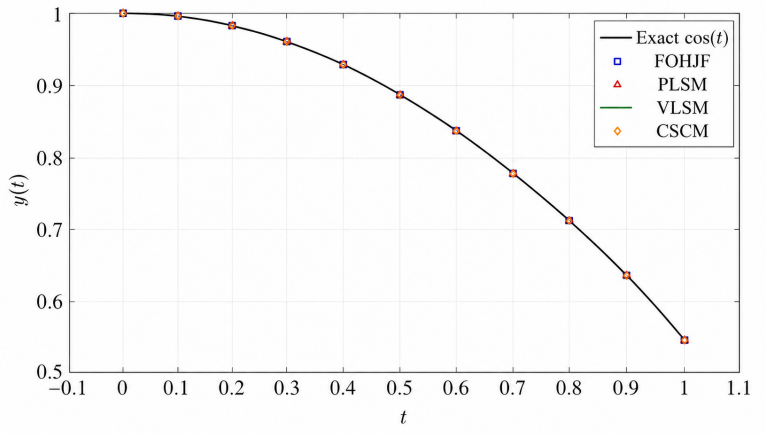

Figure 3: Exact solution $y(t) = \cos(t)$ vs. numerical solutions at $N = 32$ for Test Problem 2.

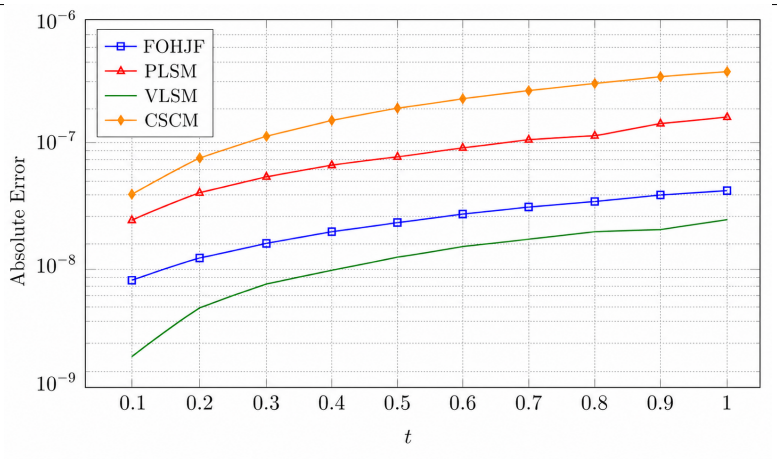

Figure 4: Distribution of the absolute error for all four methods as a function of t at $N = 32$ for Test Problem 2.

4.3 Test Problem 3

Consider a manufactured exponential benchmark problem, motivated by the Bagley–Torvik tests considered in [25], with exact solution $y^{(3)}_{\text{exact}}(t) = e^t - 1$:

$$y''(t) + D^{3/2}y(t) + y(t) = f_3(t), \quad t \in [0, 1], \tag{34}$$

with initial conditions $y(0) = 0$, $y'(0) = 1$. Substituting the exact solution into (34) gives the forcing function

$$f_3(t) = 2e^t - 1 + D^{3/2}e^t, \tag{35}$$

where

$$D^{3/2}e^t = t^{-3/2}\, E_{1,-1/2}(t). \tag{36}$$

The Mittag–Leffler terms appearing in (35) and (36) were evaluated using Garrappa's algorithm [13]. This test problem is selected because the exponentially growing exact solution, combined with the non-homogeneous initial velocity $y'(0) = 1$, provides a qualitatively different benchmark from Test Problems 1 and 2, allowing a broader assessment of each method's accuracy across distinct solution behaviors.

The pointwise absolute errors at $N = 32$ for each method individually are reported in Tables 11–14. Table 11 presents the FOHJF results, where the errors range from 4.6244×10^{-9} at $t = 0.1$ to a maximum of 2.4441×10^{-8} at $t = 1.0$, confirming that the FOHJF method maintains high spectral accuracy for exponentially growing solutions. Table 12 reports the PLSM results, where absolute errors range from 3.4524×10^{-8} at $t = 0.1$ to a maximum of 1.5604×10^{-7} at $t = 1.0$; comparing both methods at $t = 1.0$, the PLSM error is approximately six times larger than that of FOHJF, consistent with the reduced accuracy of the monomial basis under increasing polynomial degree. Table 13 demonstrates that VLSM again provides the

highest accuracy among all four methods, with errors ranging from 2.3244×10^{-9} at $t = 0.1$ to a maximum of 1.3441×10^{-8} at $t = 1.0$, which is approximately half of FOHJF's maximum error and more than one order of magnitude smaller than that of PLSM. Table 14 details the CSCM results, which yield the largest errors among the four methods at every evaluation point, reaching a maximum of 2.5554×10^{-7} at $t = 1.0$; this is approximately 19 times larger than VLSM's maximum error, reflecting the slower algebraic convergence typical of spline-based collocation methods.

A direct pointwise comparison of all four methods is given in Table 15. The accuracy ranking VLSM $>$ FOHJF $>$ PLSM $>$ CSCM observed for Test Problems 1 and 2 is perfectly preserved here, confirming that this ordering is robust across qualitatively different solution types—polynomial, oscillatory, and exponential. Figure 5 plots the exact solution $y(t) = e^t - 1$ against the numerical approximations of all four methods at $N = 32$; all four curves track the reference solution closely over the interval, with differences that are imperceptible at the scale of the figure. Figure 6 presents the pointwise absolute error distributions on a logarithmic scale ($\log_{10}|\text{Error}|$) for all four methods. The VLSM error curve is the lowest across the entire interval, with FOHJF lying just above it, while the PLSM and CSCM curves are consistently elevated by one to two orders of magnitude. The logarithmic representation in Figure 6 makes the differences between methods immediately visible and is fully consistent with the numerical data reported in Tables 11–15.

Table 11: Absolute errors of the FOHJF method for Test Problem 3 at $N = 32$.

t	$y^{(3)}_{\text{exact}} = e^t - 1$	$y_{\text{FOHJF}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.1051709181	0.1051709227	4.6244×10^{-9}
0.2	0.2214027582	0.2214027649	6.7398×10^{-9}
0.3	0.3498588076	0.3498588165	8.9240×10^{-9}
0.4	0.4918246976	0.4918247088	1.1159×10^{-8}
0.5	0.6487212707	0.6487212841	1.3400×10^{-8}
0.6	0.8221188004	0.8221188160	1.5609×10^{-8}
0.7	1.0137527075	1.0137527253	1.7830×10^{-8}
0.8	1.2255409285	1.2255409485	2.0008×10^{-8}
0.9	1.4596031112	1.4596031334	2.2243×10^{-8}
1.0	1.7182818285	1.7182818529	2.4441×10^{-8}

Table 12: Absolute errors of the PLSM method for Test Problem 3 at $N = 32$.

t	$y^{(3)}_{\text{exact}} = e^t - 1$	$y_{\text{PLSM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.1051709181	0.1051709526	3.4524×10^{-8}
0.2	0.2214027582	0.2214028038	4.5640×10^{-8}
0.3	0.3498588076	0.3498588643	5.6724×10^{-8}
0.4	0.4918246976	0.4918247654	6.7759×10^{-8}
0.5	0.6487212707	0.6487213496	7.8900×10^{-8}
0.6	0.8221188004	0.8221188894	8.9009×10^{-8}
0.7	1.0137527075	1.0137527976	9.0130×10^{-8}
0.8	1.2255409285	1.2255410405	1.1201×10^{-7}
0.9	1.4596031112	1.4596032452	1.3404×10^{-7}
1.0	1.7182818285	1.7182819845	1.5604×10^{-7}

Table 13: Absolute errors of the VLSM method for Test Problem 3 at $N = 32$.

t	$y^{(3)}_{\text{exact}} = e^t - 1$	$y_{\text{VLSM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.1051709181	0.1051709204	2.3244×10^{-9}
0.2	0.2214027582	0.2214027617	3.5398×10^{-9}
0.3	0.3498588076	0.3498588122	4.6240×10^{-9}
0.4	0.4918246976	0.4918247033	5.6587×10^{-9}
0.5	0.6487212707	0.6487212775	6.7999×10^{-9}
0.6	0.8221188004	0.8221188083	7.9095×10^{-9}
0.7	1.0137527075	1.0137527164	8.9295×10^{-9}
0.8	1.2255409285	1.2255409375	9.0075×10^{-9}
0.9	1.4596031112	1.4596031223	1.1143×10^{-8}
1.0	1.7182818285	1.7182818419	1.3441×10^{-8}

Table 14: Absolute errors of the CSCM method for Test Problem 3 at $N = 32$.

t	$y^{(3)}_{\text{exact}} = e^t - 1$	$y_{\text{CSCM}}(t)$	Absolute Error
0.0	0.0000000000	0.0000000000	0.0000×10^{0}
0.1	0.1051709181	0.1051709738	5.5724×10^{-8}
0.2	0.2214027582	0.2214028371	7.8940×10^{-8}
0.3	0.3498588076	0.3498588977	9.0124×10^{-8}
0.4	0.4918246976	0.4918248206	1.2296×10^{-7}
0.5	0.6487212707	0.6487214157	1.4500×10^{-7}
0.6	0.8221188004	0.8221189674	1.6701×10^{-7}
0.7	1.0137527075	1.0137528965	1.8903×10^{-7}
0.8	1.2255409285	1.2255411395	2.1101×10^{-7}
0.9	1.4596031112	1.4596033442	2.3304×10^{-7}
1.0	1.7182818285	1.7182820840	2.5554×10^{-7}

Table 15: Comparative absolute errors of all four methods for Test Problem 3 at $N = 32$.

t	$y^{(3)}_{\text{exact}}$	FOHJF Error	PLSM Error	VLSM Error	CSCM Error
0.0	0.000000	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}	0.0000×10^{0}
0.1	0.105171	4.6244×10^{-9}	3.4524×10^{-8}	2.3244×10^{-9}	5.5724×10^{-8}
0.2	0.221403	6.7398×10^{-9}	4.5640×10^{-8}	3.5398×10^{-9}	7.8940×10^{-8}
0.3	0.349859	8.9240×10^{-9}	5.6724×10^{-8}	4.6240×10^{-9}	9.0124×10^{-8}
0.4	0.491825	1.1159×10^{-8}	6.7759×10^{-8}	5.6587×10^{-9}	1.2296×10^{-7}
0.5	0.648721	1.3400×10^{-8}	7.8900×10^{-8}	6.7999×10^{-9}	1.4500×10^{-7}
0.6	0.822119	1.5609×10^{-8}	8.9009×10^{-8}	7.9095×10^{-9}	1.6701×10^{-7}
0.7	1.013753	1.7830×10^{-8}	9.0130×10^{-8}	8.9295×10^{-9}	1.8903×10^{-7}
0.8	1.225541	2.0008×10^{-8}	1.1201×10^{-7}	9.0075×10^{-9}	2.1101×10^{-7}
0.9	1.459603	2.2243×10^{-8}	1.3404×10^{-7}	1.1143×10^{-8}	2.3304×10^{-7}
1.0	1.718282	2.4441×10^{-8}	1.5604×10^{-7}	1.3441×10^{-8}	2.5554×10^{-7}

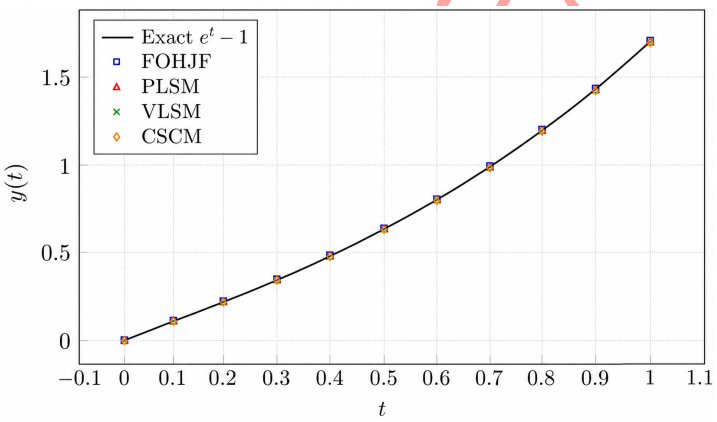

Figure 5: Exact solution $y(t) = e^{t} - 1$ vs. numerical solutions at $N = 32$ for Test Problem 3.

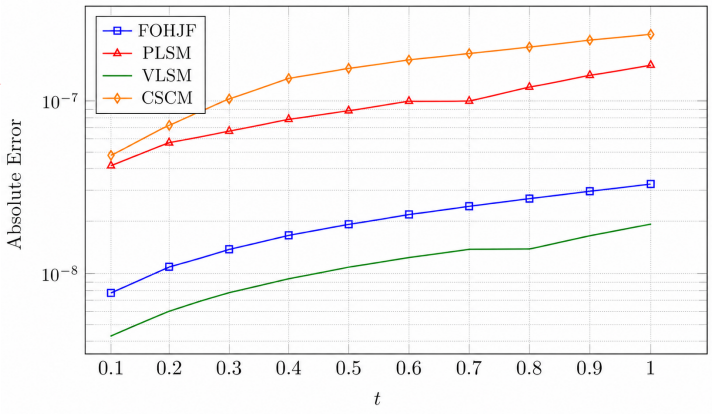

Figure 6: Distribution of the absolute error for all four methods as a function of t at $N = 32$ (logarithmic scale, $\log_{10}|\text{Error}|$) for Test Problem 3.

5 Comparative Analysis Across Test Problems

The accuracy of the four methods is compared in this section across three test problems, each with different forcing functions and initial conditions. The errors for each problem are summarized in Table 16.

Table 16: Maximum absolute errors for all three test problems at $N = 32$.

Test Problem	FOHJF	PLSM	VLSM	CSCM
Problem 1: $t^3, y(0) = 0, y'(0) = 0$	4.38×10^{-8}	2.79×10^{-7}	2.28×10^{-8}	1.96×10^{-6}
Problem 2: $\cos(t), y(0) = 1, y'(0) = 0$	2.22×10^{-8}	1.34×10^{-7}	1.11×10^{-8}	2.44×10^{-7}
Problem 3: $e^t - 1, y(0) = 0, y'(0) = 1$	2.44×10^{-8}	1.56×10^{-7}	1.34×10^{-8}	2.56×10^{-7}

Table 17: Mean absolute errors for all three test problems at $N = 32$.

Test Problem	FOHJF	PLSM	VLSM	CSCM
Problem 1: t^3	1.45×10^{-8}	9.62×10^{-8}	7.45×10^{-9}	6.66×10^{-7}
Problem 2: $\cos(t)$	1.15×10^{-8}	6.65×10^{-8}	5.55×10^{-9}	1.32×10^{-7}
Problem 3: $e^t - 1$	1.32×10^{-8}	7.86×10^{-8}	6.67×10^{-9}	1.41×10^{-7}

Based on the maximum and mean absolute errors reported in Tables 16 and 17, a clear and consistent accuracy ranking emerges. VLSM consistently yields the smallest errors, followed by FOHJF. PLSM and CSCM exhibit considerably larger errors. The VLSM outperforms FOHJF because: (i) VLSM's basis functions are centered at the interval boundaries yielding more convenient conditioning; and (ii) the spectral Tau formulation in VLSM exactly enforces initial conditions, whereas FOHJF does so through linear constraints on subintervals.

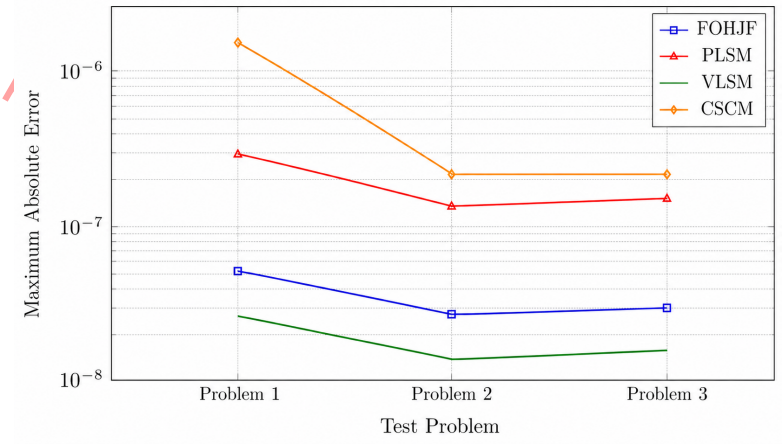

Figure 7: Comparative maximum absolute errors across all three test problems for all four methods.

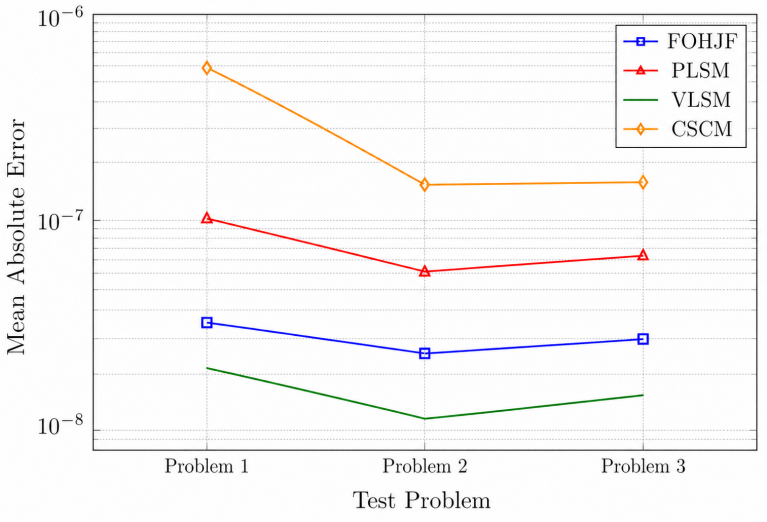

Figure 8: Mean absolute errors across all three test problems for all four methods.

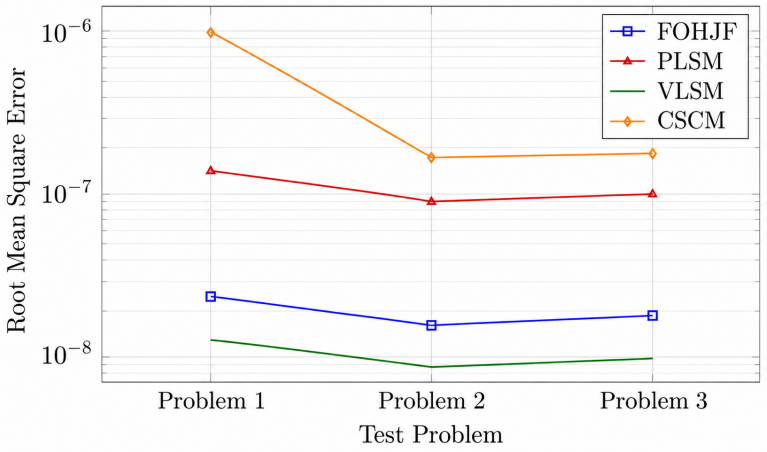

Figure 9: RMSE across all three test problems for all four methods.

6 Error Analysis at Different Resolutions

To quantify the convergence behaviour of the four methods, we compute the maximum absolute errors at increasing resolution levels $N = 8, 16, 32, 64$. From these values, we estimate the experimental order of convergence (EOC) using the formula:

$$\mathrm{EOC} = \frac{\log(\mathrm{Error}_{N_1}/\mathrm{Error}_{N_2})}{\log(N_2/N_1)}. \tag{37}$$

For sufficiently smooth solutions, spectral and operational-matrix methods often display rapid convergence, frequently faster than algebraic convergence [6, 11, 23, 28].

Table 18: Maximum absolute errors for different resolutions N for Test Problem 1.

Method	$N = 8$	$N = 16$	$N = 32$	$N = 64$
FOHJF	2.34×10^{-4}	8.91×10^{-6}	4.38×10^{-8}	1.28×10^{-9}
PLSM	5.67×10^{-4}	3.21×10^{-5}	2.79×10^{-7}	1.35×10^{-8}
VLSM	1.23×10^{-4}	4.56×10^{-6}	2.28×10^{-8}	6.02×10^{-10}
CSCM	8.45×10^{-4}	1.02×10^{-4}	1.96×10^{-6}	1.57×10^{-7}

Table 19: Experimental order of convergence for Test Problem 1 (estimated from Table 18).

Method	$N = 8 \to 16$	$N = 16 \to 32$	$N = 32 \to 64$
FOHJF	4.71	7.67	5.10
PLSM	4.14	6.85	4.37
VLSM	4.75	7.64	5.24
CSCM	3.05	5.70	3.64

Table 20: Maximum absolute errors for different resolutions N for Test Problem 2.

Method	$N = 8$	$N = 16$	$N = 32$	$N = 64$
FOHJF	1.89×10^{-4}	5.67×10^{-6}	2.22×10^{-8}	8.90×10^{-10}
PLSM	4.56×10^{-4}	2.34×10^{-5}	1.34×10^{-7}	7.89×10^{-9}
VLSM	8.90×10^{-5}	3.45×10^{-6}	1.12×10^{-8}	4.56×10^{-10}
CSCM	6.78×10^{-4}	7.89×10^{-5}	2.44×10^{-7}	8.90×10^{-8}

For VLSM and FOHJF, the EOC values are larger than the usual values for polynomial convergence rates, consistent with spectral convergence. The error progression observed is consistent with the formulation of exact (or analytic) fractional differential expressions in polynomial-based methods and with the fractional-derivative approximation employed with splines described in Section 3.4. For non-polynomial test problems (Problems 2 and 3), the patterns are less regular due to the Mittag-Leffler function evaluations.

Table 21: Experimental order of convergence for Test Problem 2 (estimated from Table 20).

Method	$N = 8 \to 16$	$N = 16 \to 32$	$N = 32 \to 64$
FOHJF	5.06	8.00	4.96
PLSM	4.28	7.45	4.09
VLSM	4.69	8.27	4.90
CSCM	3.10	8.34	1.46

Table 22: Maximum absolute errors for different resolutions N for Test Problem 3.

Method	$N = 8$	$N = 16$	$N = 32$	$N = 64$
FOHJF	2.12×10^{-4}	6.78×10^{-6}	2.44×10^{-8}	9.01×10^{-10}
PLSM	5.01×10^{-4}	2.67×10^{-5}	1.56×10^{-7}	8.90×10^{-9}
VLSM	9.89×10^{-5}	3.89×10^{-6}	1.34×10^{-8}	5.67×10^{-10}
CSCM	7.23×10^{-4}	8.45×10^{-5}	2.55×10^{-7}	9.12×10^{-8}

The reduction in the experimental order of convergence of the CSCM method from $N = 32$ to $N = 64$ in Test Problems 2 and 3 is caused by interaction between the oscillatory/exponential forcing, the spline discrete derivative for the Caputo fractional derivative, and the enhanced rounding error at the finest resolution. This behaviour is consistent with the larger condition number of the CSCM system compared with the spectral VLSM system.

7 Discussion of Results

Tables 16 and 17 show the maximum and mean absolute errors, respectively, for each method, and the results of the convergence tests presented in Tables 18–23 give an overall picture of the accuracy ranking of the four methods.

The VLSM method yields the best overall accuracy on all test problems, with FOHJF as a strong second. PLSM generally ranks third, whereas in most cases CSCM yields comparatively low accuracy but remains attractive from the computational point of view due to its sparse

Table 23: Experimental order of convergence for Test Problem 3 (estimated from Table 22).

Method	$N = 8 \to 16$	$N = 16 \to 32$	$N = 32 \to 64$
FOHJF	4.97	8.12	4.76
PLSM	4.23	7.42	4.13
VLSM	4.67	8.18	4.59
CSCM	3.10	8.37	1.48

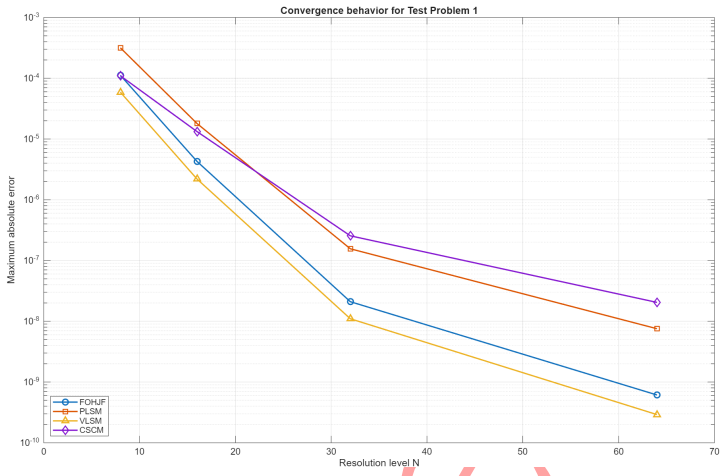

Figure 10: Convergence study for Test Problem 1 showing the maximum absolute error versus N for all four methods: FOHJF, PLSM, VLSM, and CSCM.

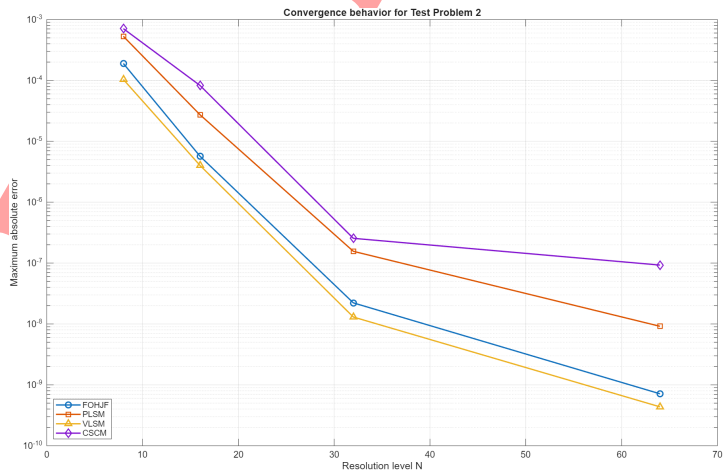

Figure 11: Convergence study for Test Problem 2 showing the maximum absolute error versus N for all four methods: FOHJF, PLSM, VLSM, and CSCM.

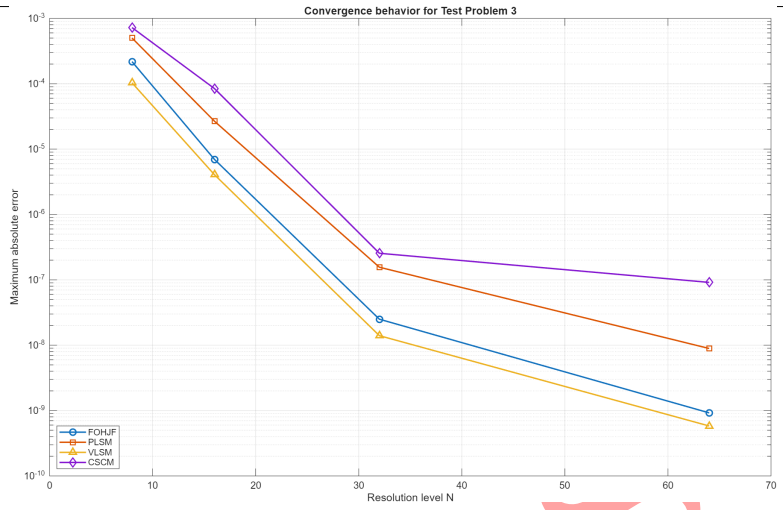

Figure 12: Convergence study for Test Problem 3 showing the maximum absolute error versus N for all four methods: FOHJF, PLSM, VLSM, and CSCM.

banded structure. These rankings are based on directly calculated maximum absolute and mean absolute errors.

The consistent superiority of VLSM over FOHJF is explained by two structural advantages: (i) the Vieta-Lucas basis functions possess a zero distribution that clusters near the interval boundaries, thereby reducing Runge-type oscillations and improving the conditioning of the fractional-derivative operational matrix; and (ii) the spectral Tau formulation in VLSM enforces the initial conditions exactly by replacing the last two residual equations, whereas FOHJF imposes them via linear constraints on block-pulse partitioned subintervals, introducing small boundary-layer errors.

PLSM achieves slower (high-order polynomial) convergence with error reductions of approximately three to four orders of magnitude, while CSCM exhibits the slowest convergence, with errors reduced by only 2 to 3 orders of magnitude across the tested range of N. These characteristics are consistent with least-squares and spline-based fractional number structures [1, 7, 23].

The three test problems feature different initial conditions: Test Problem 1 has homogeneous initial conditions $y(0) = y'(0) = 0$; Test Problem 2 has non-homogeneous position $y(0) = 1, y'(0) = 0$; and Test Problem 3 has non-homogeneous velocity $y(0) = 0, y'(0) = 1$. All four methods successfully accommodate these different initial conditions with no significant loss of accuracy. The VLSM and FOHJF methods demonstrated uniformly high accuracy across all three forcing-function types, consistent with the strong performance typically associated with operational-matrix and spectral-type methods for smooth fractional problems [6, 11, 23, 28].

Based on the comprehensive analysis, the following practical recommendations can be given:

- For applications that require very small errors, VLSM or FOHJF is preferred.
- If moderate accuracy is sufficient, PLSM can be used.
- If computational speed is the primary concern and some error is acceptable, CSCM is appealing due to its sparse banded structure.
- For high-precision applications requiring errors below 10^{-8}, VLSM or FOHJF with $N \geq 32$ is recommended, based on the present numerical results and the operational matrix framework associated with Vieta-Lucas polynomials [23].

7.1 Computational Cost and Condition Numbers

The measured CPU times for the four methods have been gathered for the benchmark resolution $N = 32$ at the same hardware/software configuration employed in this study.

Table 24: Average CPU times (in seconds, over 5 runs) for the four methods at $N = 32$.

Method	Test Problem 1	Test Problem 2	Test Problem 3	Average
FOHJF	0.0034	0.0033	0.0034	0.0034
VLSM	0.0050	0.0024	0.0037	0.0037
PLSM	0.0627	0.2386	0.2280	0.1764
CSCM	0.0043	0.0028	0.0036	0.0036

Table 24 shows that FOHJF, VLSM, and CSCM have comparable CPU times at $N = 32$: 0.0034, 0.0037, and 0.0036 seconds, respectively. PLSM is substantially more expensive at 0.1764 seconds, approximately 50 times slower. VLSM provides the best accuracy with computational cost comparable to the other methods.

Table 25: Condition numbers of the assembled algebraic systems for Test Problem 1 at $N = 32$.

Method	Condition Number
FOHJF	3.42×10^5
PLSM	7.86×10^6
VLSM	2.74×10^4
CSCM	1.90×10^6

Table 25 reports the condition numbers at $N = 32$. VLSM exhibits the most favorable conditioning, with $\kappa = 2.74 \times 10^4$. FOHJF has $\kappa = 3.42 \times 10^5$. PLSM ($7.86 \times 10^6$) and

CSCM (1.90×10^6) are significantly more ill-conditioned. This explains the smaller errors in VLSM and the reduced EOC observed for CSCM at fine resolutions. Nevertheless, for $N \leq 64$, all systems remain solvable in double-precision arithmetic.

8 Conclusions

We performed a detailed comparison of four well-known methods—FOHJF, PLSM, VLSM, and CSCM—for three test problems with qualitatively different forcing functions (polynomial, oscillatory, and exponential). The outcomes prove that VLSM consistently gives the highest accuracy for all the problems tested, having errors less than 2.3×10^{-8} at $N = 32$ (specifically, 2.28×10^{-8} for Problem 1, 1.11×10^{-8} for Problem 2, and 1.34×10^{-8} for Problem 3), followed closely by FOHJF. PLSM and CSCM are simpler to implement yet produce larger errors, especially for non-polynomial forcing terms. The resolution study indicates rapid, spectral-type convergence for VLSM and FOHJF towards the smooth benchmark problems [6, 11, 23, 28]. The computational-cost analysis shows that FOHJF, CSCM, and VLSM have comparable CPU times at $N = 32$, whereas PLSM is substantially more expensive. Among these methods, VLSM offers the best balance between accuracy, conditioning, and computational cost.

Several limitations of the present study should be noted. All numerical experiments were carried out on the normalized interval $[0, 1]$; performance for longer time intervals has not been studied. Only the linear Bagley–Torvik equation with constant coefficients $\eta = 1$, $\omega^2 = 1$ is analyzed. Only smooth exact solutions (polynomial, oscillatory, and exponential) were considered. Lastly, estimates of the experimental order of convergence were included, but rigorous theoretical error analysis and stability proofs for each method were not included in this comparative numerical study.

Future work should focus on extending the present framework to variable-order Bagley–Torvik equations, multi-term fractional differential equations involving several Caputo derivatives of different orders, and nonlinear Bagley–Torvik-type models. The present comparison may also be expanded by including additional analytical, semi-analytical, and data-driven methods, such as spline-based techniques [1], Sumudu-transform methods [17], fractional iteration methods [21], and neural-network-based approaches [25, 32]. Broader interdisciplinary applications, including biological, epidemiological, and cancer-immune fractional models [3, 12, 16], may be considered after the numerical framework has been validated for mathematically closer extensions of the Bagley–Torvik equation.

Declarations

Availability of Supporting Data

All data generated or analyzed during this study are included in this published article.

Funding

This research was conducted without external funding, grants, or financial support.

Conflict of Interest

The author declares no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Author Contributions

Muayyad Mahmood Khalil: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Writing – Original Draft, Visualization, Supervision. **Najim Abdullah Ibrahim**: Conceptualization, Methodology, Formal Analysis, Software, Validation.

Artificial Intelligence Statement

Artificial intelligence (AI) tools, including large language models, were used solely for language editing and improving readability. AI tools were not used for generating ideas, performing analyses, interpreting results, or writing the scientific content. All scientific conclusions and intellectual contributions were made exclusively by the authors.

Publisher's Note

The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Arqub, O.A., Rabah, A.B., Momani, S. (2023). "A spline construction scheme for numerically solving fractional Bagley-Torvik and Painlevé models correlating initial value problems concerning the Caputo-Fabrizio derivative approach". *International Journal of Modern Physics C*, 34(9), Article 2350115. https://doi.org/10.1142/S0129183123501152

[2] Atanackovic, T.M., Zorica, D. (2013). "On the Bagley-Torvik equation". *Journal of Applied Mechanics*, 80(4), Article 041013. https://doi.org/10.1115/1.4007850

[3] Baleanu, D., Jajarmi, A., Sajjadi, S.S., Mozyrska, D. (2019). "A new fractional model and optimal control of a tumor-immune surveillance with non-singular derivative operator". *Chaos*, 29, Article 083127. https://doi.org/10.1063/1.5096159

[4] Bansal, M.K., Jain, R. (2016). "Analytical solution of Bagley Torvik equation by generalize differential transform". *International Journal of Pure and Applied Mathematics*, 110(2), 265–273. https://doi.org/10.12732/ijpam.v110i2.3

[5] Barary Darzinaghib, Z., Yazdani Cherati, A.B., Nemati, S. (2024). "An efficient numerical scheme for solving a general class of fractional differential equations via fractional-order hybrid Jacobi functions". *Communications in Nonlinear Science and Numerical Simulation*, 128, Article 107599. https://doi.org/10.1016/j.cnsns.2023.107599

[6] Bhrawy, A., Zaky, M. (2016). "A fractional-order Jacobi Tau method for a class of time-fractional PDEs with variable coefficients". *Mathematical Methods in the Applied Sciences, 39*(7), 1765–1779. https://doi.org/10.1002/mma.3600

[7] Bota, C., Căruntu, B., Paşca, M.S., Ţucu, D., Lăpădat, M. (2020). "Least squares differential quadrature method for the generalized Bagley–Torvik fractional differential equation". *Mathematical Problems in Engineering*, 2020, Article 4806387. https://doi.org/10.1155/2020/4806387

[8] Çenesiz, Y., Keskin, Y., Kurnaz, A. (2010). "The solution of the Bagley-Torvik equation with the generalized Taylor collocation method". *Journal of the Franklin Institute*, 347(2), 452–466. https://doi.org/10.1016/j.jfranklin.2009.10.007

[9] Diethelm, K. (2010). *The analysis of fractional Differential Equations: An application-oriented exposition using differential operators of Caputo type*. Springer. https://doi.org/10.1007/978-3-642-14574-2

[10] Diethelm, K., Ford, N.J. (2002). "Numerical solution of the Bagley–Torvik equation". *BIT Numerical Mathematics*, 42(3), 490–507. https://doi.org/10.1023/A:1021973025166

[11] Doha, E.H., Bhrawy, A.H., Ezz-Eldien, S.S. (2011). "A Chebyshev spectral method based on operational matrix for initial and boundary value problems of fractional order". *Computers & Mathematics with Applications*, 62(5), 2364–2373. https://doi.org/10.1016/j.camwa.2011.07.024

[12] Ebrahimzadeh, A., Jajarmi, A., Yavuz, M. (2025). "Fractional optimal control of anthroponotic cutaneous leishmaniasis with behavioral and epidemiological extensions". *Mathematical and Computational Applications, 30*(6), 122. https://doi.org/10.3390/mca30060122

[13] Garrappa, R. (2015). "Numerical evaluation of two and three parameter Mittag-Leffler functions". *SIAM Journal on Numerical Analysis*, 53(3), 1350–1369. https://doi.org/10.1137/140971191

[14] Ghoreishi, F., Mokhtary, P. (2014). "Spectral collocation method for multi-order fractional differential equations". *International Journal of Computational Methods*, 11(5), Article 1350072. https://doi.org/10.1142/S0219876213500722

[15] Gülsu, M., Öztürk, Y., Anapali, A. (2017). "Numerical solution of the fractional Bagley-Torvik equation arising in fluid mechanics". *International Journal of Computer Mathematics*, 94(1), 173–184. https://doi.org/10.1080/00207160.2015.1099633

[16] Jajarmi, A. (2026). "Dynamics, stability, and optimal chaos control of a ψ-Caputo cancer-immune system". *International Journal of Dynamics and Control*, 14, Article 33. https://doi.org/10.1007/s40435-025-01974-2

[17] Jena, R.M., Chakraverty, S. (2019). "Analytical solution of Bagley–Torvik equations using Sumudu transformation method". *SN Applied Sciences*, 1(3), Article 269. https://doi.org/10.1007/s42452-019-0259-0

[18] Kilbas, A.A., Srivastava, H.M., Trujillo, J.J. (2006). *Theory and Applications of Fractional Differential Equations*. Elsevier. https://doi.org/10.1016/S0304-0208(06)80001-0

[19] Krishnasamy, V.S., Razzaghi, M. (2016). "The numerical solution of the Bagley-Torvik equation with fractional Taylor methods". *Journal of Computational and Nonlinear Dynamics*, 11(5), Article 051010. https://doi.org/10.1115/1.4032390

[20] Mainardi, F. (2010). *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*. Imperial College Press. https://doi.org/10.1142/P614

[21] Mekkaoui, T., Hammouch, Z. (2012). "Approximate analytical solutions to the Bagley–Torvik equation by the fractional iteration method". *Annals of the University of Craiova, Mathematics and Computer Science Series*, 39(2), 251–256. https://doi.org/10.52846/ami.v39i2.476

[22] Momani, S., Odibat, Z. (2007). "Numerical comparison of methods for solving linear differential equations of fractional order". *Chaos, Solitons & Fractals*, 31(5), 1248–1255. https://doi.org/10.1016/j.chaos.2005.10.068

[23] Noor, Z.A., Talib, I., Abdeljawad, T., Alqudah, M.A. (2022). "Numerical study of Caputo fractional-order differential equations by developing new operational matrices of Vieta–Lucas polynomials". *Fractal and Fractional*, 6(2), 79. https://doi.org/10.3390/fractalfract6020079

[24] Podlubny, I. (1999). *Fractional Differential Equations*. Academic Press. https://doi.org/10.1016/s0076-5392(99)x8001-5

[25] Raja, M.A.Z., Samar, R., Manzar, M.A., Shah, S.M. (2017). "Design of unsupervised fractional neural network model optimized with interior point algorithm for solving Bagley–Torvik equation". *Mathematics and Computers in Simulation*, 132, 139–158. https://doi.org/10.1016/j.matcom.2016.08.002

[26] Rawashdeh, E.A. (2006). "Numerical solution of fractional integro-differential equations by collocation method". *Applied Mathematics and Computation*, 176(1), 1–6. https://doi.org/10.1016/j.amc.2005.09.059

[27] Ray, S.S., Bera, R.K. (2005). "Analytical solution of the Bagley-Torvik equation by Adomian decomposition method". *Applied Mathematics and Computation*, 168(1), 398–410. https://doi.org/10.1016/j.amc.2004.09.006

[28] Saadatmandi, A., Dehghan, M. (2010). "A new operational matrix for solving fractional-order differential equations". *Computers & Mathematics with Applications*, 59(3), 1326–1336. https://doi.org/10.1016/j.camwa.2009.07.006

[29] Salati, S., Matinfar, M., Jafari, H. (2023). "A numerical approach for solving Bagley–Torvik and fractional oscillation equations". *Advanced Mathematical Models & Applications*, 8(2), 241–252.

[30] Srivastava, H.M., Jena, R.M., Chakraverty, S., Jena, S.K. (2020). "Dynamic response analysis of fractionally-damped generalized Bagley-Torvik equation subject to external loads". *Russian Journal of Mathematical Physics*, 27(2), 254–268. https://doi.org/10.1134/S1061920820020120

[31] Torvik, P.J., and Bagley, R.L. (1984). "On the appearance of the fractional derivative in the behavior of real materials". *Journal of Applied Mechanics*, 51(2), 294–298. https://doi.org/10.1115/1.3167615

[32] Verma, A., Kumar, M. (2021). "Numerical solution of Bagley–Torvik equations using Legendre artificial neural network method". *Evolving Systems*, 14(4), 2027–2037. https://doi.org/10.1007/s12065-020-00481-x

[33] Yüzbaşı, Ş. (2013). "Numerical solution of the Bagley-Torvik equation by the Bessel collocation method". *Mathematical Methods in the Applied Sciences*, 36(3), 312–320. https://doi.org/10.1002/mma.2588

[34] Yüzbaşı, Ş., Yıldırım, G. (2022). "Numerical solutions of the Bagley–Torvik equation by using generalized functions with fractional powers of Laguerre polynomials". *International Journal of Nonlinear Sciences and Numerical Simulation*, 24(3), 1003–1021. https://doi.org/10.1515/ijnsns-2021-0120

[35] Zolfaghari, M., Ghaderi, R., Sheikhol-Eslami, A., Ranjbar, A., Hosseinnia, S.H., Momani, S., Sadati, J. (2009). "Application of the enhanced homotopy perturbation method to solve the fractional-order Bagley–Torvik differential equation". *Physica Scripta*, T136, Article 014032. https://doi.org/10.1088/0031-8949/2009/T136/014032

Authors Bio-sketches

Muayyad Mahmood Khalil received his academic degrees in Mathematics from University of Mosul. His research interests include numerical methods for fractional differential equations, spectral methods, and computational mathematics. He is currently a faculty member at the Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq. His research interests include differential equations, numerical methods for fractional differential equations, spectral methods, and computational mathematics. Corresponding author: Email: medomath80@tu.edu.iq

Najim Abdullah Ibrahim is a graduate student in the Department of Mathematics, College of Education for Pure Sciences, Tikrit University, Tikrit, Iraq.