Math. & Comput. Sci., 7(3) (2026), 142–159

Online: ISSN 2717-2708

Mathematics and Computational Sciences

Journal Homepage: http://mcs.qut.ac.ir/mcs

A comparative analysis of Daftardar-Gejji and Jafari method and Chebyshev least squares method in solving Volterra integro-differential equations

Maria Rahman[a], Dipta Roy[a], Faria Tasnim[a], Tania S. Khaleque[a,*]

[a]Department of Applied Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

Abstract

This study presents a comparative analysis of two efficient numerical approaches for solving linear and nonlinear Volterra integro-differential equations: the Daftardar-Gejji and Jafari Method (DJM) and the Chebyshev Least Squares Method (CLSM). The DaftardarGejji and Jafari Method (DJM) provides robust results without linearization or perturbation assumptions, whereas CLSM employs shifted Chebyshev polynomial approximations together with least-squares minimization to obtain highly accurate spectral solutions. Convergence and stability properties of both methods are briefly discussed within fixed-point and weighted residual frameworks. Several benchmark first- through fourth-order VIDEs are solved to evaluate the performance of the proposed approaches. The obtained numerical results show excellent agreement with exact solutions and previously published methods. Comparative analyses based on absolute error, L_∞-norm, residual behavior, and CPU time are presented to examine the convergence, computational efficiency, and numerical stability of the proposed methods.

Keywords: Volterra Integro-Differential equation, Daftardar-Gejji and Jafari method, Chebyshev Least Squares method, Shifted Chebyshev polynomial, Computational stability analysis

2020 MSC: 45J05, 45D05, 65R20

1. Introduction

The theory and application of integro-differential equations originated with the work of Volterra, who not only pioneered research in this area but also coined the term for these functional equations in his earliest publication on the subject in 1910 [43]. IDEs serve as models in various scientific and engineering disciplines, including population dynamics, one-dimensional viscoelasticity, reactor kinetics, heat conduction with memory effects, signal processing in control systems, and fluidstructure interaction problems [25, 34, 26, 29]. These equations are generally classified into three main types: Fredholm, Volterra, and Volterra-Fredholm.

Many partial differential equation (PDE) models involving memory effects, hereditary behavior, and nonlocal interactions can be reformulated as Volterra integro-differential equations through temporal convolution kernels of the form

*Corresponding author
Email addresses: mariarahman2025d@gmail.com (Maria Rahman), diptaroy2025@gmail.com (Dipta Roy), tasnim.nahin@gmail.com (Faria Tasnim), tania.khaleque@du.ac.bd (Tania S. Khaleque)

doi: 10.30511/mcs.2026.2073962.1491

Received: 09 October 2025 Accepted: 17 June 2026

$$y'(x) = f(x, y(x)) + \int_0^x K(x, t, y(t))\, dt.$$

In particular, time-fractional differential equations are also closely related to Volterra integral operators since fractional derivatives are commonly represented through weakly singular convolution kernels. Such formulations naturally lead to VIDE-type structures in which the present state depends on accumulated temporal effects [31, 33]. Recent studies involving nonlocal diffusion models, memory-dependent PDE systems, and fractional dynamical formulations have further highlighted the importance of robust numerical approaches for equations containing convolution-type kernels and hereditary memory effects [42].

Over the past thirty years, there has been growing and sustained interest in the quantitative and qualitative analysis of Volterra integro-differential equations (VIDEs) involving both bounded and unbounded delays [16]. The standard form of the Volterra integro-differential equation is

$$y^{(n)}(x) = f\left(x, y(x), y'(x), \ldots, y^{(n-1)}(x)\right) + \int_0^x K\left(x, t, y(t), y'(t), \ldots, y^{(n-m)}(t)\right) dt, \quad x > 0 \tag{1.1}$$

with initial conditions $y(0) = y_0, y'(0) = y_1, y^{(n-1)}(0) = y_{n-1}$, where the kernel function K depends not only on the past state, but also on the present state.

Research on the solutions of integral and IDEs has been ongoing since the pioneering works of Volterra [44, 1, 21]. While many studies address linear forms of these equations, obtaining exact analytical solutions is often not feasible. As a result, numerical methods have become increasingly prominent in the literature [13]. To address this challenge, a variety of numerical algorithms have been developed and applied to approximate solutions effectively [5]. These include the Wavelet-Galerkin method [15], monotone iterative techniques [9, 47], the homotopy perturbation method combined with the reproducing kernel approach [7], the Adomian decomposition method [20], finite difference method [11], Picard-Greens method [6, 40], the Tau method [22], spectral collocation methods [24], Taylor polynomial approximations [30], Lagrange interpolation [36], the exponential spline method [39], and many more. The monotone iterative technique, incorporating the RiemannLiouville fractional derivative, was proposed to address fractional integro-differential equations (FIDEs) with advanced arguments [28]. The collocation method based on Bessel polynomials was employed to solve linear Fredholm integro-differential-difference equations [37, 12].

Among iterative analytical-numerical techniques, one of the earliest techniques proposed for integral equations is the Jeffery method, introduced by G.B. Jeffery in the early 20th century [23]. This method transforms the integral part into a recursive computational scheme and is appreciated for its conceptual simplicity and ease of implementation. It has been particularly useful for solving linear Volterra integral equations and some classes of nonlinear problems. However, the method tends to lose accuracy and stability for equations with strong nonlinearity, rapidly changing kernels, or high-order differential terms. A more recent technique developed for handling nonlinear functional equations is the Daftardar–Gejji and Jafari Method (DJM), introduced by V. Daftardar–Gejji and H. Jafari in 2006 [18]. Since its introduction, the method has attracted considerable attention for solving nonlinear functional equations, integral equations, and integro-differential systems. The method constructs a recursive decomposition scheme without requiring linearization, perturbation parameters, or Adomian polynomials, making it computationally attractive for strongly nonlinear problems. Its recursive formulation enables rapid convergence under mild smoothness assumptions. In recent years, DJM and its modified variants have been successfully extended to nonlinear fractional and integro-differential models, demonstrating accurate and stable numerical approximations for complex nonlinear systems [35, 4].

On the other hand, spectral approximation techniques based on orthogonal polynomials have emerged as highly accurate tools for solving differential and integro-differential equations. Among these methods, Chebyshev polynomials of the first kind have gained considerable attention due to their orthogonality, computational efficiency, and excellent approximation properties. These polynomialsparticularly in their

shifted formprovide stable and rapidly converging approximations for functions defined over finite intervals. The application of Chebyshev polynomial basis functions in solving Volterra integro-differential equations (VIDEs) has gained considerable interest in recent years due to their ability to produce accurate numerical approximations with reduced computational effort. Adebisi et al. explored this potential using the Galerkin method, where the approximate solution is expanded in terms of shifted Chebyshev polynomials and substituted into the given VIDE [2]. The resulting residual was minimized by taking its inner product with the same basis, forming a solvable system of algebraic equations. Recent studies, such as the work of Sunthrayuth et al., have further demonstrated the effectiveness of Chebyshev-based spectral techniques for solving Volterra integro-differential equations, yielding accurate and stable numerical approximations with rapid convergence properties [38]. Adebisi and Okunola extended Chebyshev-based least squares methods to VolterraFredholm integro-differential equations, showing robustness for linear and nonlinear systems [3]. Recent advances in Chebyshev-based spectral techniques and spectral collocation methods have further demonstrated strong approximation capabilities for deterministic and stochastic integro-differential equations with favorable convergence properties [32, 45].

Despite the large number of available numerical techniques, comparatively limited attention has been devoted to a systematic investigation of recursive decomposition-based approaches such as the Daftardar-Gejji and Jafari Method (DJM) alongside spectral least-squares frameworks based on shifted Chebyshev polynomials, particularly for higher-order nonlinear Volterra integro-differential equations (VIDEs). Moreover, relatively few studies have examined the convergence behavior, residual reduction, approximation accuracy, and computational performance of these methods within a unified numerical framework.

Motivated by these observations, the present study investigates the effectiveness of the Daftardar-Gejji and Jafari Method (DJM) and the Chebyshev Least Squares Method (CLSM) for solving first- through fourth-order linear and nonlinear Volterra integro-differential equations. The proposed approaches are applied to several benchmark problems, and the obtained numerical solutions are compared with exact solutions as well as previously published numerical results available in the literature. In addition to solution accuracy, the study examines residual behavior, L_∞-error analysis, convergence characteristics, and numerical behavior under different problem settings. The overall objective is to establish an effective computational framework for evaluating recursive and spectral numerical methods for higher-order memory-dependent integro-differential systems.

2. Methodology

2.1. The Daftardar–Gejji and Jafari Method (DJM)

The Daftardar–Gejji and Jafari Method (DJM) is a highly effective and flexible iterative approach for solving a wide range of equations, particularly nonlinear Volterra integro-differential equations (VIDEs). One of its primary strengths lies in its simplicity and direct applicationit avoids the complications involved in traditional analytical methods. DJM constructs a sequence of approximations that reliably converges to the exact solution under broad conditions [8]. The core idea behind DJM is to rewrite the target equation into a canonical operator form and then apply an iterative solution process.

Let $X = \mathcal{C}([0, T])$ denote the Banach space of continuous functions equipped with the supremum norm

$$\|y\|_\infty = \max_{x\in[0,T]} |y(x)|.$$

A general nonlinear functional equation can be represented in the operator form

$$y(x) = G(x) + \mathcal{L}(y)(x) + \mathcal{N}(y)(x), \tag{2.1}$$

where y denotes the unknown solution function, G is a known source term, $\mathcal{L}$ represents a bounded linear operator, and $\mathcal{N}$ denotes the nonlinear operator associated with the integral and differential terms.

DJM assumes that the solution y can be expressed as an infinite series

$$y = \sum_{m=0}^{\infty} y_m, \tag{2.2}$$

where the components y_m are recursively determined through the iterative scheme. The individual components y_m of the series are determined by the following recurrence relation, where the initial approximation is selected from the known function $G(x)$.

$$y_0(x) = G(x).$$

$$y_{m+1} = \mathcal{L}(y_m) + \mathcal{N}\left(\sum_{j=0}^{m} y_j\right) - \mathcal{N}\left(\sum_{j=0}^{m-1} y_j\right), \quad m \geqslant 0. \tag{2.3}$$

An approximate solution can be obtained by summing the first $J+1$ terms as

$$y \approx \sum_{j=0}^{J} y_j, \tag{2.4}$$

where J is chosen large enough to achieve the desired accuracy.
Consider a general first-order nonlinear Volterra integro-differential equation of the form

$$y'(x) = F(x, y(x)) + \int_0^x K(x, t, y(t))\, dt, \tag{2.5}$$

where $F(x, y(x))$ represents the differential or source component of the equation, while $K(x, t, y(t))$ denotes the Volterra kernel associated with the memory-dependent integral term, with the initial condition $y(0) = y_0$. Such equations describe processes where the current rate of change is influenced not only by the present state but also by an accumulated memory term [14].
By integrating both sides from 0 to x, we obtain the equivalent integral equation

$$y(x) = y_0 + \int_0^t F(t, y(t))\, dt + \int_0^x \left(\int_0^t K(t, \tau, y(\tau))\, d\tau\right) dt. \tag{2.6}$$

To solve this numerically using DJM, we discretize the time domain. Let the mesh be $0 = x_0 < x_1 < \cdots < x_M = T$ with a constant step size $h = x_{j+1} - x_j$. Applying the trapezoidal rule for both inner and outer integrals, the discretized form at x_{j+1} becomes

$$y(x_{j+1}) = y(x_j) + \frac{h}{2}[F(x_j, u(x_j)) + F(x_{j+1}, y(x_{j+1}))] + \text{Discretized Integral Terms}. \tag{2.7}$$

The DJM constructs recursive approximations without linearization, perturbation assumptions, or Adomian polynomials. In the present work, trapezoidal quadrature is additionally employed for numerical evaluation of the integral terms arising in the VIDE formulation.
The nested integral term involving K is approximated using successive applications of the trapezoidal rule. As a result, the discrete formula for y_{j+1} depends on known values at previous steps and includes terms like $F(x_j, y_j)$, $K(x_j, \cdot, y(\cdot))$, and $F(x_{j+1}, y_{j+1})$, $K(x_{j+1}, \cdot, y(\cdot))$. Within this framework, DJM is used to iteratively approximate y_{j+1}. The discretized equation is expressed as

$$y_{j+1} = \Phi(y_{j+1}, y_j, y_{j-1}, ..., y_0), \tag{2.8}$$

where Φ represents the nonlinear discrete operator obtained after applying trapezoidal discretization to the transformed integral equation. The DJM generates an iterative sequence

$$y_{j+1}^{(0)} = \text{Known values from } x_j \text{ and earlier steps}$$

$$y_{j+1}^{(k+1)} = \text{Linear part of } \Phi(y_{j+1}^{(k)}) + \text{Nonlinear part of } \Phi\left(\sum_{p=0}^{k} y_{j+1}^{(p)}\right) - \text{Nonlinear part of } \Phi\left(\sum_{p=0}^{k-1} y_{j+1}^{(p)}\right)$$

In practical computation, a finite number of terms in this DJM iteration suffices. For example, a three-term approximation may be expressed as

$$y_{j+1} \approx P_1 + \frac{h}{2}F(x_{j+1}, P_2) + \frac{h^2}{4}K(x_{j+1}, x_{j+1}, P_2), \tag{2.9}$$

where P_1 contains all previously computed quantities, while P_2 denotes the recursive approximation associated with the implicit nonlinear term at t_{j+1}.

$$\begin{aligned} P_1 = y_j + \frac{h}{2}F(x_j, y_j) + \frac{h^2}{4}\left[K(x_j, x_0, y_0) + K(x_j, x_j, y_j) + K(x_{j+1}, x_0, y_0)\right] \\ + \frac{h^2}{2}\sum_{i=1}^{j-1} K(x_j, x_i, y_i) + \frac{h^2}{2}\sum_{i=1}^{j} K(x_{j+1}, x_i, y_i). \end{aligned} \tag{2.10}$$

$$P_2 = P_1 + \frac{h}{2}F(x_{j+1}, P_2) + \frac{h^2}{4}K(x_{j+1}, x_{j+1}, P_2). \tag{2.11}$$

This implicit equation for P_2 is often solved using fixed-point iteration or further DJM steps, showcasing the methods self-recursive nature. The domain is discretized into M points, and the value y_{j+1} is updated sequentially for each $j = 0, 1, \ldots, M-1$.

Equations (2.5)–(2.9) may be interpreted as a fixed-point iterative scheme of the form

$$y = \mathcal{T}(y),$$

where the nonlinear operator $\mathcal{T}$ is defined through the recursive DJM formulation. The successive approximations generated by DJM converge toward the fixed point of $\mathcal{T}$ under suitable contraction conditions.

Assume that the operators involved in the recursive formulation satisfy the Lipschitz conditions

$$\|\mathcal{L}(y) - \mathcal{L}(z)\| \leqslant \alpha\|y - z\|, \quad \text{and} \quad \|\mathcal{N}(y) - \mathcal{N}(z)\| \leqslant \beta\|y - z\|,$$

for some constants $\alpha, \beta \geqslant 0$ satisfying $\alpha + \beta < 1$. Then the operator $\mathcal{T}$ becomes contractive on the Banach space $X = \mathcal{C}([0, T])$. Hence, by Banachs Fixed-Point Theorem, the DJM iterative sequence converges to the unique solution of the Volterra integro-differential equation [27, 46]. Furthermore, under the imposed Lipschitz assumptions, the DJM iterative process exhibits linear convergence behavior[27].

DJM can also be extended to solve higher-order VIDEs by reducing them to systems of first-order equations. This is accomplished by defining new variables for each derivative of the unknown function.

Consider a second-order equation

$$\frac{d^2y(x)}{dx^2} = F\left(x, y(x), \frac{dy(x)}{dx}, \int_0^x K(x, t, y(t))dt\right). \tag{2.12}$$

We introduce

$$y_1(x) = y(x), \quad y_2(x) = \frac{dy(x)}{dx}.$$

This converts the equation into the following first-order system

$$\frac{dy_1(x)}{dx} = y_2(x), \quad \frac{dy_2(x)}{dx} = F\left(x, y_1(x), y_2(x), \int_0^x K(x, t, y_1(t))dt\right). \tag{2.13}$$

The DJM is then applied to each component of the system separately. The iterative approximations for $y_1(x)$ and $y_2(x)$ are updated concurrently using the same recursive scheme employed for the scalar case. This ensures consistency and accuracy across all variables in the system.

The stability of DJM is closely related to the contraction property of the corresponding nonlinear operator. Under the imposed Lipschitz conditions, perturbations introduced during successive iterations do not amplify uncontrollably, thereby ensuring stable numerical behavior throughout the computational interval [27].

The convergence behavior of DJM is associated with the reduction of the iterative residual. Let

$$R^{(m)}(x) = y^{(m)}(x) - G(x) - \mathcal{L}(y^{(m)})(x) - \mathcal{N}(y^{(m)})(x)$$

denote the residual corresponding to the m-th approximation. As the number of iterations increases, the residual norm decreases progressively, indicating convergence toward the exact solution. The convergence properties of DJM are strongly influenced by the regularity of the kernel function $K(x,t)$. If the kernel is sufficiently smooth and satisfies the Lipschitz continuity condition with respect to the solution variable, the associated nonlinear operator remains contractive, leading to stable and rapidly convergent iterative approximations. In contrast, if the kernel possesses weak singularities or non-smooth derivatives, the contraction property may weaken locally, which can slow the convergence rate of the iterative sequence. In such cases, additional discretization errors may arise from the numerical evaluation of the memory integral term. Nevertheless, under suitable boundedness assumptions on the kernel function, the DJM iteration remains stable and convergent for a broad class of nonlinear Volterra integro-differential equations [27].

The DJM is well-reputed for its fast convergence, even under minimal smoothness assumptions. Its primary advantage lies in its ability to approximate nonlinear systems without linearization or repetitive discretization of nonlinear terms. This makes the method computationally efficient. Moreover, the accurate handling of memory terms via numerical integration (e.g., trapezoidal rule) ensures precise modeling of hereditary effects, which is critical in integro-differential systems. These strengths make DJM a robust and reliable technique for solving complex nonlinear VIDEs.

2.2. Chebyshev Least Squares Method (CLSM)

The Chebyshev Least Squares Method (CLSM) is a powerful tool for solving Volterra integro-differential equations (VIDEs), utilizing the approximation strength of Chebyshev polynomials. These polynomials, especially those of the first kind, are extensively used in numerical analysis due to their minimization of interpolation errors and excellent approximation properties [2]. There are two primary families of Chebyshev polynomials: the first kind $T_n(x)$ and the second kind $U_n(x)$, both orthogonal over the interval $[-1,1]$ with respect to different weight functions.

The first kind, $T_k(x)$, can be defined using the cosine formulation on the interval $[-1,1]$

$$T_k(x) = \cos(k\cos^{-1}x), \quad x \in [-1,1],$$

which is equivalent to

$$T_k(x) = \cos(k\theta), \quad \text{with } \theta = \cos^{-1}(x).$$

These polynomials satisfy the recurrence relation

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad k \geqslant 1.$$

They are orthogonal with respect to the weight

$$w(x) = \frac{1}{\sqrt{1-x^2}}, \quad x \in [-1,1].$$

The initial few Chebyshev polynomials of the first kind are

$$T_0(x) = 1, T_1(x) = x, T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x,$$
$$T_4(x) = 8x^4 - 8x^2 + 1, T_5(x) = 16x^5 - 20x^3 + 5x.$$

To use these polynomials over a domain different from $[-1,1]$, a shifting transformation is applied to define the shifted Chebyshev polynomials $T_k^*(x)$ over an arbitrary interval $[r,s]$

$$T_k^*(x) = \cos\left(k\cos^{-1}\left(\frac{2x-s-r}{s-r}\right)\right), \quad x \in [r,s].$$

Since $\cos^{-1}(x)$ is well-defined for $x \in [-1,1]$, the above representation is valid throughout the computational domain. This trigonometric form provides several useful approximation properties and leads to numerically stable spectral approximations.

These shifted polynomials also satisfy a recurrence relation

$$T_0^*(x) = 1, \quad T_1^*(x) = \frac{2x-s-r}{s-r},$$

$$T_{k+1}^*(x) = 2\left(\frac{2x-s-r}{s-r}\right)T_k^*(x) - T_{k-1}^*(x), \quad k \geqslant 1.$$

For the interval $[0,1]$, the first several shifted Chebyshev polynomials become

$$\begin{aligned}
T_0^*(x) &= 1,\\
T_1^*(x) &= 2x-1,\\
T_2^*(x) &= 8x^2-8x+1,\\
T_3^*(x) &= 32x^3-48x^2+18x-1,\\
T_4^*(x) &= 128x^4-256x^3+160x^2-32x+1,\\
T_5^*(x) &= 512x^5-1280x^4+1120x^3-400x^2+50x-1,\\
T_6^*(x) &= 2048x^6-6144x^5+6912x^4-3840x^3+840x^2-72x+1,\\
T_7^*(x) &= 8192x^7-28672x^6+39424x^5-26990x^4+9408x^3-1568x^2+98x-1,
\end{aligned}$$

and so on.

The shifted Chebyshev polynomials satisfy an important orthogonality property with respect to the weighted inner product

$$\langle T_i, T_j\rangle = \int_{-1}^{1} \frac{T_i(x)T_j(x)}{\sqrt{1-x^2}}\,dx. \tag{2.14}$$

Using this weighted inner product, the orthogonality condition is given by

$$\langle T_i, T_j\rangle = \begin{cases} 0, & i \neq j,\\ \pi, & i = j = 0,\\ \dfrac{\pi}{2}, & i = j \neq 0. \end{cases} \tag{2.15}$$

The orthogonality of the Chebyshev basis functions plays a fundamental role in reducing the residual error and improving numerical stability in the least-squares approximation process.

To apply the CLSM, consider a general nth-order Volterra integro-differential equation

$$y^{(n)}(x) = g(x) + \lambda\int_{x_0}^{x} K(x,t)\,y(t)\,dt, \quad x \in [x_0, x_f] \tag{2.16}$$

with associated initial conditions

$$y^{(i)}(x_0) = \phi_i, \quad i = 0,1,\ldots,n-1. \tag{2.17}$$

An approximate solution is constructed using a finite expansion in shifted Chebyshev polynomials

$$y_N(x) = \sum_{k=0}^{N} a_k T_k^*(x) \tag{2.18}$$

where the polynomials $T_k^*(x)$ are defined over $[x_0, x_f]$, and a_k are the unknown coefficients to be computed. The variable change

$$s = \frac{2(x - x_0)}{x_f - x_0} - 1$$

maps the working interval to $[-1, 1]$ to facilitate use of standard formulas.

The nth derivative of the trial solution is

$$y_N^{(n)}(x) = \frac{d^n}{dx^n} \left(\sum_{k=0}^{N} a_k T_k^*(x) \right).$$

Substituting $y_N(x)$ into the original integro-differential equation results in the residual function

$$R(x) = \sum_{k=0}^{N} a_k \frac{d^n}{dx^n} T_k^*(x) - g(x) - \lambda \int_{x_0}^{x} K(x,t) \left(\sum_{k=0}^{N} a_k T_k^*(t) \right) dt. \tag{2.19}$$

In the Chebyshev Least Squares Method (CLSM), the unknown coefficients are determined by minimizing the weighted L^2 norm of the residual function. Specifically, the least-squares approximation is obtained by enforcing

$$\frac{\partial}{\partial a_k} \|R(x)\|_2^2 = 0, \quad k = 0, 1, \ldots, N,$$

where

$$\|R(x)\|_2^2 = \int_{-1}^{1} \frac{R^2(x)}{\sqrt{1 - x^2}} \, dx.$$

This minimization procedure generates a system of algebraic equations for the unknown Chebyshev coefficients.

To determine the best-fit coefficients a_k, the least squares technique is employed. The error is measured by the square of the residual over the domain

$$E(a_0, a_1, \ldots, a_N) = \int_{x_0}^{x_f} [R(x)]^2 \, dx \tag{2.20}$$

and minimized by satisfying the conditions:

$$\frac{\partial E}{\partial a_i} = 0, \quad i = 0, 1, \ldots, N \tag{2.21}$$

Solving this nonlinear algebraic system yields the coefficients a_k. When initial conditions are imposed, they are substituted directly into the approximate solution prior to minimizing the error functional, reducing the number of free parameters and ensuring the constraints are met [3].

Theorem 2.1. *(Convergence of CLSM). Assume that the exact solution $y(x)$ belongs to the Sobolev space $H^m([-1, 1])$, and suppose that the kernel function $k(x, t)$ is sufficiently smooth over the computational domain. Then the Chebyshev least-squares approximation $y_N(x)$ satisfies the error estimate*

$$\|y - y_N\| \leqslant C N^{-m}, \tag{2.22}$$

where C *is a positive constant independent of* N, *and* m *represents the regularity of the exact solution. Furthermore, if the solution* $y(x)$ *is analytic in the domain, then the approximation exhibits spectral convergence of the form*

$$\|y - y_N\| \leqslant Ce^{-\alpha N}, \tag{2.23}$$

where $\alpha > 0$ *is a constant depending on the analyticity region of the solution.*

Proof. The shifted Chebyshev polynomials form a complete orthogonal basis in the weighted function space over the computational interval. Since the least-squares approximation corresponds to the orthogonal projection of the exact solution onto the finite-dimensional polynomial subspace, standard spectral approximation theory implies that the projection error decreases algebraically for sufficiently smooth functions and exponentially for analytic functions. Hence the above estimates follow directly from classical Chebyshev approximation results.

□

The stability of CLSM is closely related to the orthogonality of the Chebyshev basis functions and the minimization of the weighted residual norm. Since the approximation coefficients are determined through a least-squares projection, the method suppresses the propagation of numerical errors and improves the conditioning of the resulting algebraic system.

Based on Theorem 2.1, the convergence behavior of CLSM depends fundamentally on the regularity of both the exact solution and the kernel function. For analytic kernels and sufficiently smooth solutions, the Chebyshev approximation exhibits spectral convergence, characterized by an exponential decay of the approximation error as the polynomial degree N increases. If the kernel function contains weak singularities, discontinuous derivatives, or limited smoothness, the regularity of the solution space is reduced, and the convergence rate may deteriorate from exponential to algebraic order. Nevertheless, due to the global orthogonality of the Chebyshev basis functions, CLSM generally maintains superior approximation accuracy and numerical stability compared with low-order local approximation methods [17, 41, 19].

This method effectively captures the behavior of the unknown function and its integral term, making CLSM a suitable symbolic-numerical technique for solving both linear and nonlinear VIDEs with high accuracy and efficiency.

3. Test Problems

In this section, the effectiveness of the DJM and CLSM methods is demonstrated through the following test examples. In tables and figures, we try to compare the exact solution and the approximate solutions obtained by the DJM and CLSM method. We also analyze the accuracy of this method by comparing the approximate solutions and absolute errors with different methods that are used to solve the same problems in different reference papers. All numerical computations presented in this work were carried out in MATLAB using standard double-precision floating-point arithmetic. Errors near 10^{-10} to 10^{-16} are close to machine precision and arise from smooth benchmark problems rather than exact cancellation.

Example 3.1. Here we consider a first-order nonlinear Volterra integro-differential equation,

$$y'(x) + \frac{y^2(x)}{2} + y(x) - \frac{1}{2} + \int_0^x y^2(t)\,dt = 0,$$

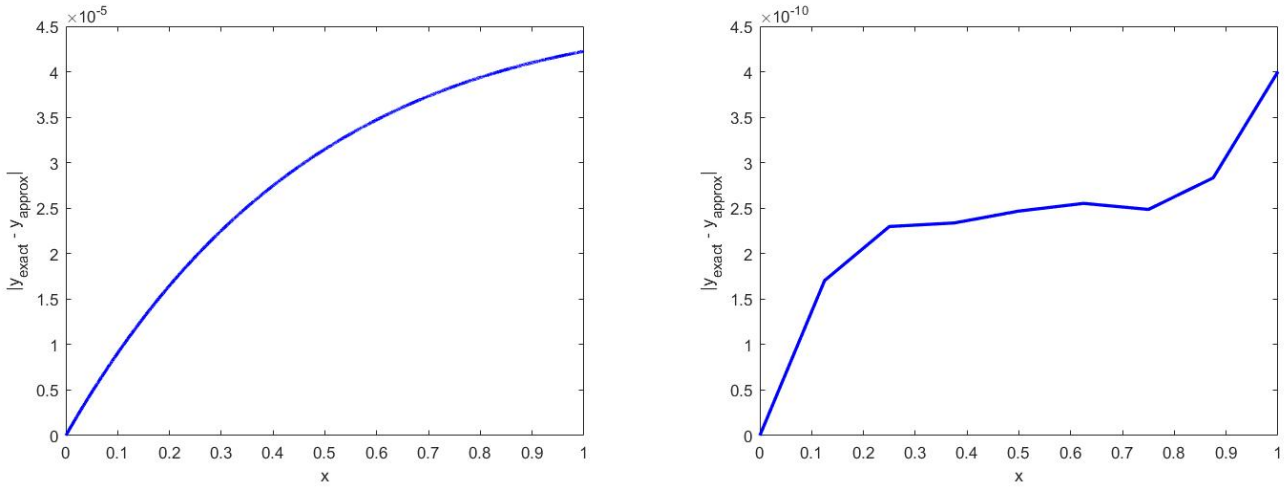
with the initial condition $y(0) = 1$. The exact solution of the problem is $y(x) = e^{-x}$.

Using DJM and CLSM, we determine the approximate solutions and their absolute errors. We compare the outcomes of both methods with the approximate solutions produced by an innovative numerical approach, detailed in Table 1. A visual comparison of the absolute error is illustrated in Figure 1.

Table 1: Comparison of approximate solutions and absolute errors for Example 3.1

x	Exact	Present Study		Present Study		Previous Study [18]	
		DJM	Abs. Error	CLSM	Abs. Error	Approx. Sol.	Abs. Error
0.000	1.000000000	1.000000000	0	1.000000000	0	1.000000000	0
0.125	0.882496902	0.882495520	1.382×10^{-6}	0.882496903	1.704×10^{-10}	0.879922493	2.574×10^{-3}
0.250	0.778800783	0.778795868	4.916×10^{-6}	0.778800783	2.298×10^{-10}	0.778748879	5.190×10^{-5}
0.375	0.687289278	0.687279403	9.876×10^{-6}	0.687289279	2.337×10^{-10}	0.694229264	6.939×10^{-3}
0.500	0.606530659	0.606514918	1.574×10^{-5}	0.606530660	2.468×10^{-10}	0.623990032	1.746×10^{-2}
0.625	0.535261428	0.535239288	2.214×10^{-5}	0.535261429	2.553×10^{-10}	0.565788861	3.053×10^{-2}
0.750	0.472366552	0.472337746	2.881×10^{-5}	0.472366553	2.486×10^{-10}	0.517629831	4.526×10^{-2}
0.875	0.416862019	0.416826467	3.555×10^{-5}	0.416862020	2.834×10^{-10}	0.477799240	6.094×10^{-2}
1.000	0.367879441	0.367837189	4.225×10^{-5}	0.367879442	4.002×10^{-10}	0.444850916	7.697×10^{-2}

Table 1 presents a comparison of approximate solutions and absolute errors for Example 1 using the methodsDJM and CLSMalongside a numerical method using a finite difference scheme with integral identities and quasilinearization [18], with the exact solution as a benchmark. The CLSM yields highly accurate results, with absolute errors consistently around 10^{-10}, closely matching the exact values. DJM also shows strong performance, maintaining a maximum error of 4.225×10^{-5} at $x = 1.000$, which is significantly lower than the reference method's error of 7.697×10^{-2}. Although DJMs error increases slightly with x, it remains far more accurate than the reference. In contrast, CLSM produces minimal error across all points. These results highlight that both proposed methods, particularly CLSM, provide substantial improvements in accuracy over previous approaches and are effective for solving nonlinear integro-differential equations.



(a) Daftardar-Gejji and Jafari Method **(b)** Chebyshev Least Square Method

Figure 1: Absolute Errors obtained using (a) the Daftardar-Gejji and Jafari Method and (b) the Chebyshev Least Square Method for Example 3.1.

Example 3.2. Here we consider a second-order linear Volterra integro-differential equation,

$$y''(x) = 1 + xe^x - \int_0^x e^{x-t} y(t)\, dt,$$

with the initial conditions $y(0) = 0, y'(0) = 1$. The exact solution of the problem is $y(x) = e^x - 1$.

We evaluate and compare the approximate solutions and absolute errors obtained through the DJM and CLSM approaches.The outcomes of both methods are presented alongside the numerical results derived from the Adomian Decomposition and Variational Iteration methods [10], in Table 2.

Table 2: Comparison of approximate solutions and absolute errors for Example 3.2

x	Exact	Present Study				Previous Study			
		DJM	Abs. Error	CLSM	Abs. Error	ADM [10]	Abs. Error	VIM [10]	Abs. Error
0.0	0.00000000	0.00000000	0	0.00000000	0	0.00000000	0	0.00000000	0
0.1	0.10517092	0.10517092	9.00×10^{-13}	0.10517100	8.62×10^{-8}	0.10517093	9.80×10^{-9}	0.10517096	4.60×10^{-8}
0.2	0.22140276	0.22140276	8.10×10^{-12}	0.22140561	2.85×10^{-6}	0.22140278	2.19×10^{-8}	0.22140282	6.25×10^{-8}
0.3	0.34985881	0.34985881	3.03×10^{-11}	0.34988122	2.24×10^{-5}	0.34985874	6.53×10^{-8}	0.34985869	1.21×10^{-8}
0.4	0.49182470	0.49182470	7.92×10^{-11}	0.49192244	9.77×10^{-5}	0.49182471	6.53×10^{-8}	0.49182457	1.28×10^{-8}
0.5	0.64872127	0.64872127	1.70×10^{-10}	0.64903005	3.09×10^{-4}	0.64872124	1.00×10^{-8}	0.64872131	4.05×10^{-8}
0.6	0.82211880	0.82211880	3.24×10^{-10}	0.82291440	7.96×10^{-4}	0.82211883	3.11×10^{-8}	0.82211886	5.91×10^{-8}
0.7	1.01375271	1.01375271	5.64×10^{-10}	1.01553376	1.78×10^{-3}	1.01375273	2.88×10^{-8}	1.01375277	6.20×10^{-8}
0.8	1.22554093	1.22554093	9.00×10^{-10}	1.22913829	3.60×10^{-3}	1.22554104	2.50×10^{-8}	1.22554098	5.50×10^{-8}
0.9	1.45960311	1.45960311	1.40×10^{-9}	1.46632034	6.72×10^{-3}	1.45960327	1.09×10^{-7}	1.45960315	3.70×10^{-8}
1.0	1.71828183	1.71828183	2.10×10^{-9}	1.73007179	1.18×10^{-2}	1.71828230	1.59×10^{-7}	1.71828193	9.89×10^{-8}

Table 2 shows the exact solution as a benchmark and the absolute errors are calculated from this exact solution. The DJM maintains excellent accuracy, with absolute errors remaining below 10^{-9} for all values of x. In contrast, the CLSM method, while still accurate, shows a gradual increase in error, reaching a maximum of 1.18×10^{-2} at $x = 1.0$. CLSMs performance is comparable to ADM and VIM for lower values of x, but its error grows faster with increasing x. Both ADM and VIM show relatively small errors (in the order of 10^{-8} to 10^{-7}), but the DJM consistently outperforms them throughout the domain. This table suggests that DJM is the most stable and accurate among the four methods.

Example 3.3. Here we consider a second-order nonlinear Volterra integro-differential equation,

$$y''(x) = \sinh(x) + \frac{1}{2}x + \frac{1}{2}\cosh(x)\sinh(x) - \int_0^x \left[y'(t)\right]^2 \, dt,$$

with the initial condition $y(0) = 0, y'(0) = 1$. The exact solution of the example is $y(x) = \sinh(x)$.

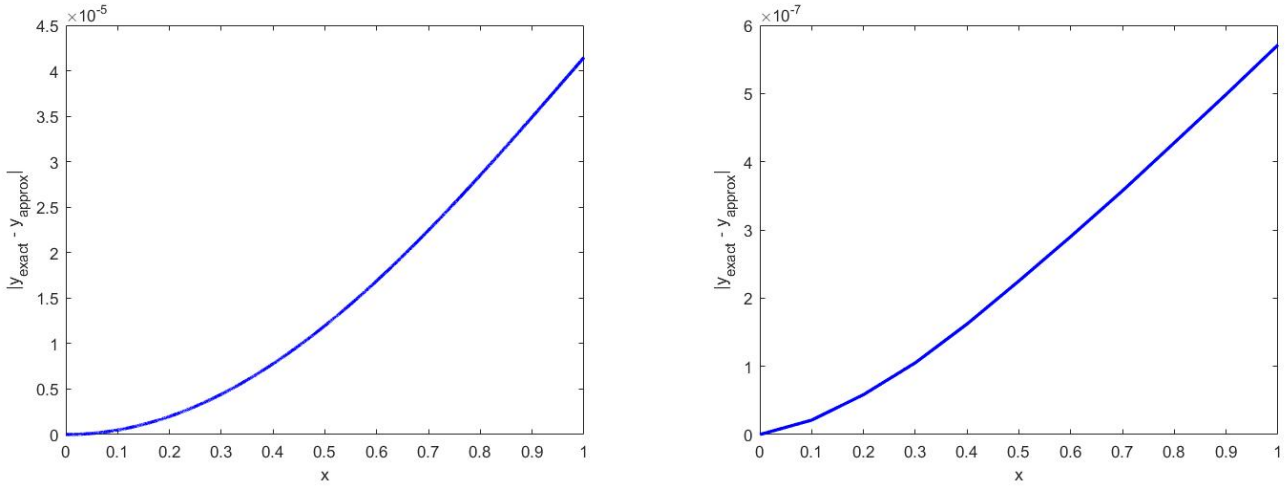
To assess the performance of the DJM and CLSM methods, we compare their approximate solutions and absolute errors. These findings are then compared with the numerical results from the Series Solution method, as shown in Table 3. The absolute errors are also visualized in Figure 2.

Table 3: Comparison of approximate solutions and absolute errors for Example 3.3

x	Exact	Present Study		Present Study		Previous Study [14]	
	solution	DJM	Abs. Error	CLSM	Abs. Error	SSM	Abs. Error
0.0	0.00000000	0.00000000	0	0.00000000	0	0.00000000	0
0.1	0.10016675	0.10016670	4.99×10^{-8}	0.10016673	2.11×10^{-8}	0.10016675	0
0.2	0.20133600	0.20133561	3.97×10^{-7}	0.20133594	5.85×10^{-8}	0.20133600	0
0.3	0.30452029	0.30451896	1.33×10^{-6}	0.30452019	1.05×10^{-7}	0.30452029	6.00×10^{-11}
0.4	0.41075233	0.41074921	3.11×10^{-6}	0.41075216	1.62×10^{-7}	0.41075233	7.20×10^{-10}
0.5	0.52109531	0.52108932	5.99×10^{-6}	0.52109508	2.25×10^{-7}	0.52109530	5.39×10^{-9}
0.6	0.63665358	0.63664344	1.01×10^{-5}	0.63665329	2.90×10^{-7}	0.63665355	2.79×10^{-8}
0.7	0.75858370	0.75856797	1.57×10^{-5}	0.75858334	3.57×10^{-7}	0.75858359	1.12×10^{-7}
0.8	0.888810598	0.88808316	2.28×10^{-5}	0.88810555	4.28×10^{-7}	0.88810561	3.72×10^{-7}
0.9	1.02651673	1.02648530	3.14×10^{-5}	1.02651623	4.99×10^{-7}	1.02651561	1.08×10^{-6}
1.0	1.17520119	1.17515972	4.15×10^{-5}	1.17520062	5.71×10^{-7}	1.17519841	2.78×10^{-6}

Table 3 illustrates the comparative performance of the DJM and CLSM methods with the SSM technique from [14], using the exact solution as a reference. Both DJM and CLSM show excellent agreement with the exact solution across the domain $x \in [0, 1]$. DJM maintains low absolute errors, ranging from 4.99×10^{-8} to 4.15×10^{-5}, while CLSM achieves even higher precision, with errors below 6×10^{-7} throughout. Although the SSM method shows zero or near-zero error in early values of x, its accuracy diminishes slightly at larger x values, with errors reaching up to 2.78×10^{-6} at $x = 1.000$. Notably, CLSM consistently provides

smaller errors than SSM for all non-zero values of x, confirming its robustness and reliability. Overall, the results demonstrate that both present methods are highly accurate, with CLSM offering a slight edge in stability and precision over existing approaches in this example.



(a) Daftardar-Gejji and Jafari Method **(b)** Chebyshev Least Square Method

Figure 2: Absolute Errors obtained using the (a) Daftardar-Gejji and Jafari Method and (b) the Chebyshev Least Square Method for Example 3.3.

Example 3.4. Here we consider a third-order linear Volterra integro-differential equation,

$$y'''(x) = -1 + \int_0^x y(t)\,dt,$$

with the initial condition $y(0) = 1, y'(0) = 1, y''(0) = -1$. The exact solution of the example is $\cos(x) + \sin(x)$.

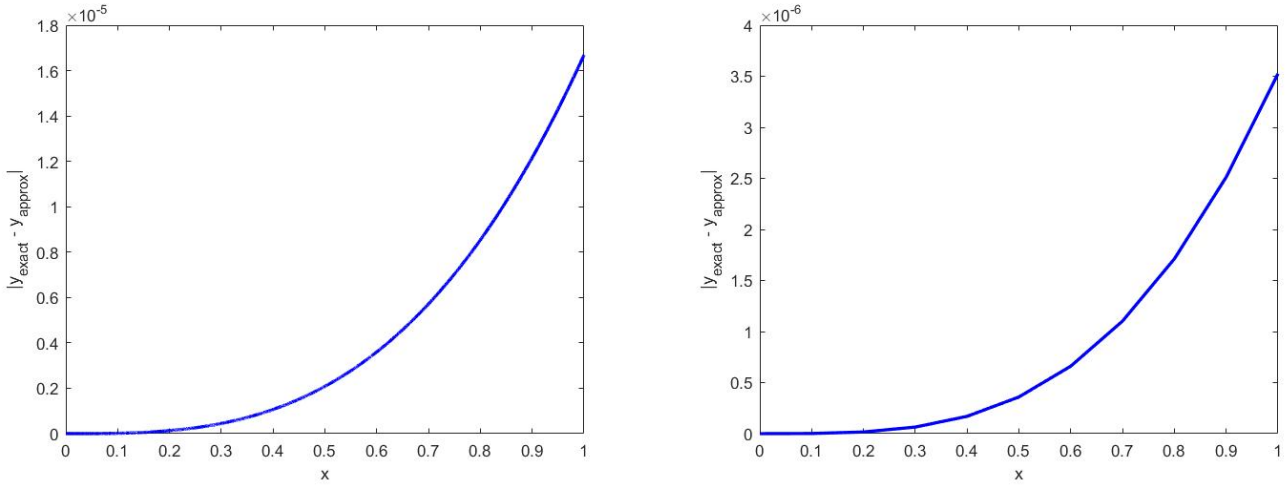
A comparative analysis of the approximate solutions and the absolute errors derived from the DJM and CLSM is provided in Table 4. In addition, the absolute error is depicted graphically in Figure 3.

Table 4: Error Approximation for Example 3.4

x	Exact Solution	DJM	Abs. Error	CLSM	Abs. Error
0.0	1.00000000	1.00000000	0	1.00000000	0
0.1	1.09483758	1.09483758	1.67×10^{-9}	1.09483758	1.63×10^{-9}
0.2	1.17873591	1.17873594	2.67×10^{-8}	1.17873593	1.64×10^{-8}
0.3	1.25085670	1.25085683	1.40×10^{-7}	1.25085676	6.43×10^{-8}
0.4	1.31047934	1.31047976	4.30×10^{-7}	1.31047951	1.70×10^{-7}
0.5	1.35700810	1.35700914	1.04×10^{-6}	1.35700846	3.59×10^{-7}
0.6	1.38997809	1.38998025	2.16×10^{-6}	1.38997875	6.61×10^{-7}
0.7	1.40905987	1.40906388	4.00×10^{-6}	1.40906098	1.10×10^{-6}
0.8	1.41406280	1.41406963	6.83×10^{-6}	1.41406451	1.71×10^{-6}
0.9	1.40493688	1.40494782	1.09×10^{-5}	1.40493939	2.51×10^{-6}
1.0	1.38177329	1.38178998	1.67×10^{-5}	1.38177682	3.53×10^{-6}

Table 4 reveals that both methods demonstrate high accuracy across the entire interval, especially at lower values of x, where the absolute errors remain close to zero. As x increases, a gradual rise in error is observed for both techniques, which is expected due to cumulative approximation effects. However,

the DJM maintains absolute errors within the order of 10^{-5}, whereas CLSM consistently achieves better accuracy, keeping errors notably lowernever exceeding 3.53×10^{-6}. This consistent advantage highlights CLSMs superior numerical stability, especially for larger values of x. Despite the slight increase in error for both methods toward $x = 1.0$, the approximations remain highly reliable. These results affirm the effectiveness of DJM and further reinforce the precision of CLSM in solving linear problems with smooth solution behavior.



(a) Daftardar-Gejji and Jafari Method **(b)** Chebyshev Least Square Method

Figure 3: Absolute Errors obtained using (a) the Daftardar-Gejji and Jafari Method and (b) the Chebyshev Least Square Method for Example 3.4.

Example 3.5. Here we consider a fourth-order linear Volterra integro-differential equation,

$$y^{(4)}(x) = -x + 5e^x - 1 + \int_0^x y(t)\,dt,$$

with the boundary conditions $y(0) = 1, y'(0) = 1, y(1) = 1 + e, y'(1) = 2e$. The exact solution of the problem is $y(x) = 1 + xe^x$.

Approximate solutions and absolute errors are calculated using DJM and CLSM, and the results are compared to assess accuracy. We also compare the results of the two approaches with numerical approximations obtained via the Picard-Greens Embedding method [5], as summarized in Table 5.

Table 5: Comparison of approximate solutions and absolute errors for Example 3.5

x	Exact	Present Study				Previous Study [5]	
		DJM	Abs. Error	CLSM	Abs. Error	PGEM	Abs. Error
0.0	1.00000000	1.00000000	0	1.00000000	0	-	-
0.1	1.11051709	1.11051709	4.00×10^{-11}	1.11051708	1.57×10^{-8}	1.11119245	1.68×10^{-6}
0.2	1.24428055	1.24428055	1.32×10^{-9}	1.24428027	2.79×10^{-7}	-	-
0.3	1.40495764	1.40495765	1.01×10^{-8}	1.40495615	1.49×10^{-6}	1.40863488	1.05×10^{-5}
0.4	1.59672988	1.59672992	4.26×10^{-8}	1.59672504	4.84×10^{-6}	-	-
0.5	1.82436064	1.82436077	1.30×10^{-7}	1.82434881	1.18×10^{-5}	1.82957259	1.70×10^{-5}
0.6	2.09327128	2.09327160	3.24×10^{-7}	2.09324725	2.40×10^{-5}	-	-
0.7	2.40962690	2.40962759	6.99×10^{-7}	2.40958392	4.29×10^{-5}	2.41330476	1.35×10^{-5}
0.8	2.78043274	2.78043411	1.36×10^{-6}	2.78036297	6.98×10^{-5}	-	-
0.9	3.21364280	3.21364526	2.46×10^{-6}	3.21353779	1.05×10^{-4}	3.21431838	2.74×10^{-6}
1.0	3.71828183	3.71828599	4.16×10^{-6}	3.71813321	1.48×10^{-4}	-	-

Table 5 shows that from the present study and the PGEM method from [5], DJM maintains outstanding precision throughout the interval, with absolute errors starting as low as 4.00×10^{-11} and gradually

increasing to just 4.16×10^{-6} at $x = 1.0$. In contrast, the CLSM shows comparatively higher errors, particularly at larger x values, where it reaches 1.48×10^{-4}, although still within acceptable bounds for practical applications. The PGEM method shows moderate accuracy, but again, DJM outperforms PGEM method in terms of error minimization. Notably, for all available data points, DJM surpasses both CLSM and PGEM in maintaining proximity to the exact solution. This example further confirms DJMs capability as a highly accurate and stable method, especially in cases involving rapid solution growth.

Example 3.6. Here we consider a fourth-order nonlinear Volterra integro-differential equation,

$$y^{(4)}(x) = 1 + \int_0^x e^{-ty^2(t)}\, dt,$$

with the boundary conditions $y(0) = 1, y'(0) = 1, y(1) = e, y'(1) = e$. The exact solution of the problem is $y(x) = e^x$.

Table 6: Comparison of approximate solutions and absolute errors for Example 3.6

x	Exact	Present Study				Previous Study [5]	
		DJM	Abs. Error	CLSM	Abs. Error	PGEM	Abs. Error
0.0	1.00000000	1.00000000	0	1.00000000	0	-	-
0.1	1.10517092	1.10517092	4.00×10^{-11}	1.10517086	5.48×10^{-8}	1.10517092	5.85×10^{-7}
0.2	1.22140276	1.22140276	1.09×10^{-9}	1.22140096	1.80×10^{-6}	-	-
0.3	1.34985881	1.34985882	7.49×10^{-9}	1.34984478	1.40×10^{-5}	1.34985880	3.66×10^{-6}
0.4	1.49182740	1.49182474	2.89×10^{-8}	1.49176401	6.07×10^{-5}	-	-
0.5	1.64872127	1.64872140	7.89×10^{-8}	1.64853111	1.90×10^{-4}	1.64872126	5.93×10^{-6}
0.6	1.82211880	1.82211912	1.78×10^{-7}	1.82163293	4.86×10^{-4}	-	-
0.7	2.01375271	2.01375341	3.48×10^{-7}	2.01267428	1.08×10^{-3}	2.01375270	4.71×10^{-6}
0.8	2.22554093	2.22554229	6.13×10^{-7}	2.22338156	2.16×10^{-3}	-	-
0.9	2.45960311	2.45960557	1.00×10^{-6}	2.45560630	4.00×10^{-3}	2.45960311	9.55×10^{-7}
1.0	2.71828183	2.71828600	1.53×10^{-6}	2.71132881	6.95×10^{-3}	-	-

Table 6 compares the approximate solutions and absolute errors for Example 6 obtained via the proposed DJM and CLSM methods with the Picard-Green Embedding method from [5], using the exact solution as the reference. The DJM consistently delivers high accuracy, with errors remaining below 2×10^{-6} even at $x = 1.0$. CLSM, while still reasonably accurate, shows a noticeable increase in error as x grows, reaching up to 6.95×10^{-3} at the upper bound, which is higher than both DJM and PGEM in most cases. PGEM generally performs well, but DJM surpasses it by maintaining smaller errors across the domain. This example further highlights DJMs reliability and precision in approximating solutions with exponential growth, whereas CLSMs accuracy tends to degrade faster for larger values of x.

4. Computational Efficiency and Stability Criteria

Table 7 presents the computational performance of the DJM in terms of iteration count, CPU time, residual norm, and L^∞ norm. The results show that increasing the number of iterations from $M = 100$ to $M = 500$ improves the accuracy for all test problems. In each case, both the residual norms and maximum errors decrease noticeably, indicating stable convergence behavior of the method. For lower-order as well as higher-order problems, DJM maintains good numerical stability without any sudden increase in error. The reduction in L^∞ error is particularly clear in Example 3.1 and 3.3, where the errors decrease from the order of 10^{-3} to 10^{-4}. Similarly, the residual norms for all problems become much smaller at higher iteration numbers, showing better accuracy. The CPU time increases with the number of iterations, which is expected due to the additional computational steps. However, the overall computational cost remains relatively low for all examples. Even for higher-order nonlinear problems, the method achieves good accuracy within a short computational time. Overall, the results demonstrate that DJM is computationally

efficient, stable, and reliable for solving both linear and nonlinear Volterra integro-differential equations. The steady reduction in residuals and errors with increasing iteration number confirms the robustness and convergence capability of the method.

Table 7: Computational Efficiency and Stability Metrics for DJM (M is iteration number)

Example	**Order/Type**	**M**	**CPU Time (s)**	**Residual Norm**	**L^∞ Norm**
Example 3.1	1st Order Nonlinear	100	0.003 000	5.12×10^{-3}	4.23×10^{-3}
		500	0.008 221	1.01×10^{-3}	8.45×10^{-4}
Example 3.2	2nd Order Linear	100	0.002 899	1.35×10^{-2}	2.16×10^{-5}
		500	0.035 003	2.71×10^{-3}	8.61×10^{-7}
Example 3.3	2nd Order Nonlinear	100	0.005 880	5.83×10^{-3}	4.14×10^{-3}
		500	0.032 683	1.17×10^{-3}	8.29×10^{-4}
Example 3.4	3rd Order Linear	100	0.006 319	7.01×10^{-3}	1.68×10^{-3}
		500	0.035 016	1.41×10^{-3}	3.34×10^{-4}
Example 3.5	4th Order Linear	100	0.006 193	4.03×10^{-2}	3.92×10^{-4}
		500	0.034 539	8.13×10^{-3}	8.23×10^{-5}
Example 3.6	4th Order Nonlinear	100	0.006 199	1.35×10^{-2}	4.11×10^{-4}
		500	0.036 473	2.71×10^{-3}	8.31×10^{-5}

In Table 8, the computational performance of CLSM for different polynomial orders is tabulated. The results show that the accuracy of CLSM strongly depends on the selected polynomial degree. For $N = 5$, the method performs reasonably well for simpler problems, but relatively large errors and residuals appear in some higher-order problems, especially Example 3.4 and 3.5. When the polynomial degree is increased to $N = 8$, the accuracy improves significantly for most cases. In particular, Example 3.2 and 3.4 show major reductions in both residual and L^∞ norms. For example, in Example 3.4, the error decreases from 4.28×10^{-1} to 3.53×10^{-6}. Similarly, Example 3.2 achieves an error of order 10^{-8}, showing the strong approximation capability of CLSM at higher degree polynomial. The CPU time generally increases with increased polynomial degree, although the computational cost remains acceptable for all problems. The residual norms at $N = 8$ are also much smaller in most cases, indicating improved numerical stability and convergence. However, the improvement is not the same for all examples. In Example 3.5, the error decreases, but the residual norm remains comparatively large. Example 3.6 shows almost no improvement in the L^∞ error when increasing the polynomial degree from $N = 5$ to $N = 8$. This suggests that the nonlinear structure of the problem may affect the efficiency of the polynomial approximation. Overall, CLSM provides highly accurate results when a suitable polynomial order is used, but its performance is more sensitive to the choice of polynomial degree compared to DJM.

Table 8: Computational Efficiency and Stability Metrics for CLSM (N is polynomial order)

Example	Order/Type	N	CPU Time (s)	Residual Norm	L^∞ Norm
Example 3.1	1st Order Nonlinear	5	0.005 202	1.12×10^{-16}	7.41×10^{-5}
		8	0.052 080	8.31×10^{-15}	4.00×10^{-10}
Example 3.2	2nd Order Linear	5	0.091 248	7.30×10^{-3}	3.71×10^{-4}
		8	0.432 745	7.32×10^{-7}	1.67×10^{-8}
Example 3.3	2nd Order Nonlinear	5	0.007 449	3.64×10^{-12}	9.33×10^{-5}
		8	0.020 951	7.11×10^{-9}	6.71×10^{-7}
Example 3.4	3rd Order Linear	5	0.015 561	1.92×10^{1}	4.28×10^{-1}
		8	0.191 967	1.63×10^{-4}	3.53×10^{-6}
Example 3.5	4th Order Linear	5	0.010 547	1.49×10^{1}	7.17×10^{-2}
		8	0.172 804	3.31×10^{-2}	1.49×10^{-4}
Example 3.6	4th Order Nonlinear	5	0.065 409	1.23×10^{-16}	6.95×10^{-3}
		8	0.144 001	1.23×10^{-16}	6.95×10^{-3}

5. Conclusion

This study investigates the effectiveness of the Daftardar-Gejji and Jafari Method (DJM) and the Chebyshev Least Squares Method (CLSM) for solving higher-order linear and nonlinear Volterra integro differential equations. The convergence and stability behavior of the proposed methods are further discussed using fixed-point arguments for DJM and spectral approximation properties of CLSM. The numerical results obtained using absolute error, residual behaviors, L^∞ norm, and CPU time confirm the effectiveness of the proposed methods are in excellent agreement with exact solutions. The DJM approach provides a flexible and simple framework based on recursive schemes, suitable for nonlinear kernels, but slower for stiff problems. On the other hand, the CLSM provides high-order accuracy for smooth kernels, but involves global matrix inversion. This study shows the feasibility of DJM and CLSM as efficient tools for solving high-order memory-dependent integro-differential systems. Future work may focus on incorporating additional analytical benchmark solutions and extending the proposed framework to multidimensional and more complex memory-dependent systems.

CRediT authorship contribution statement

Maria Rahman: Writing – original draft, Conceptualization, Investigation, Software, Data Curation, Formal analysis, Visualization, Validation. **Dipta Roy:** Writing – original draft, Methodology, Investigation, Formal Analysis, Visualization. **Faria Tasnim:** Conceptualization, Investigation. **Tania S. Khaleque:** Writing – review & editing, Formal analysis, Supervision, Project administration.

References

[1] Abed, A. M., Younis, M. F., & Hamoud, A. A. (2022). Numerical solutions of nonlinear VolterraFredholm integro-differential equations by using MADM and VIM. Nonlinear Functional Analysis and Applications, 189201. 1

[2] Adebisi, F., Ojurongbe, T. A., Okunlola, K. A., & Peter, O. J. (2021). Application of Chebyshev polynomial basis function on the solution of Volterra integro-differential equations using Galerkin method. Mathematics and Computational Sciences, 2(4), 4151. 1, 2.2

[3] Adebisi, A. F., & Okunola, K. A. (2025). Solution of VolterraFredholm integro-differential equations using Chebyshev least square method. Science World Journal, 20(1), 271275. 1, 2.2

[4] Ahmed, A. (2025). Solving nonlinear fractional autonomous dynamical systems using a LaplaceDaftardarJafari method. Mathematics and Computational Sciences, 6(3), 143152. 1

[5] Akgun, F. A., & Rasulov, Z. (2025). Numerical method for solution of fourth-order Volterra integro-differential equations by Greens function. Kragujevac Journal of Mathematics, 49(3), 485494. 1, 3, 5, 3, 6, 3

[6] Akgun, F. A., & Rasulov, Z. (2021). Generalized iteration method for the solution of fourth order BVP via Greens function. European Journal of Pure and Applied Mathematics, 14(3). 1
[7] Akram, G., & Rehman, H. (2014). Homotopy perturbation method with reproducing kernel method for third order nonlinear boundary value problems. Journal of Basic and Applied Science Research, 4(1), 6067. 1
[8] Al-Hayani, W. (2017). DaftardarJafari method for fractional heat-like and wave-like equations with variable coefficients. Applied Mathematics, 8(2), 215228. 2.1
[9] Al-Mdallal, Q. M. (2011). Monotone iterative sequences for nonlinear integro-differential equations of second order. Nonlinear Analysis: Real World Applications, 12(6), 36653673. 1
[10] Alao, S., Akinboro, F. S., Akinpelu, F. O., & Oderinu, R. (2014). Numerical solution of integro-differential equation using Adomian decomposition and variational iteration methods. IOSR Journal of Mathematics, 10(4), 1822. 3, 2
[11] Ali, K. K., Cattani, C., Gómez-Aguilar, J. F., Baleanu, D., & Osman, M. S. (2020). Analytical and numerical study of the DNA dynamics arising in oscillator-chain of PeyrardBishop model. Chaos, Solitons & Fractals, 139, 110089. 1
[12] Ali, K. K., Abd El Salam, M. A., Mohamed, E. M., Samet, B., Kumar, S., & Osman, M. S. (2020). Numerical solution for generalized nonlinear fractional integro-differential equations with linear functional arguments using Chebyshev series. Advances in Difference Equations, 2020, 123. 1
[13] Ali, F., Ali, J., & Uddin, I. (2021). A novel approach for the solution of BVPs via Greens function and fixed point iterative method. Journal of Applied Mathematics and Computation, 66(1), 167181. 1
[14] Ansari, A., Ahmad, N., & Ali, A. H. (2024). Numerical study of the series solution method to analysis of Volterra integro-differential equations. Journal of Applied Mathematics and Informatics, 42(4), 899913. 2.1, 3
[15] Avudainayagam, A., & Vani, C. (2000). WaveletGalerkin method for integro-differential equations. Applied Numerical Mathematics, 32(3), 247254. 1
[16] Brunner, H. (1989). The numerical treatment of Volterra integro-differential equations with unbounded delay. Journal of Computational and Applied Mathematics, 28, 523. 1
[17] Boyd, J. P. (2001). Chebyshev and Fourier spectral methods. Dover Publications. 2.2
[18] Cakir, M., Gunes, B., & Duru, H. (2021). A novel computational method for solving nonlinear Volterra integro-differential equation. Kuwait Journal of Science, 48(1). 1, 1, 3
[19] Canuto, C., Hussaini, M. Y., Quarteroni, A., & Zang, T. A. (2006). Spectral methods (Vol. 285). Springer. 2.2
[20] El-Sayed, S. M., & Abdel-Aziz, M. R. (2003). A comparison of Adomians decomposition method and wavelet-Galerkin method for solving integro-differential equations. Applied Mathematics and Computation, 136(1), 151159. 1
[21] Hamoud, A. A., & Ghadle, K. P. (2018). Existence and uniqueness of the solution for VolterraFredholm integro-differential equations. Journal of the Siberian Federal University. Mathematics and Physics, 11(6), 692701. 1
[22] Hosseini, S. M., & Shahmorad, S. (2003). Tau numerical solution of Fredholm integro-differential equations with arbitrary polynomial bases. Applied Mathematical Modelling, 27(2), 145154. 1
[23] Jeffery, G. B. (1914). On the form of the Volterra equations. Proceedings of the Royal Society A, 90(619), 204215. 1
[24] Jiang, Y., & Ma, J. (2013). Spectral collocation methods for Volterra integro-differential equations with noncompact kernels. Journal of Computational and Applied Mathematics, 244, 115124. 1
[25] Kot, M. (2001). Elements of mathematical ecology. Cambridge University Press. 1
[26] Kopeikin, I. D., & Shishkin, V. P. (1984). Integral form of the general solution of equations of steady-state thermoelasticity. Journal of Applied Mathematics and Mechanics, 48(1), 117119. 1
[27] Kreyszig, E. (1991). Introductory functional analysis with applications. John Wiley & Sons. 2.1, 2.1
[28] Liu, Z., Sun, J., & Szántó, I. (2013). Monotone iterative technique for RiemannLiouville fractional integro-differential equations with advanced arguments. Results in Mathematics, 63(3), 12771287. 1
[29] Lotka, A. J. (1939). On an integral equation in population analysis. The Annals of Mathematical Statistics, 10(2), 144161. 1
[30] Maleknejad, K., & Mahmoudi, Y. (2003). Taylor polynomial solution of high-order nonlinear VolterraFredholm integro-differential equations. Applied Mathematics and Computation, 145(23), 641653. 1
[31] Mishra, A. K., Verma, L., Nikan, O., & Molavi-Arabshahi, M. (2025). Numerical pricing of European options under time-fractional BlackScholes equation in financial markets. Chaos: An Interdisciplinary Journal of Nonlinear Science, 35(8). 1
[32] Ndidiamaka, D. E., Oyedepo, T., & Adenipekun, A. E. (2024). Fourth-kind Chebyshev computational approach for integro-differential equations. Mathematics and Computational Sciences, 5(1), 5564. 1
[33] Nikan, O., Rashidinia, J., & Jafari, H. (2025). An improved local radial basis function method for pricing options under the time-fractional BlackScholes model. Journal of Computational Science, 89, 102610. 1
[34] Pougaza, D. B. (2007). The Lotka integral equation as a stable population model. AIMS, 130. 1
[35] Rashid, S., Kubra, K. T., Jafari, H., & Lehre, S. U. (2022). A semi-analytical approach for fractional order Boussinesq equation in gradient unconfined aquifers. Mathematical Methods in the Applied Sciences, 45(2), 10331062. 1
[36] Rashed, M. T. (2004). Lagrange interpolation to compute numerical solutions of differential, integral and integro-differential equations. Applied Mathematics and Computation, 151(3), 869878. 1
[37] Sahin, N., Yüzba, ., & Sezer, M. (2011). A Bessel polynomial approach for solving general linear Fredholm integro-differentialdifference equations. International Journal of Computer Mathematics, 88(14), 30933111. 1

[38] Sunthrayuth, P., Ullah, R., Khan, A., Shah, R., Kafle, J., Mahariq, I., & Jarad, F. (2021). Numerical analysis of the fractional-order nonlinear system of Volterra integro-differential equations. Journal of Function Spaces, 2021, Article ID 1537958. 1

[39] Tahernezhad, T., & Jalilian, R. (2020). Exponential spline for the numerical solutions of linear Fredholm integro-differential equations. Advances in Difference Equations, 2020, 115. 1

[40] Tisdell, C. C. (2019). On Picards iteration method to solve differential equations and a pedagogical space for otherness. International Journal of Mathematical Education in Science and Technology, 50(5), 788799. 1

[41] Trefethen, L. N. (2000). Spectral methods in MATLAB. SIAM. 2.2

[42] Verma, L., Meher, R., Nikan, O., & Al-Saedi, A. A. (2025). Numerical study on fractional order nonlinear SIR-SI model for dengue fever epidemics. Scientific Reports, 15(1), 30677. 1

[43] Volterra, V. (1910). Sulle equazioni integro-differenziali. Tipografia della R. Accademia dei Lincei. 1

[44] Volterra, V. (1930). Theory of functionals and of integral and integro-differential equations. Blackie & Son Limited. 1

[45] Zaboli, M., & Tajadodi, H. (2025). A spectral collocation method for solving stochastic fractional integro-differential equation. Mathematics and Computational Sciences, 6(2), 131. 1

[46] Zeidler, E. (2013). Nonlinear functional analysis and its applications II/B: nonlinear monotone operators. Springer. 2.1

[47] Zhao, J., & Corless, R. M. (2006). Compact finite difference method for integro-differential equations. Applied Mathematics and Computation, 177(1), 271288. 1