Journal of Innovative Applied Mathematics and Computational Sciences

J. Innov. Appl. Math. Comput. Sci. **5**(2) (2025), 253–290. DOI: 10.58205/jiamcs.v5i2.1953

http://jiamcs.univ-mila.dz/

A comparative study of analytical and numerical methods for solving systems of nonlinear Volterra integral equations with applications

Fatimah Ali Othman and Sizar Abid Mohammed ✉

Department of Mathematics, College of Basic Education, University of Duhok, Duhok, Iraq

Received July 8, 2025, Accepted December 24, 2025, Published January 08, 2026

Abstract. This paper presents a detailed comparison of three relatively recent methods for the numerical solution of systems of Nonlinear Volterra Integral Equations of the second kind (NVIEs-II): the Modified Adomian Decomposition Method (MADM), the Hussein-Jassim Method (H-JM), and the Cubic Non-Polynomial Spline Function Method (CNPSFM). The objective of this study is to evaluate the performance of these methods in terms of accuracy, convergence, and numerical stability. To achieve this, all three methods are applied to standard benchmark problems with known exact solutions, enabling quantitative assessment.

The numerical results reveal distinct performance characteristics for each method. Both MADM and H-JM demonstrate excellent performance, yielding solutions with high accuracy and very low errors, occasionally approaching machine precision. MADM exhibits rapid convergence, while H-JM provides robust numerical stability and ease of implementation. CNPSFM displays good numerical stability and accurately captures the overall solution behavior; however, it produces relatively larger errors, particularly as the integration interval lengthens.

This comparison concludes that the optimal choice among these methods is highly problem-specific. MADM and H-JM are best suited for high-precision applications requiring analytical insight (e.g., quantum mechanics or population dynamics), whereas CNPSFM remains viable for applications prioritizing solution smoothness over absolute accuracy. This study provides practical, evidence-based recommendations to assist researchers and engineers in selecting appropriate solvers for real-world systems modeled by NVIEs-II. Future research should extend these methods to systems with singularities and/or delays, which have been underexplored in the current literature. Another promising direction involves developing hybrid approaches that integrate artificial neural networks with traditional computational solvers.

Keywords: Nonlinear Volterra integral equations; modified Adomian decomposition method; Hussein-Jassim method; cubic non-polynomial spline function method; Maclaurin series.

2020 Mathematics Subject Classification: 39A10, 67B89, 47H30.

✉ Corresponding author. Email: sizar.mohammed@uod.ac

ISSN (electronic): 2773-4196

1 Introduction

Nonlinear Volterra Integral Equations of the second kind (NVIEs-II) represent a fundamental class of mathematical models adept at describing complex, memory-dependent dynamic systems across a wide spectrum of scientific and engineering applications [5,21]. These equations model phenomena in which the current state of a system is intrinsically dependent on its entire historical evolution, making them indispensable in fields such as population dynamics, radiative heat transfer, quantum mechanics, and viscoelastic material responses [14,18]. The challenge intensifies when these equations involve weakly singular kernels, an area that has attracted significant research interest [3,4,15,16].

Given their nonlinear and integral structure, analytical solutions to NVIEs-II are often intractable for all but the simplest cases [17]. This inherent complexity has driven the development of robust numerical and approximate analytical methods, which are central themes in computational mathematics. A diverse array of techniques has been explored, including the Adomian Decomposition Method (ADM) and its variants [1,19,22], variational iteration and homotopy-based approaches [6], and various spline-based collocation techniques [8,13,15,16]. Among these, Non-Polynomial Splines (NPS) have garnered particular attention due to the flexibility offered by their trigonometric or exponential basis functions, which can be highly effective for handling solutions with oscillatory behavior or singularities [7,11,20]. Despite these advancements, persistent challenges remain, including convergence instabilities exacerbated by strong nonlinearities [17], the need for specialized treatment of singular kernels [4], and a relative scarcity of comprehensive comparative studies that benchmark advanced variants against one another [10,12]. Recent progress in adapting NPS methods shows promise in addressing issues of singularity and improving convergence rates [3,13].

This study aims to bridge this gap by conducting a systematic investigation into solving systems of NVIEs-II using advanced analytical and numerical schemes, with a focus on ensuring accuracy, convergence, and computational efficiency. We rigorously examine three sophisticated methods:

- A Modified Adomian Decomposition Method (MADM) incorporating adaptive weights [1,19].
- The Hussein-Jassim Method (H-JM), which leverages a recursive Maclaurin series expansion framework [9].
- A Cubic Non-Polynomial Spline Function Method (CNPSFM), utilizing a trigonometric-exponential basis [8,13].

Our research makes significant contributions to the existing body of work in several key ways. We present, for the first time, a structured framework for applying both the MADM [1,19] and the CNPSFM [7,11] specifically to systems of NVIEs-II. Furthermore, we significantly extend the applicability of the H-JM to this important class of systems [9]. These developments are crucial, as previous studies have primarily focused on applying these powerful techniques to single equations [2,10,20].

In addition to these novel methodological extensions, this study seeks to:

- Integrate recent advances in formulations robust to singularities [13,15,16].
- Establish a benchmark against state-of-the-art non-spline methods, such as MADM and H-JM [9,19].

- Quantify the computational trade-offs associated with solving higher-order systems [14].

The principal objectives of this study are threefold:

(i) To perform a theoretical analysis of the convergence properties and establish a priori error bounds for the proposed numerical schemes [5].

(ii) To conduct an empirical evaluation and comparison of their accuracy, convergence rate, computational efficiency, and robustness when confronted with singularities or complex nonlinearities [12, 17].

(iii) To formulate evidence-based recommendations for selecting the most appropriate method for specific application contexts [8, 21].

Fulfilling these objectives will enhance the available toolkit for tackling complex systems of NVIEs-II, thereby facilitating advancements in their application across scientific and engineering disciplines [5].

2 Theoretical background

This section establishes the theoretical framework for this study. We begin by defining the specific class of integral equations under focus: NVIEs-II. We then introduce spline functions, which form the basis for one of the numerical methods explored.

2.1 Nonlinear Volterra integral equations of the second kind

An integral equation is one where the unknown function, $u(x)$, appears within an integral. We are specifically interested in Volterra integral equations of the second kind. These have a variable upper limit of integration and feature the unknown function both inside and outside the integral.

A single NVIE-II is generally expressed as:

$$u(x) = f(x) + \lambda \int_a^x K(x, t, u(t))\, dt, \tag{2.1}$$

where:

- λ is a constant parameter,
- $u(x)$ is the unknown function to be determined,
- $f(x)$ is a known function,
- $K(x, t, u(t))$ is the kernel, a known function that is nonlinear in $u(t)$,
- The interval $[a, x]$ signifies the "memory" or history-dependent nature of the system.

This study specifically addresses systems of NVIEs-II, which involve multiple coupled equations. A system of two equations, for instance, takes the form:

$$\begin{cases} u(x) = f_1(x) + \int_0^x \left[K_{11}(x, t, u(t), v(t))\right] dt, \\ v(x) = f_2(x) + \int_0^x \left[K_{21}(x, t, u(t), v(t))\right] dt, \end{cases} \tag{2.2}$$

where K_{11} and K_{21} are nonlinear kernels coupling the unknown functions $u(x)$ and $v(x)$.

Due to their nonlinear and integral nature, these systems are extremely challenging to solve analytically. This provides the impetus for developing and analyzing robust numerical and semi-analytic methods, which is the main focus of this thesis.

2.2 Numerical approximation using spline functions

One powerful approach for numerically solving integral equations is the use of spline functions. A spline is a piecewise-defined function valued for its smoothness and flexibility in approximating complex behaviors.

Core concepts of spline functions

Given a partition $\triangle : a = x_0 < x_1 < \cdots < x_n = b$ of an interval $[a, b]$, a spline function $S(x)$ of degree k satisfies:

1. It is a polynomial of degree at most k on each subinterval $[x_i, x_{i+1}]$.
2. It is continuously differentiable up to order $k - 1$ on $[a, b]$, ensuring smoothness at the junction points (knots).

While classical splines use polynomial bases, NPS incorporate trigonometric or exponential terms to enhance accuracy, especially for oscillatory or non-standard solution behaviors.

The cubic non-polynomial spline used in this study is defined over each subinterval $[x_i, x_{i+1}]$ as:

$$S_i(x) = a_i \cos(\omega(x - x_i)) + b_i \sin(\omega(x - x_i)) + c_i(x - x_i) + d_i(x - x_i)^2 + e_i,$$

where a_i, b_i, c_i, d_i, and e_i are coefficients determined by enforcing continuity and differentiability conditions at the knots, and ω is a frequency parameter. This structure provides a flexible basis for numerically solving integral equations and forms the foundation for the CNPSFM method analyzed in our comparative study.

3 Methodology

This section presents the mathematical framework for solving systems of NVIEs-II. We introduce three advanced methods: MADM, H-JM, and CNPSFM.

The unified structure for each method includes:

1. Fundamental theory and formulation,
2. Algorithmic steps,
3. Convergence analysis, and
4. Unique advantages for handling nonlinear systems.

3.1 Modified Adomian techniques for solving a system of NVIEs-II

This section presents our enhanced ADM designed for the efficient solution of coupled NLVIEs. The proposed modifications, strategic splitting of nonhomogeneous terms and adaptive series expansion, significantly improve convergence rates and computational efficiency compared to the standard ADM, as summarized in Table 3.1.

Remark 3.1. All three numerical techniques discussed in this study MADM, H-JM, and CNPSFM are applied to the same general system of coupled NVIEs-II , ensuring consistent and comparable numerical analysis.

Table 3.1: Key enhancements of MADM vs. standard ADM

Feature	Standard ADM	MADM (Our Approach)
Nonhomogeneous term handling	Direct use	Smart splitting/series expansion
Convergence rate	Linear	Superlinear
Computational load	High	Reduced
Coupled systems	Limited efficiency	Highly efficient

In this section, we broaden the application of the MADM to address a system of NVIEs-II, presented in the general form:

$$u(x) = f_1(x) + \int_0^x \left[k_{11}(x,t)F_{11}(u(t)) + k_{12}(x,t)F_{12}(v(t))\right] dt, \tag{3.1}$$

$$v(x) = f_2(x) + \int_0^x \left[k_{21}(x,t)F_{21}(u(t)) + k_{22}(x,t)F_{22}(v(t))\right] dt. \tag{3.2}$$

Here, $F_{11}(u(t))$, $F_{12}(v(t))$, $F_{21}(u(t))$, $F_{22}(v(t))$ are nonlinear functions of the unknowns $u(x)$ and $v(x)$, and $k_{ij}(x,t)$ $(i,j = 1,2)$ denote continuous kernels associated with the integral operators [21].

3.1.1 Standard ADM

We assume the solutions $u(x)$ and $v(x)$ can be expressed as infinite series:

$$u(x) = \sum_{n=0}^{\infty} u_n(x), \quad v(x) = \sum_{n=0}^{\infty} v_n(x). \tag{3.3}$$

The nonlinear terms in the system are decomposed using Adomian polynomials. The general formula for these polynomials is given by:

$$A_n^{ij} = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[F_{ij}\left(\sum_{k=0}^{n}\lambda^k \phi_k\right)\right]_{\lambda=0}, \tag{3.4}$$

where ϕ_k represents u_k or v_k.

For our specific system of equations, we apply this formula to each nonlinear function F_{ij}. This yields the following set of Adomian polynomials:

$$\begin{cases} A_n^{11} = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[F_{11}\left(\sum_{k=0}^{n}\lambda^k u_k\right)\right]_{\lambda=0}, \\ A_n^{12} = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[F_{12}\left(\sum_{k=0}^{n}\lambda^k v_k\right)\right]_{\lambda=0}, \\ A_n^{21} = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[F_{21}\left(\sum_{k=0}^{n}\lambda^k u_k\right)\right]_{\lambda=0}, \\ A_n^{22} = \frac{1}{n!}\frac{d^n}{d\lambda^n}\left[F_{22}\left(\sum_{k=0}^{n}\lambda^k v_k\right)\right]_{\lambda=0} \end{cases} \tag{3.5}$$

where $A_n^{11}, A_n^{12}, A_n^{21}$, and A_n^{22} are the Adomian polynomials for the respective nonlinear functions.

The standard Adomian decomposition method generates the recursive relation:

$$\begin{cases} u_0(x) = f_1(x), \\ v_0(x) = f_2(x), \\ u_{n+1}(x) = \int_0^x \left(k_{11}(x,t)A_n^{11} + k_{12}(x,t)A_n^{12}\right) dt, \quad n \geq 0, \\ v_{n+1}(x) = \int_0^x \left(k_{21}(x,t)A_n^{21} + k_{22}(x,t)A_n^{22}\right) dt, \quad n \geq 0. \end{cases} \tag{3.6}$$

3.1.2 First modified Adomian technique

To enhance convergence and reduce computational difficulties, we split the functions $f_1(x)$ and $f_2(x)$ into two parts:

$$\begin{cases} f_1(x) = f_{11}(x) + f_{12}(x), \\ f_2(x) = f_{21}(x) + f_{22}(x). \end{cases} \tag{3.7}$$

The modified recursive scheme becomes:

$$\begin{aligned} u_0(x) &= f_{11}(x), \\ v_0(x) &= f_{21}(x), \\ u_1(x) &= f_{12}(x) + \int_0^x \left[k_{11}(x,t)A_0^{11} + k_{12}(x,t)A_0^{12}\right] dt, \\ v_1(x) &= f_{22}(x) + \int_0^x \left[k_{21}(x,t)A_0^{21} + k_{22}(x,t)A_0^{22}\right] dt, \\ u_{n+1}(x) &= \int_0^x \left[k_{11}(x,t)A_n^{11} + k_{12}(x,t)A_n^{12}\right] dt, \quad n \geq 1, \\ v_{n+1}(x) &= \int_0^x \left[k_{21}(x,t)A_n^{21} + k_{22}(x,t)A_n^{22}\right] dt, \quad n \geq 1. \end{aligned} \tag{3.8}$$

Here, A_n^{ij} denote the Adomian polynomials corresponding to the nonlinear functions F_{ij}.

3.1.3 Second modification: Series expansion

For strongly nonlinear systems, we expand $f_1(x)$ and $f_2(x)$ into series:

$$\begin{aligned} u_0(x) &= f_{1,0}(x), \\ v_0(x) &= f_{2,0}(x), \\ u_{n+1}(x) &= f_{1,(n+1)}(x) + \int_0^x \left[k_{11}(x,t)A_n^{11} + k_{12}(x,t)A_n^{12}\right] dt, \quad n \geq 0, \\ v_{n+1}(x) &= f_{2,(n+1)}(x) + \int_0^x \left[k_{21}(x,t)A_n^{21} + k_{22}(x,t)A_n^{22}\right] dt, \quad n \geq 0. \end{aligned} \tag{3.9}$$

3.1.4 Algorithm 1: Second MADM-II

Input:

- Nonhomogeneous functions: $f_1(x)$, $f_2(x)$.
- Kernels: $k_{11}(x,t)$, $k_{12}(x,t)$, $k_{21}(x,t)$, $k_{22}(x,t)$.
- Maximum Taylor order (K) and tolerance (ε).

- Domain of interest: $[a, b]$ (used for norm evaluation).

Output:

- Approximate solutions: $u(x) \approx \sum_{k=0}^{M} u_k(x)$, $v(x) \approx \sum_{k=0}^{M} v_k(x)$.

Steps:

1. **Taylor series expansion:**

   Expand the nonhomogeneous functions using a Taylor series:

$$f_1(x) \approx \sum_{j=0}^{K} c_j x^j,$$
$$f_2(x) \approx \sum_{j=0}^{K} d_j x^j.$$

2. **Initialization:**

   - $u_0(x) = c_0$ (constant term of f_1 Taylor series)
   - $v_0(x) = d_0$ (constant term of f_2 Taylor series)
   - Initialize iteration counter: $n = 0$

3. **Iteration loop** (while $n \leq K - 1$):

   a. Compute Adomian polynomials for nonlinear operators:

$$\begin{aligned} A_n^{11} &= \text{AdomianPoly}\big(F_{11}[u_0, \ldots, u_n], n\big), \\ A_n^{12} &= \text{AdomianPoly}\big(F_{12}[v_0, \ldots, v_n], n\big), \\ A_n^{21} &= \text{AdomianPoly}\big(F_{21}[u_0, \ldots, u_n], n\big), \\ A_n^{22} &= \text{AdomianPoly}\big(F_{22}[v_0, \ldots, v_n], n\big). \end{aligned}$$

   b. Compute the next terms:

$$u_{n+1}(x) = c_{n+1} x^{n+1} + \int_0^x \left[k_{11}(x,t) A_n^{11} + k_{12}(x,t) A_n^{12}\right] dt,$$
$$v_{n+1}(x) = d_{n+1} x^{n+1} + \int_0^x \left[k_{21}(x,t) A_n^{21} + k_{22}(x,t) A_n^{22}\right] dt.$$

   c. Update partial sums:

$$s_u^{n+1}(x) = \sum_{k=0}^{n+1} u_k(x), \quad \text{(Cumulative solution for } u\text{)}$$
$$s_v^{n+1}(x) = \sum_{k=0}^{n+1} v_k(x). \quad \text{(Cumulative solution for } v\text{)}$$

   d. Termination check:

   If $\|u_{n+1}\| < \varepsilon$ and $\|v_{n+1}\| < \varepsilon$ over $[a, b]$, then stop.

   Otherwise, set $n \leftarrow n + 1$ and repeat from step 3a.

4. **Solution approximations:**

   Set $M = \min(n+1, K)$ (actual terms used), then:

$$u(x) \approx \sum_{k=0}^{M} u_k(x),$$
$$v(x) \approx \sum_{k=0}^{M} v_k(x).$$

Note:

- The Adomian polynomials A_n^{ij} are computed using standard recurrence relations [1].
- The norm $\|\cdot\|$ refers to the maximum norm $\max\limits_{x\in[a,b]} |\cdot|$ over the interval $[a, b]$.

3.2 Application of H-JM for solving a system of NVIEs-II

This work presents a systematic extension of the H-JM originally designed for single NVIEs-II to coupled systems. Such systems model interdependent phenomena in physics and engineering, where unknown functions $u(x)$ and $v(x)$ exhibit mutual nonlinear interactions. The extended H-JM leverages recursive Maclaurin series expansions to generate successive approximations. By preserving the iterative convergence properties of the original method under appropriate initial conditions, it achieves exponential decay of error for smooth kernels. Crucially, we introduce coupling-adapted recurrence schemes to handle cross-term nonlinearities ($F_{ij}(u, v)$), ensuring stability even for high-order derivatives (u_5, v_5).

As in the MADM, the general system of NLVIEs considered here is given by Eqs. (3.1) and (3.2).

$$\begin{cases} u(x) = f_1(x) + \lambda \int_0^x \left[k_{11}(x,t)F_{11}(u(t)) + k_{12}(x,t)F_{12}(v(t))\right] dt, \\ v(x) = f_2(x) + \lambda \int_0^x \left[k_{21}(x,t)F_{21}(u(t)) + k_{22}(x,t)F_{22}(v(t))\right] dt \end{cases} \tag{3.10}$$

where $u(x)$ and $v(x)$ are the unknown functions, $f_1(x)$ and $f_2(x)$ are known functions, $k_{ij}(x, t)$ for $i, j = 1, 2$ are continuous kernel functions, F_{ij} are nonlinear operators, and λ is a real parameter.

We begin by applying the Maclaurin series expansion to each equation around $x = 0$. Using the differential operator $D_x^i = \frac{d^i}{dx^i}$, this leads to:

$$\begin{cases} u(x) = \sum\limits_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \Big(f_1(x) + \lambda \int_0^x \left[k_{11}(x,t)F_{11}(u(t)) + k_{12}(x,t)F_{12}(v(t))\right] dt \Big)\Big|_{x=0}, \\ v(x) = \sum\limits_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \Big(f_2(x) + \lambda \int_0^x \left[k_{21}(x,t)F_{21}(u(t)) + k_{22}(x,t)F_{22}(v(t))\right] dt \Big)\Big|_{x=0} \end{cases} \tag{3.11}$$

Assuming that the solutions of $u(x)$ and $v(x)$ can be written as power series:

$$u(x) = \sum_{i=0}^{\infty} u_i(x), \quad v(x) = \sum_{i=0}^{\infty} v_i(x), \tag{3.12}$$

we substitute (3.12) into (3.11), then equate the coefficients of like powers of x, yielding:

$$\begin{cases} \sum_{i=0}^{\infty} u_i(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \left(f_1(x) + \lambda \int_0^x \left[k_{11}(x,t) F_{11} \left(\sum_{k=0}^{\infty} u_k(t) \right) \right.\right. \\ \qquad \left.\left. + k_{12}(x,t) F_{12} \left(\sum_{k=0}^{\infty} v_k(t) \right) \right] dt \right) \Big|_{x=0}, \\ \sum_{i=0}^{\infty} v_i(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \left(f_2(x) + \lambda \int_0^x \left[k_{21}(x,t) F_{21} \left(\sum_{k=0}^{\infty} u_k(t) \right) \right.\right. \\ \qquad \left.\left. + k_{22}(x,t) F_{22} \left(\sum_{k=0}^{\infty} v_k(t) \right) \right] dt \right) \Big|_{x=0} \end{cases} \tag{3.13}$$

Isolating the $i = 0$ terms from the series on the left-hand side of Equation (3.13) gives:

$$\begin{cases} u_0(x) + \sum_{i=0}^{\infty} u_{i+1}(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \left(f_1(x) + \lambda \int_0^x \left[k_{11}(x,t) F_{11} \left(\sum_{k=0}^{\infty} u_k(t) \right) \right.\right. \\ \qquad \left.\left. + k_{12}(x,t) F_{12} \left(\sum_{k=0}^{\infty} v_k(t) \right) \right] dt \right) \Big|_{x=0}, \\ v_0(x) + \sum_{i=0}^{\infty} v_{i+1}(x) = \sum_{i=0}^{\infty} \frac{x^i}{i!} D_x^i \left(f_2(x) + \lambda \int_0^x \left[k_{21}(x,t) F_{21} \left(\sum_{k=0}^{\infty} u_k(t) \right) \right.\right. \\ \qquad \left.\left. + k_{22}(x,t) F_{22} \left(\sum_{k=0}^{\infty} v_k(t) \right) \right] dt \right) \Big|_{x=0} \end{cases} \tag{3.14}$$

Equating terms of the same powers for both sides, we obtain the recursive formulas. The zeroth-order approximations are obtained as:

$$u_0(x) = u(0) = f_1(0),$$

$$v_0(x) = v(0) = f_2(0).$$

The first-order approximations are:

$$u_1(x) = \frac{x}{1!} D_x \left(f_1(x) + \lambda \int_0^x \left[k_{11}(x,t) F_{11}(u_0(t)) + k_{12}(x,t) F_{12}(v_0(t)) \right] dt \right)_{x=0},$$
$$v_1(x) = \frac{x}{1!} D_x \left(f_2(x) + \lambda \int_0^x \left[k_{21}(x,t) F_{21}(u_0(t)) + k_{22}(x,t) F_{22}(v_0(t)) \right] dt \right)_{x=0}.$$

The second-order approximations are given by:

$$\begin{aligned} u_2(x) = \frac{x^2}{2!} D_x^2 \Big(f_1(x) + \lambda \int_0^x \big[& k_{11}(x,t) F_{11}(u_0(t) + u_1(t)) \\ & + k_{12}(x,t) F_{12}(v_0(t) + v_1(t)) \big] dt \Big)_{x=0}, \\ v_2(x) = \frac{x^2}{2!} D_x^2 \Big(f_2(x) + \lambda \int_0^x \big[& k_{21}(x,t) F_{21}(u_0(t) + u_1(t)) \\ & + k_{22}(x,t) F_{22}(v_0(t) + v_1(t)) \big] dt \Big)_{x=0}. \end{aligned}$$

The general recurrence relations for higher-order terms are:

$$\begin{cases} u_{i+1}(x) = \dfrac{x^{i+1}}{(i+1)!} D_x^{i+1} \left(f_1(x) + \lambda \displaystyle\int_0^x \big[k_{11}(x,t) F_{11}\big(\sum_{j=0}^{i} u_j(t) \big) \right. \\ \qquad \left. + k_{12}(x,t) F_{12}\big(\displaystyle\sum_{j=0}^{i} v_j(t) \big) \big] dt \right)_{x=0}, \\ v_{i+1}(x) = \dfrac{x^{i+1}}{(i+1)!} D_x^{i+1} \left(f_2(x) + \lambda \displaystyle\int_0^x \big[k_{21}(x,t) F_{21}\big(\sum_{j=0}^{i} u_j(t) \big) \right. \\ \qquad \left. + k_{22}(x,t) F_{22}\big(\displaystyle\sum_{j=0}^{i} v_j(t) \big) \big] dt \right)_{x=0} \end{cases} \tag{3.15}$$

for $i \geq 0$.

After computing a finite number n of terms from the recursive relations above, the approximate solutions to the system are given by:

$$\begin{cases} u(x) = \displaystyle\sum_{i=0}^{n} u_i(x), \\ v(x) = \displaystyle\sum_{i=0}^{n} v_i(x) \end{cases}$$

where n is chosen according to the desired level of accuracy.

3.2.1 Algorithm 2: Hussein–Jassim iterative method for coupled NVIEs–II

Input:

- Nonhomogeneous functions: $f_1(x)$, $f_2(x)$.
- Kernels: $k_{11}(x,t)$, $k_{12}(x,t)$, $k_{21}(x,t)$, $k_{22}(x,t)$.
- Nonlinear operators: F_{11}, F_{12}, F_{21}, F_{22}.
- Parameter: λ.
- Maximum iterations ($N_{\max}$) – upper iteration limit.
- Tolerance (ε) – convergence threshold.
- Norm domain $[a,b]$ – evaluation interval.

Output:

- Approximate solutions:

$$u(x) \approx \sum_{i=0}^{M} u_i(x), \quad v(x) \approx \sum_{i=0}^{M} v_i(x).$$

- Computed iterations (M) – actual iterations used.

Steps:

1. **Initialization:**

$$\begin{aligned}
u_0(x) &= f_1(0), \quad \text{(initial condition for } u\text{)} \\
v_0(x) &= f_2(0), \quad \text{(initial condition for } v\text{)} \\
S_u^{(0)}(x) &= u_0(x), \quad \text{(cumulative solution for } u\text{)} \\
S_v^{(0)}(x) &= v_0(x), \quad \text{(cumulative solution for } v\text{)} \\
i &= 0, \\
\text{converged} &= \text{false}. \quad \text{(convergence flag)}
\end{aligned}$$

2. **Iterative computation** (while $i < N_{\max}$ and not converged):

   a. **Partial solutions:**

$$S_u^{(i)}(t) = \sum_{j=0}^{i} u_j(t), \quad \text{(cumulative } u \text{ up to } i\text{)}$$

$$S_v^{(i)}(t) = \sum_{j=0}^{i} v_j(t). \quad \text{(cumulative } v \text{ up to } i\text{)}$$

   b. **Integral expressions:**

$$G_1^{(i)}(x) = f_1(x) + \lambda \int_0^x \left[k_{11}(x,t)F_{11}\big(S_u^{(i)}(t)\big) + k_{12}(x,t)F_{12}\big(S_v^{(i)}(t)\big)\right] dt,$$

$$G_2^{(i)}(x) = f_2(x) + \lambda \int_0^x \left[k_{21}(x,t)F_{21}\big(S_u^{(i)}(t)\big) + k_{22}(x,t)F_{22}\big(S_v^{(i)}(t)\big)\right] dt.$$

   c. **Next term calculation:**

$$\begin{aligned}
u_{i+1}(x) &= \frac{x^{i+1}}{(i+1)!} D_x^{i+1} G_1^{(i)}(x)\bigg|_{x=0}, \\
v_{i+1}(x) &= \frac{x^{i+1}}{(i+1)!} D_x^{i+1} G_2^{(i)}(x)\bigg|_{x=0}.
\end{aligned}$$

   d. **Solution update:**

$$\begin{aligned}
S_u^{(i+1)}(x) &= S_u^{(i)}(x) + u_{i+1}(x), \\
S_v^{(i+1)}(x) &= S_v^{(i)}(x) + v_{i+1}(x).
\end{aligned}$$

   e. **Convergence check:**

$$\begin{aligned}
\delta_u &= \big\|S_u^{(i+1)} - S_u^{(i)}\big\|_{\infty,[a,b]} = \|u_{i+1}\|_{\infty,[a,b]}, \\
\delta_v &= \big\|S_v^{(i+1)} - S_v^{(i)}\big\|_{\infty,[a,b]} = \|v_{i+1}\|_{\infty,[a,b]}.
\end{aligned}$$

   If $\delta_u < \varepsilon$ **and** $\delta_v < \varepsilon$:

   converged $\leftarrow$ true

   Else:

   $i \leftarrow i + 1$

3. **Solution assembly:**

$$M = i + 1, \quad \text{(final iteration count)}$$
$$u(x) \approx \sum_{k=0}^{M} u_k(x),$$
$$v(x) \approx \sum_{k=0}^{M} v_k(x).$$

Note:

- The norm $\|\cdot\|_{\infty,[a,b]}$ denotes the maximum norm: $\max_{x\in[a,b]} |\cdot|$.
- Convergence is achieved when the incremental terms u_{i+1} and v_{i+1} become sufficiently small.

3.3 Application of the CNPSFM for solving systems of NVIEs-II

Classical cubic splines often fail to capture the oscillatory or rapidly varying behaviors inherent in coupled nonlinear Volterra systems. To address this, the CNPSFM enriches the standard cubic basis with trigonometric/hyperbolic functions and a tunable frequency parameter, k, yielding C^2-continuous approximations with enhanced adaptability.

In this work, we formulate a rigorous CNPSFM framework for systems of NVIEs-II and construct a fifth-derivative matching scheme that leverages boundary derivatives generated via repeated differentiation of the Volterra operators. We also incorporate an adaptive selection of k through residual minimization and design a coupling-aware block discretization for the nonlinear kernels.

Under standard smoothness and Lipschitz assumptions, we establish the convergence and stability of the method and demonstrate its accuracy on benchmarks exhibiting both oscillatory and fast-growing solutions.

As in the previous methods, the general system of NLVIEs considered here is given by Eqs. (3.1) and (3.2), which are reformulated in this section for the purpose of applying the CNPSFM scheme.

$$u(x) = f_1(x) + \int_0^x \left[k_{11}(x,t)F_{11}(u(t)) + k_{12}(x,t)F_{12}(v(t))\right] dt, \tag{3.16}$$
$$v(x) = f_2(x) + \int_0^x \left[k_{21}(x,t)F_{21}(u(t)) + k_{22}(x,t)F_{22}(v(t))\right] dt. \tag{3.17}$$

In this formulation, $u(x)$ and $v(x)$ denote the unknown functions to be determined, while $f_1(x)$ and $f_2(x)$ are known continuous functions. The terms $k_{ij}(x,t)$, $i,j = 1,2$, represent the kernel functions associated with the integral operators. The nonlinear behavior of the system is captured through the functions F_{ij}, which may take different forms depending on the specific application or model under investigation. This general representation enables the modeling of a wide range of nonlinear interactions and coupling effects within the same framework.

The interval $[a,b]$ is divided into n subintervals with equal width $h = (b-a)/n$, and nodes $x_i = a + ih$, where $i = 0,1,\dots,n$. On each subinterval $[x_i, x_{i+1}]$, the functions $u(x)$ and $v(x)$

are approximated by the following cubic non-polynomial spline functions:

$$\varphi_{1i}(x) = a_{1i}\cos\big(k(x-x_i)\big) + b_{1i}\sin\big(k(x-x_i)\big) + c_{1i}(x-x_i) + d_{1i}(x-x_i)^2 + e_{1i}(x-x_i)^3 + g_{1i}, \tag{3.18}$$

$$\varphi_{2i}(x) = a_{2i}\cos\big(k(x-x_i)\big) + b_{2i}\sin\big(k(x-x_i)\big) + c_{2i}(x-x_i) + d_{2i}(x-x_i)^2 + e_{2i}(x-x_i)^3 + g_{2i}. \tag{3.19}$$

Here, k is a frequency parameter chosen adaptively to enhance convergence and capture oscillatory behavior.

To determine the unknown coefficients $a_{1i}, b_{1i}, c_{1i}, d_{1i}, e_{1i}, g_{1i}$ and $a_{2i}, b_{2i}, c_{2i}, d_{2i}, e_{2i}, g_{2i}$ of the spline functions used to approximate $u(x)$ and $v(x)$, we compute the first through fifth derivatives of the expressions in Eqs. (3.18) and (3.19), resulting in the following relations:

$$\begin{cases} \varphi'_{1i}(x) = -ka_{1i}\sin\big(k(x-x_i)\big) + kb_{1i}\cos\big(k(x-x_i)\big) \\ \qquad + c_{1i} + 2d_{1i}(x-x_i) + 3e_{1i}(x-x_i)^2, \\ \varphi''_{1i}(x) = -k^2a_{1i}\cos\big(k(x-x_i)\big) - k^2b_{1i}\sin\big(k(x-x_i)\big) \\ \qquad + 2d_{1i} + 6e_{1i}(x-x_i), \\ \varphi^{(3)}_{1i}(x) = k^3a_{1i}\sin\big(k(x-x_i)\big) - k^3b_{1i}\cos\big(k(x-x_i)\big) + 6e_{1i}, \\ \varphi^{(4)}_{1i}(x) = k^4a_{1i}\cos\big(k(x-x_i)\big) + k^4b_{1i}\sin\big(k(x-x_i)\big), \\ \varphi^{(5)}_{1i}(x) = -k^5a_{1i}\sin\big(k(x-x_i)\big) + k^5b_{1i}\cos\big(k(x-x_i)\big), \\ \varphi'_{2i}(x) = -ka_{2i}\sin\big(k(x-x_i)\big) + kb_{2i}\cos\big(k(x-x_i)\big) \\ \qquad + c_{2i} + 2d_{2i}(x-x_i) + 3e_{2i}(x-x_i)^2, \\ \varphi''_{2i}(x) = -k^2a_{2i}\cos\big(k(x-x_i)\big) - k^2b_{2i}\sin\big(k(x-x_i)\big) \\ \qquad + 2d_{2i} + 6e_{2i}(x-x_i), \\ \varphi^{(3)}_{2i}(x) = k^3a_{2i}\sin\big(k(x-x_i)\big) - k^3b_{2i}\cos\big(k(x-x_i)\big) + 6e_{2i}, \\ \varphi^{(4)}_{2i}(x) = k^4a_{2i}\cos\big(k(x-x_i)\big) + k^4b_{2i}\sin\big(k(x-x_i)\big), \\ \varphi^{(5)}_{2i}(x) = -k^5a_{2i}\sin\big(k(x-x_i)\big) + k^5b_{2i}\cos\big(k(x-x_i)\big). \end{cases} \tag{3.20}$$

By substituting $x = x_i$ into the above equations (Eqs. (3.18)–(3.20)), we obtain simplified expressions for the spline values and their derivatives at the nodes, which will be used to construct the numerical scheme:

$$\left\{\begin{aligned} \varphi_{1i}(x_i) &= a_{1i} + g_{1i}, \\ \varphi'_{1i}(x_i) &= kb_{1i} + c_{1i}, \\ \varphi''_{1i}(x_i) &= -k^2a_{1i} + 2d_{1i}, \\ \varphi^{(3)}_{1i}(x_i) &= -k^3b_{1i} + 6e_{1i}, \\ \varphi^{(4)}_{1i}(x_i) &= k^4a_{1i}, \\ \varphi^{(5)}_{1i}(x_i) &= k^5b_{1i} \end{aligned}\right\} \qquad \left\{\begin{aligned} \varphi_{2i}(x_i) &= a_{2i} + g_{2i}, \\ \varphi'_{2i}(x_i) &= kb_{2i} + c_{2i}, \\ \varphi''_{2i}(x_i) &= -k^2a_{2i} + 2d_{2i}, \\ \varphi^{(3)}_{2i}(x_i) &= -k^3b_{2i} + 6e_{2i}, \\ \varphi^{(4)}_{2i}(x_i) &= k^4a_{2i}, \\ \varphi^{(5)}_{2i}(x_i) &= k^5b_{2i}. \end{aligned}\right.$$

And we get the values of $a_{1i}, b_{1i}, c_{1i}, d_{1i}, e_{1i}$, and g_{1i} as follows:

$$a_{1i} = \frac{1}{k^4}\varphi_{1i}^{(4)}(x_i), \tag{3.21}$$

$$b_{1i} = \frac{1}{k^5}\varphi_{1i}^{(5)}(x_i), \tag{3.22}$$

$$c_{1i} = \varphi_{1i}'(x_i) - kb_{1i}, \tag{3.23}$$

$$d_{1i} = \frac{1}{2}\left(\varphi_{1i}''(x_i) + k^2 a_{1i}\right), \tag{3.24}$$

$$e_{1i} = \frac{1}{6}\left(\varphi_{1i}^{(3)}(x_i) + k^3 b_{1i}\right), \tag{3.25}$$

$$g_{1i} = \varphi_{1i}(x_i) - a_{1i}. \tag{3.26}$$

Similarly, we can get the values of $a_{2i}, b_{2i}, c_{2i}, d_{2i}, e_{2i}$, and g_{2i} as follows:

$$a_{2i} = \frac{1}{k^4}\varphi_{2i}^{(4)}(x_i), \tag{3.27}$$

$$b_{2i} = \frac{1}{k^5}\varphi_{2i}^{(5)}(x_i), \tag{3.28}$$

$$c_{2i} = \varphi_{2i}'(x_i) - kb_{2i}, \tag{3.29}$$

$$d_{2i} = \frac{1}{2}\left(\varphi_{2i}''(x_i) + k^2 a_{2i}\right), \tag{3.30}$$

$$e_{2i} = \frac{1}{6}\left(\varphi_{2i}^{(3)}(x_i) + k^3 b_{2i}\right), \tag{3.31}$$

$$g_{2i} = \varphi_{2i}(x_i) - a_{2i}. \tag{3.32}$$

3.3.1 Derivative matching conditions

Having established the explicit coefficient formulas, we now determine the nodal derivative values $\varphi_{1i}^{(k)}(x_i)$ and $\varphi_{2i}^{(k)}(x_i)$ $(k = 0, 1, \ldots, 5)$ through systematic derivative matching. This process involves differentiating the system of integral equations (3.16)–(3.17) up to fifth order and evaluating at each node x_i. The derivation is extensive but follows a systematic pattern. We present the complete derivation for transparency and reproducibility, organized by derivative order.

Derivative matching for $u(x)$ **Zeroth derivative (Initial condition):**

$$u_0 = u(a) = f_1(a). \tag{3.33}$$

First derivative: Differentiating equation (3.16) once and applying Leibniz's rule,

$$\begin{aligned} u'(x) &= f_1'(x) + k_{11}(x,x)F_{11}(u(x)) + k_{12}(x,x)F_{12}(v(x)) \\ &\quad + \int_0^x \left[\frac{\partial k_{11}(x,t)}{\partial x}F_{11}(u(t)) + \frac{\partial k_{12}(x,t)}{\partial x}F_{12}(v(t))\right] dt. \end{aligned}$$

Evaluating at $x = a$ (where the integral vanishes):

$$u_0' = f_1'(a) + k_{11}(a,a)F_{11}(u_0) + k_{12}(a,a)F_{12}(v_0). \tag{3.34}$$

Second derivative: Differentiating again and applying the product rule to $k_{ij}(x,x)F_{ij}$:

$$\begin{aligned}u''(x) &= f_1''(x) + \frac{d}{dx}\left[k_{11}(x,x)F_{11}(u(x))\right] + \frac{d}{dx}\left[k_{12}(x,x)F_{12}(v(x))\right] \\ &+ \frac{d}{dx}\left[\int_0^x \frac{\partial k_{11}(x,t)}{\partial x} F_{11}(u(t))\,dt\right] \\ &+ \frac{d}{dx}\left[\int_0^x \frac{\partial k_{12}(x,t)}{\partial x} F_{12}(v(t))\,dt\right].\end{aligned}$$

For the first term involving $k_{11}(x,x)F_{11}(u(x))$:

$$\frac{d}{dx}\left[k_{11}(x,x)F_{11}(u(x))\right] = \left[\frac{\partial k_{11}}{\partial x} + \frac{\partial k_{11}}{\partial t}\right]_{t=x} F_{11}(u(x)) + k_{11}(x,x)F_{11}'(u(x))u'(x).$$

Evaluating at $x = a$:

$$\begin{aligned}u_0'' = f_1''(a) &+ \left[\frac{\partial k_{11}}{\partial x} + \frac{\partial k_{11}}{\partial t}\right]_{(a,a)} F_{11}(u_0) + k_{11}(a,a)F_{11}'(u_0)u_0' \\ &+ \left[\frac{\partial k_{12}}{\partial x} + \frac{\partial k_{12}}{\partial t}\right]_{(a,a)} F_{12}(v_0) + k_{12}(a,a)F_{12}'(v_0)v_0'.\end{aligned} \tag{3.35}$$

Third derivative: Continuing the differentiation process:

$$\begin{aligned}u_0^{(3)} = f_1^{(3)}(a) &+ \left[\frac{\partial^2 k_{11}}{\partial x^2} + 2\frac{\partial^2 k_{11}}{\partial x \partial t} + \frac{\partial^2 k_{11}}{\partial t^2}\right]_{(a,a)} F_{11}(u_0) \\ &+ 2\left[\frac{\partial k_{11}}{\partial x} + \frac{\partial k_{11}}{\partial t}\right]_{(a,a)} F_{11}'(u_0)u_0' \\ &+ k_{11}(a,a)\left[F_{11}''(u_0)(u_0')^2 + F_{11}'(u_0)u_0''\right] \\ &+ \left[\frac{\partial^2 k_{12}}{\partial x^2} + 2\frac{\partial^2 k_{12}}{\partial x \partial t} + \frac{\partial^2 k_{12}}{\partial t^2}\right]_{(a,a)} F_{12}(v_0) \\ &+ 2\left[\frac{\partial k_{12}}{\partial x} + \frac{\partial k_{12}}{\partial t}\right]_{(a,a)} F_{12}'(v_0)v_0' \\ &+ k_{12}(a,a)\left[F_{12}''(v_0)(v_0')^2 + F_{12}'(v_0)v_0''\right].\end{aligned} \tag{3.36}$$

Fourth derivative:

$$
\begin{aligned}
u_0^{(4)} = f_1^{(4)}(a) &+ \left[\frac{\partial^3 k_{11}}{\partial x^3} + 3\frac{\partial^3 k_{11}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{11}}{\partial x \partial t^2} + \frac{\partial^3 k_{11}}{\partial t^3}\right]_{(a,a)} F_{11}(u_0) \\
&+ 3\left[\frac{\partial^2 k_{11}}{\partial x^2} + 2\frac{\partial^2 k_{11}}{\partial x \partial t} + \frac{\partial^2 k_{11}}{\partial t^2}\right]_{(a,a)} F'_{11}(u_0)u'_0 \\
&+ 3\left[\frac{\partial k_{11}}{\partial x} + \frac{\partial k_{11}}{\partial t}\right]_{(a,a)} \left[F''_{11}(u_0)(u'_0)^2 + F'_{11}(u_0)u''_0\right] \\
&+ k_{11}(a,a)\left[F_{11}^{(3)}(u_0)(u'_0)^3 + 3F''_{11}(u_0)u'_0u''_0 + F'_{11}(u_0)u_0^{(3)}\right] \\
&+ \left[\frac{\partial^3 k_{12}}{\partial x^3} + 3\frac{\partial^3 k_{12}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{12}}{\partial x \partial t^2} + \frac{\partial^3 k_{12}}{\partial t^3}\right]_{(a,a)} F_{12}(v_0) \\
&+ 3\left[\frac{\partial^2 k_{12}}{\partial x^2} + 2\frac{\partial^2 k_{12}}{\partial x \partial t} + \frac{\partial^2 k_{12}}{\partial t^2}\right]_{(a,a)} F'_{12}(v_0)v'_0 \\
&+ 3\left[\frac{\partial k_{12}}{\partial x} + \frac{\partial k_{12}}{\partial t}\right]_{(a,a)} \left[F''_{12}(v_0)(v'_0)^2 + F'_{12}(v_0)v''_0\right] \\
&+ k_{12}(a,a)\left[F_{12}^{(3)}(v_0)(v'_0)^3 + 3F''_{12}(v_0)v'_0v''_0 + F'_{12}(v_0)v_0^{(3)}\right].
\end{aligned} \tag{3.37}
$$

Fifth derivative:

$$
\begin{aligned}
u_0^{(5)} = f_1^{(5)}(a)& \\
&+ \left[\frac{\partial^4 k_{11}}{\partial x^4} + 4\frac{\partial^4 k_{11}}{\partial x^3 \partial t} + 6\frac{\partial^4 k_{11}}{\partial x^2 \partial t^2} + 4\frac{\partial^4 k_{11}}{\partial x \partial t^3} + \frac{\partial^4 k_{11}}{\partial t^4}\right]_{(a,a)} F_{11}(u_0) \\
&+ 4\left[\frac{\partial^3 k_{11}}{\partial x^3} + 3\frac{\partial^3 k_{11}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{11}}{\partial x \partial t^2} + \frac{\partial^3 k_{11}}{\partial t^3}\right]_{(a,a)} F'_{11}(u_0)u'_0 \\
&+ 6\left[\frac{\partial^2 k_{11}}{\partial x^2} + 2\frac{\partial^2 k_{11}}{\partial x \partial t} + \frac{\partial^2 k_{11}}{\partial t^2}\right]_{(a,a)} \left[F''_{11}(u_0)(u'_0)^2 + F'_{11}(u_0)u''_0\right] \\
&+ 4\left[\frac{\partial k_{11}}{\partial x} + \frac{\partial k_{11}}{\partial t}\right]_{(a,a)} \left[F_{11}^{(3)}(u_0)(u'_0)^3 + 3F''_{11}(u_0)u'_0u''_0 + F'_{11}(u_0)u_0^{(3)}\right] \\
&+ k_{11}(a,a)\big[F_{11}^{(4)}(u_0)(u'_0)^4 + 6F_{11}^{(3)}(u_0)(u'_0)^2u''_0 + 4F_{11}^{(3)}(u_0)u'_0u_0^{(3)} \\
&\qquad + 3F''_{11}(u_0)(u''_0)^2 + F'_{11}(u_0)u_0^{(4)}\big] \\
&+ \left[\frac{\partial^4 k_{12}}{\partial x^4} + 4\frac{\partial^4 k_{12}}{\partial x^3 \partial t} + 6\frac{\partial^4 k_{12}}{\partial x^2 \partial t^2} + 4\frac{\partial^4 k_{12}}{\partial x \partial t^3} + \frac{\partial^4 k_{12}}{\partial t^4}\right]_{(a,a)} F_{12}(v_0) \\
&+ 4\left[\frac{\partial^3 k_{12}}{\partial x^3} + 3\frac{\partial^3 k_{12}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{12}}{\partial x \partial t^2} + \frac{\partial^3 k_{12}}{\partial t^3}\right]_{(a,a)} F'_{12}(v_0)v'_0 \\
&+ 6\left[\frac{\partial^2 k_{12}}{\partial x^2} + 2\frac{\partial^2 k_{12}}{\partial x \partial t} + \frac{\partial^2 k_{12}}{\partial t^2}\right]_{(a,a)} \left[F''_{12}(v_0)(v'_0)^2 + F'_{12}(v_0)v''_0\right] \\
&+ 4\left[\frac{\partial k_{12}}{\partial x} + \frac{\partial k_{12}}{\partial t}\right]_{(a,a)} \left[F_{12}^{(3)}(v_0)(v'_0)^3 + 3F''_{12}(v_0)v'_0v''_0 + F'_{12}(v_0)v_0^{(3)}\right] \\
&+ k_{12}(a,a)\big[F_{12}^{(4)}(v_0)(v'_0)^4 + 6F_{12}^{(3)}(v_0)(v'_0)^2v''_0 + 4F_{12}^{(3)}(v_0)v'_0v_0^{(3)} \\
&\qquad + 3F''_{12}(v_0)(v''_0)^2 + F'_{12}(v_0)v_0^{(4)}\big].
\end{aligned} \tag{3.38}
$$

Derivative matching for $v(x)$ Similarly, for obtaining the numerical approximation of equation (3.17), we differentiate it five times with respect to x and substitute $x = a$:

Zeroth derivative (Initial condition):

$$v_0 = v(a) = f_2(a). \tag{3.39}$$

First derivative:

$$v_0' = f_2'(a) + k_{21}(a,a)F_{21}(u_0) + k_{22}(a,a)F_{22}(v_0). \tag{3.40}$$

Second derivative:

$$\begin{aligned} v_0'' = f_2''(a) &+ \left[\frac{\partial k_{21}}{\partial x} + \frac{\partial k_{21}}{\partial t}\right]_{(a,a)} F_{21}(u_0) + k_{21}(a,a)F_{21}'(u_0)u_0' \\ &+ \left[\frac{\partial k_{22}}{\partial x} + \frac{\partial k_{22}}{\partial t}\right]_{(a,a)} F_{22}(v_0) + k_{22}(a,a)F_{22}'(v_0)v_0'. \end{aligned} \tag{3.41}$$

Third derivative:

$$\begin{aligned} v_0^{(3)} = f_2^{(3)}(a) &+ \left[\frac{\partial^2 k_{21}}{\partial x^2} + 2\frac{\partial^2 k_{21}}{\partial x \partial t} + \frac{\partial^2 k_{21}}{\partial t^2}\right]_{(a,a)} F_{21}(u_0) \\ &+ 2\left[\frac{\partial k_{21}}{\partial x} + \frac{\partial k_{21}}{\partial t}\right]_{(a,a)} F_{21}'(u_0)u_0' \\ &+ k_{21}(a,a)\left[F_{21}''(u_0)(u_0')^2 + F_{21}'(u_0)u_0''\right] \\ &+ \left[\frac{\partial^2 k_{22}}{\partial x^2} + 2\frac{\partial^2 k_{22}}{\partial x \partial t} + \frac{\partial^2 k_{22}}{\partial t^2}\right]_{(a,a)} F_{22}(v_0) \\ &+ 2\left[\frac{\partial k_{22}}{\partial x} + \frac{\partial k_{22}}{\partial t}\right]_{(a,a)} F_{22}'(v_0)v_0' \\ &+ k_{22}(a,a)\left[F_{22}''(v_0)(v_0')^2 + F_{22}'(v_0)v_0''\right]. \end{aligned} \tag{3.42}$$

Fourth derivative:

$$\begin{aligned} v_0^{(4)} = f_2^{(4)}(a) &+ \left[\frac{\partial^3 k_{21}}{\partial x^3} + 3\frac{\partial^3 k_{21}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{21}}{\partial x \partial t^2} + \frac{\partial^3 k_{21}}{\partial t^3}\right]_{(a,a)} F_{21}(u_0) \\ &+ 3\left[\frac{\partial^2 k_{21}}{\partial x^2} + 2\frac{\partial^2 k_{21}}{\partial x \partial t} + \frac{\partial^2 k_{21}}{\partial t^2}\right]_{(a,a)} F_{21}'(u_0)u_0' \\ &+ 3\left[\frac{\partial k_{21}}{\partial x} + \frac{\partial k_{21}}{\partial t}\right]_{(a,a)} \left[F_{21}''(u_0)(u_0')^2 + F_{21}'(u_0)u_0''\right] \\ &+ k_{21}(a,a)\left[F_{21}^{(3)}(u_0)(u_0')^3 + 3F_{21}''(u_0)u_0'u_0'' + F_{21}'(u_0)u_0^{(3)}\right] \\ &+ \left[\frac{\partial^3 k_{22}}{\partial x^3} + 3\frac{\partial^3 k_{22}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{22}}{\partial x \partial t^2} + \frac{\partial^3 k_{22}}{\partial t^3}\right]_{(a,a)} F_{22}(v_0) \\ &+ 3\left[\frac{\partial^2 k_{22}}{\partial x^2} + 2\frac{\partial^2 k_{22}}{\partial x \partial t} + \frac{\partial^2 k_{22}}{\partial t^2}\right]_{(a,a)} F_{22}'(v_0)v_0' \\ &+ 3\left[\frac{\partial k_{22}}{\partial x} + \frac{\partial k_{22}}{\partial t}\right]_{(a,a)} \left[F_{22}''(v_0)(v_0')^2 + F_{22}'(v_0)v_0''\right] \\ &+ k_{22}(a,a)\left[F_{22}^{(3)}(v_0)(v_0')^3 + 3F_{22}''(v_0)v_0'v_0'' + F_{22}'(v_0)v_0^{(3)}\right]. \end{aligned} \tag{3.43}$$

Fifth derivative:

$$
\begin{aligned}
v_0^{(5)} &= f_2^{(5)}(a) \\
&+ \left[\frac{\partial^4 k_{21}}{\partial x^4} + 4\frac{\partial^4 k_{21}}{\partial x^3 \partial t} + 6\frac{\partial^4 k_{21}}{\partial x^2 \partial t^2} + 4\frac{\partial^4 k_{21}}{\partial x \partial t^3} + \frac{\partial^4 k_{21}}{\partial t^4}\right]_{(a,a)} F_{21}(u_0) \\
&+ 4\left[\frac{\partial^3 k_{21}}{\partial x^3} + 3\frac{\partial^3 k_{21}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{21}}{\partial x \partial t^2} + \frac{\partial^3 k_{21}}{\partial t^3}\right]_{(a,a)} F'_{21}(u_0)u'_0 \\
&+ 6\left[\frac{\partial^2 k_{21}}{\partial x^2} + 2\frac{\partial^2 k_{21}}{\partial x \partial t} + \frac{\partial^2 k_{21}}{\partial t^2}\right]_{(a,a)} \left[F''_{21}(u_0)(u'_0)^2 + F'_{21}(u_0)u''_0\right] \\
&+ 4\left[\frac{\partial k_{21}}{\partial x} + \frac{\partial k_{21}}{\partial t}\right]_{(a,a)} \left[F_{21}^{(3)}(u_0)(u'_0)^3 + 3F''_{21}(u_0)u'_0u''_0 + F'_{21}(u_0)u_0^{(3)}\right] \\
&+ k_{21}(a,a)\left[F_{21}^{(4)}(u_0)(u'_0)^4 + 6F_{21}^{(3)}(u_0)(u'_0)^2u''_0 + 4F_{21}^{(3)}(u_0)u'_0u_0^{(3)}\right. \\
&\qquad \left. + 3F''_{21}(u_0)(u''_0)^2 + F'_{21}(u_0)u_0^{(4)}\right] \\
&+ \left[\frac{\partial^4 k_{22}}{\partial x^4} + 4\frac{\partial^4 k_{22}}{\partial x^3 \partial t} + 6\frac{\partial^4 k_{22}}{\partial x^2 \partial t^2} + 4\frac{\partial^4 k_{22}}{\partial x \partial t^3} + \frac{\partial^4 k_{22}}{\partial t^4}\right]_{(a,a)} F_{22}(v_0) \\
&+ 4\left[\frac{\partial^3 k_{22}}{\partial x^3} + 3\frac{\partial^3 k_{22}}{\partial x^2 \partial t} + 3\frac{\partial^3 k_{22}}{\partial x \partial t^2} + \frac{\partial^3 k_{22}}{\partial t^3}\right]_{(a,a)} F'_{22}(v_0)v'_0 \\
&+ 6\left[\frac{\partial^2 k_{22}}{\partial x^2} + 2\frac{\partial^2 k_{22}}{\partial x \partial t} + \frac{\partial^2 k_{22}}{\partial t^2}\right]_{(a,a)} \left[F''_{22}(v_0)(v'_0)^2 + F'_{22}(v_0)v''_0\right] \\
&+ 4\left[\frac{\partial k_{22}}{\partial x} + \frac{\partial k_{22}}{\partial t}\right]_{(a,a)} \left[F_{22}^{(3)}(v_0)(v'_0)^3 + 3F''_{22}(v_0)v'_0v''_0 + F'_{22}(v_0)v_0^{(3)}\right] \\
&+ k_{22}(a,a)\left[F_{22}^{(4)}(v_0)(v'_0)^4 + 6F_{22}^{(3)}(v_0)(v'_0)^2v''_0 + 4F_{22}^{(3)}(v_0)v'_0v_0^{(3)}\right. \\
&\qquad \left. + 3F''_{22}(v_0)(v''_0)^2 + F'_{22}(v_0)v_0^{(4)}\right].
\end{aligned}
\tag{3.44}
$$

3.3.2 Algorithm for solving a system of NLVIEs using CNPSF

The following algorithm summarizes the complete procedure for solving the system of NLVIEs (3.16)–(3.17) using the CNPSFM.

Step 1: Set

$$h = \frac{b-a}{n}, \quad x_i = x_0 + ih, \quad i = 0, 1, 2, \ldots, n, \quad x_0 = a, \quad x_n = b.$$

Step 2:

1. Evaluate $a_{10}, b_{10}, c_{10}, d_{10}, e_{10}$, and g_{10} by substituting equations (3.33)–(3.38) into equations (3.21)–(3.26).
2. Evaluate $a_{20}, b_{20}, c_{20}, d_{20}, e_{20}$, and g_{20} by substituting equations (3.39)–(3.44) into equations (3.27)–(3.32).

Step 3:

1. Calculate $\varphi_{10}(x)$ using Step 2(1) and equation (3.18) for $i = 0$.
2. Calculate $\varphi_{20}(x)$ using Step 2(2) and equation (3.19) for $i = 0$.

Step 4: Approximate $u_1 \approx \varphi_{10}(x_1)$ and $v_1 \approx \varphi_{20}(x_1)$.

Step 5: For $i = 1$ to $n-1$, do the following steps:

Step 6:

1. Evaluate $a_{1i}, b_{1i}, c_{1i}, d_{1i}, e_{1i}$, and g_{1i} using equations (3.21)–(3.26) and replacing

$$u(x_i),\ u'(x_i),\ u''(x_i),\ u^{(3)}(x_i),\ u^{(4)}(x_i),\ u^{(5)}(x_i),$$

by

$$\varphi_{1i}(x_i),\ \varphi_{1i}'(x_i),\ \varphi_{1i}''(x_i),\ \varphi_{1i}^{(3)}(x_i),\ \varphi_{1i}^{(4)}(x_i),\ \varphi_{1i}^{(5)}(x_i).$$

2. Evaluate $a_{2i}, b_{2i}, c_{2i}, d_{2i}, e_{2i}$, and g_{2i} using equations (3.27)–(3.32) and replacing

$$v(x_i),\ v'(x_i),\ v''(x_i),\ v^{(3)}(x_i),\ v^{(4)}(x_i),\ v^{(5)}(x_i),$$

by

$$\varphi_{2i}(x_i),\ \varphi_{2i}'(x_i),\ \varphi_{2i}''(x_i),\ \varphi_{2i}^{(3)}(x_i),\ \varphi_{2i}^{(4)}(x_i),\ \varphi_{2i}^{(5)}(x_i).$$

Step 7: Calculate $\varphi_{1i}(x)$ and $\varphi_{2i}(x)$ using Step 6 and equations (3.18) and (3.19).

Step 8: Approximate $u_{i+1} = \varphi_{1i}(x_{i+1})$ and $v_{i+1} = \varphi_{2i}(x_{i+1})$.

4 Results and discussion

This section aims to evaluate the performance and effectiveness of three different numerical methods for solving systems of NVIEs-II. The MADM, the H-JM, and the CNPSFM will be applied to a set of illustrative examples. We will first present a graphical comparison between the approximate solution obtained from each method and the exact analytical solution. Subsequently, a detailed quantitative comparison will be provided through tables summarizing the absolute error values at different points to verify the accuracy and reliability of the proposed methods.

4.1 Application of the MADM

Example 4.1. Consider the following system of NVIEs-II [21]:

$$\begin{cases} u(x) = \cosh(x) - x + \int_0^x \left(u^2(t) - v^2(t)\right) dt, \\ v(x) = \sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\left(u^2(t) + v^2(t)\right) dt. \end{cases} \tag{4.1}$$

This system has the exact solution:

$$\begin{cases} u(x) = \cosh(x), \\ v(x) = \sinh(x). \end{cases} \tag{4.2}$$

First Modification Splitting $f_1(x)$ and $f_2(x)$ into two parts:

$$\begin{cases} f_{11}(x) = \cosh(x), & f_{12}(x) = -x, \\ f_{21}(x) = \sinh(x), & f_{22}(x) = -\dfrac{1}{2}\sinh^2(x). \end{cases} \tag{4.3}$$

Using the recursion relation (3.8), we get:

$$u_0(x) = f_{11}(x) = \cosh(x), \tag{4.4}$$
$$v_0(x) = f_{21}(x) = \sinh(x), \tag{4.5}$$

$$\begin{aligned} u_1(x) &= f_{12}(x) + \int_0^x \left(A_0^{11} - A_0^{12}\right) dt \\ &= f_{12}(x) + \int_0^x \left(u_0^2(t) - v_0^2(t)\right) dt \\ &= -x + \int_0^x \left(\cosh^2(t) - \sinh^2(t)\right) dt = 0, \end{aligned} \tag{4.6}$$

$$\begin{aligned} v_1(x) &= f_{22}(x) + \int_0^x (x-t)\left(A_0^{21} + A_0^{22}\right) dt \\ &= f_{22}(x) + \int_0^x (x-t)\left(u_0^2(t) + v_0^2(t)\right) dt \\ &= -\frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\left(\cosh^2(t) + \sinh^2(t)\right) dt = 0, \end{aligned} \tag{4.7}$$

$$u_{n+1}(x) = \int_0^x \left(A_n^{11} - A_n^{12}\right) dt = 0, \quad n \geq 1, \tag{4.8}$$
$$v_{n+1}(x) = \int_0^x (x-t)\left(A_n^{21} + A_n^{22}\right) dt = 0, \quad n \geq 1. \tag{4.9}$$

This results in the precise solutions:

$$\begin{cases} u(x) = \cosh(x), \\ v(x) = \sinh(x). \end{cases} \tag{4.10}$$

Second Modification To apply the second modified technique, let us first expand the functions $f_1(x)$ and $f_2(x)$ in terms of their Taylor series expansion:

$$f_1(x) = 1 - x + \frac{1}{2}x^2 + \frac{1}{24}x^4 + \frac{1}{720}x^6 + \frac{1}{40320}x^8 + O(x^{10}), \tag{4.11}$$
$$f_2(x) = x - \frac{1}{2}x^2 + \frac{1}{6}x^3 - \frac{1}{6}x^4 + \frac{1}{120}x^5 - \frac{1}{45}x^6 + \frac{1}{5040}x^7 - \frac{1}{630}x^8 + O(x^9). \tag{4.12}$$

Applying the recurrence relation (3.9):

$$u_0(x) = f_{1,0}(x) = 1, \tag{4.13}$$
$$v_0(x) = f_{2,0}(x) = x, \tag{4.14}$$

$$\begin{aligned} u_1(x) &= f_{1,1}(x) + \int_0^x \left(A_0^{11} - A_0^{12}\right) dt \\ &= -x + \int_0^x \left(u_0^2 - v_0^2\right) dt \\ &= -x + \int_0^x (1 - t^2)\, dt = -\frac{1}{3}x^3, \end{aligned} \tag{4.15}$$

$$\begin{aligned} v_1(x) &= f_{2,1}(x) + \int_0^x (x-t)\left(A_0^{21} + A_0^{22}\right) dt \\ &= -\frac{1}{2}x^2 + \int_0^x (x-t)(u_0^2 + v_0^2)\, dt \\ &= -\frac{1}{2}x^2 + \int_0^x (x-t)(1+t^2)\, dt = \frac{1}{12}x^4, \end{aligned} \tag{4.16}$$

$$\begin{aligned} u_2(x) &= f_{1,2}(x) + \int_0^x \left(A_1^{11} - A_1^{12}\right) dt \\ &= \frac{1}{2}x^2 + \int_0^x (2u_0u_1 - 2v_0v_1)\, dt \\ &= \frac{1}{2}x^2 - \frac{1}{6}x^4 - \frac{1}{36}x^6, \end{aligned} \tag{4.17}$$

$$\begin{aligned} v_2(x) &= f_{2,2}(x) + \int_0^x (x-t)\left(A_1^{21} + A_1^{22}\right) dt \\ &= \frac{1}{6}x^3 + \int_0^x (x-t)(2u_0u_1 + 2v_0v_1)\, dt \\ &= \frac{1}{6}x^3 - \frac{1}{30}x^5 + \frac{1}{252}x^7, \end{aligned} \tag{4.18}$$

$$\begin{aligned} u_3(x) &= f_{1,3}(x) + \int_0^x \left(A_2^{11} - A_2^{12}\right) dt \\ &= \frac{1}{24}x^4 + \int_0^x \left[(2u_0u_2 + u_1^2) - (2v_0v_2 + v_1^2)\right] dt \\ &= \frac{1}{3}x^3 + \frac{1}{24}x^4 - \frac{2}{15}x^5 + \frac{11}{630}x^7 - \frac{5}{3024}x^9, \end{aligned} \tag{4.19}$$

$$\begin{aligned} v_3(x) &= f_{2,3}(x) + \int_0^x (x-t)\left(A_2^{21} + A_2^{22}\right) dt \\ &= -\frac{1}{6}x^4 + \int_0^x (x-t)\left[(2u_0u_2 + u_1^2) + (2v_0v_2 + v_1^2)\right] dt \\ &= -\frac{1}{12}x^4 - \frac{1}{5040}x^8 + \frac{1}{6048}x^{10}, \end{aligned} \tag{4.20}$$

$$\begin{aligned} u_4(x) &= f_{1,4}(x) + \int_0^x \left(A_3^{11} - A_3^{12}\right) dt \\ &= \int_0^x \left[(2u_0u_3 + 2u_1u_2) - (2v_0v_3 + 2v_1v_2)\right] dt \\ &= \frac{1}{6}x^4 + \frac{1}{60}x^5 - \frac{17}{240}x^6 + \frac{149}{10080}x^8 + \frac{2}{945}x^{10} - \frac{1}{12096}x^{12}, \end{aligned} \tag{4.21}$$

$$
\begin{aligned}
v_4(x) &= f_{2,4}(x) + \int_0^x (x-t)\left(A_3^{21} + A_3^{22}\right) dt \\
&= \frac{1}{120}x^5 + \int_0^x (x-t)\left[(2u_0u_3 + 2u_1u_2) + (2v_0v_3 + 2v_1v_2)\right] dt \\
&= \frac{1}{24}x^5 + \frac{1}{360}x^6 - \frac{23}{1260}x^7 + \frac{73}{30240}x^9 + \frac{1}{11880}x^{11} + \frac{1}{157248}x^{13},
\end{aligned} \tag{4.22}
$$

$$
\begin{aligned}
u_5(x) &= f_{1,5}(x) + \int_0^x \left(A_4^{11} - A_4^{12}\right) dt \\
&= \frac{1}{40320}x^8 + \int_0^x \left[(2u_0u_4 + 2u_1u_3 + u_2^2) - (2v_0v_4 + 2v_1v_3 + v_2^2)\right] dt \\
&= \frac{7}{60}x^5 + \frac{1}{180}x^6 - \frac{11}{120}x^7 - \frac{167}{40320}x^8 + \frac{907}{45360}x^9 \\
&\quad - \frac{409}{831600}x^{11} + \frac{131}{926640}x^{13} - \frac{37}{9906624}x^{15},
\end{aligned} \tag{4.23}
$$

$$
\begin{aligned}
v_5(x) &= f_{2,5}(x) + \int_0^x (x-t)\left(A_4^{21} + A_4^{22}\right) dt \\
&= -\frac{1}{45}x^6 + \int_0^x (x-t)\left[(2u_0u_4 + 2u_1u_3 + u_2^2) + (2v_0v_4 + 2v_1v_3 + v_2^2)\right] dt \\
&= -\frac{7}{16200}x^{12} - \frac{1}{56700}x^{13} + \frac{151}{907200}x^{14} + \frac{1}{145800}x^{15} - \frac{41}{2916000}x^{16} \\
&\quad - \frac{689}{449064000}x^{18} - \frac{197}{1021620600}x^{20} - \frac{37}{7132769280}x^{22}.
\end{aligned} \tag{4.24}
$$

Summing the series, we obtain:

$$
\begin{cases}
u(x) = 1 + \dfrac{1}{2}x^2 + \dfrac{1}{24}x^4 + \cdots = \cosh(x), \\
v(x) = x + \dfrac{1}{6}x^3 + \dfrac{1}{120}x^5 + \cdots = \sinh(x).
\end{cases} \tag{4.25}
$$

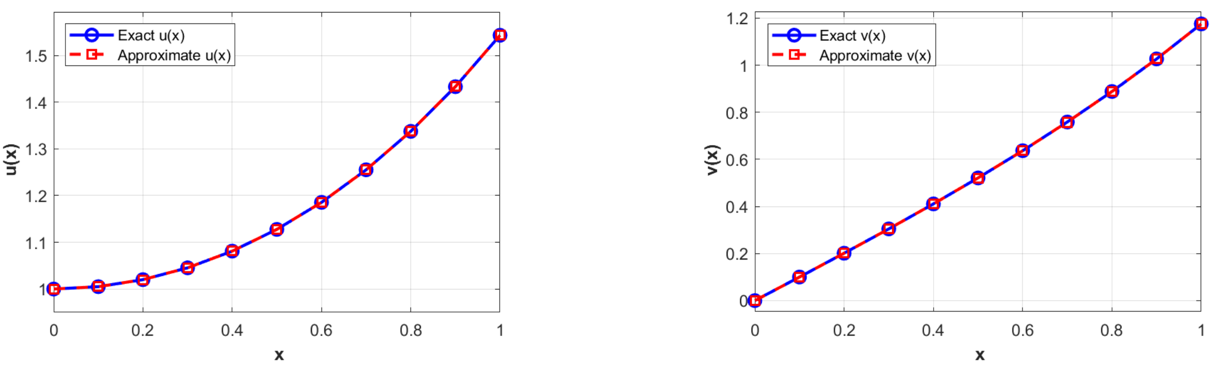


Figure 4.1: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.1 by MADM-II.

Figure 4.1 demonstrates a visual comparison between the exact and approximate solutions obtained by applying the MADM-II method. As shown in the left plot, the approximate solution of $u(x)$ (represented by red squares) closely follows the exact solution (represented by blue circles). Similarly, the right plot illustrates the high accuracy of the method for $v(x)$, where both solutions are almost overlapping. This reflects the efficiency and high accuracy of the MADM-II method in solving the given nonlinear system.

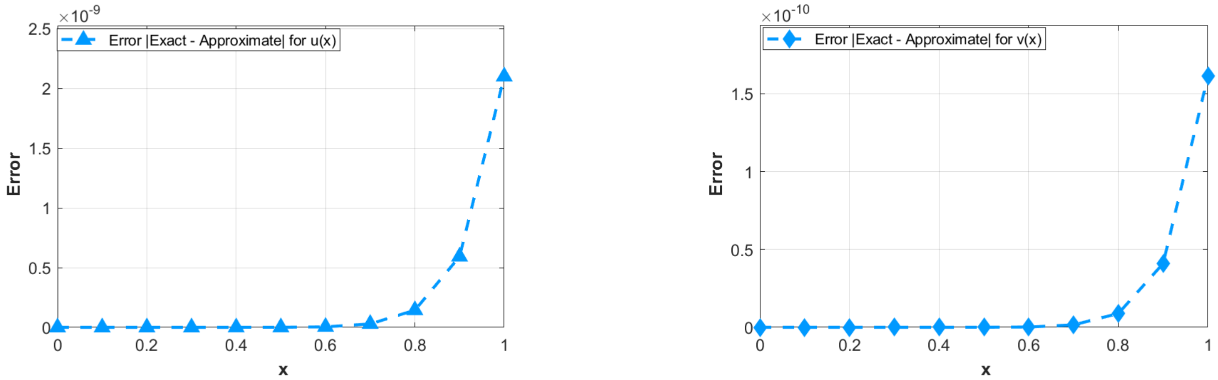


Figure 4.2: Absolute error of $u(x)$ and $v(x)$ for Example 4.1 by MADM-II.

Figure 4.2 illustrates the absolute errors $|\text{Exact} - \text{Approximate}|$ for both $u(x)$ and $v(x)$. The errors remain extremely small across the entire interval, with maximum values on the order of 10^{-9} for $u(x)$ and 10^{-10} for $v(x)$. This further confirms the high precision of the MADM-II approach. The errors are negligible except near the endpoint $x = 1$, where a slight increase is observed but remains within acceptable bounds for practical computations.

Example 4.2. Consider the following system of NVIEs-II [21]:

$$\begin{cases} u(x) = e^x - \sinh(2x) + \int_0^x \left(u^2(t) + v^2(t)\right) dt, \\ v(x) = e^{-x} + 1 - \cosh(2x) + \int_0^x \left(u^2(t) - v^2(t)\right) dt, \end{cases} \tag{4.26}$$

with exact solution:

$$\begin{cases} u(x) = e^x, \\ v(x) = e^{-x}. \end{cases}$$

First Modification: Splitting $f_1(x), f_2(x)$ into two parts:

$$\begin{cases} f_{11}(x) = e^x, \\ f_{12}(x) = -\sinh(2x), \\ f_{21}(x) = e^{-x}, \\ f_{22}(x) = 1 - \cosh(2x). \end{cases}$$

Using the recursion relation (3.8), we get:

$$\begin{aligned} u_0(x) &= f_{11}(x) = e^x, \\ v_0(x) &= f_{21}(x) = e^{-x}, \\ u_1(x) &= f_{12}(x) + \int_0^x (A_0^{11} + A_0^{12})dt = f_{12}(x) + \int_0^x (u_0^2(t) + v_0^2(t))dt \\ &= -\sinh(2x) + \int_0^x (e^{2t} + e^{-2t})dt = 0, \\ v_1(x) &= f_{22}(x) + \int_0^x (A_0^{21} - A_0^{22})dt = f_{22}(x) + \int_0^x (u_0^2(t) - v_0^2(t))dt \\ &= 1 - \cosh(2x) + \int_0^x (e^{2t} - e^{-2t})dt = 0, \\ u_{n+1}(x) &= \int_0^x (A_n^{11} + A_n^{12})dt = 0, \quad n \geq 1, \\ v_{n+1}(x) &= \int_0^x (A_n^{21} - A_n^{22})dt = 0, \quad n \geq 1. \end{aligned}$$

This results in the precise solutions:

$$\begin{cases} u(x) = e^x, \\ v(x) = e^{-x}. \end{cases}$$

Second Modification: To apply the second modified technique, let us first expand the functions $f_1(x)$ and $f_2(x)$ in terms of their Taylor series expansion:

$$\begin{aligned} f_1(x) &= 1 - x + \frac{1}{2}x^2 - \frac{7}{6}x^3 + \frac{1}{24}x^4 - \frac{31}{120}x^5 + \frac{1}{720}x^6 - \frac{127}{5040}x^7 + O(x^8), \\ f_2(x) &= 1 - x - \frac{3}{2}x^2 - \frac{1}{6}x^3 - \frac{5}{8}x^4 - \frac{1}{120}x^5 - \frac{7}{80}x^6 - \frac{1}{5040}x^7 + O(x^8). \end{aligned}$$

Applying the recurrence relation (3.9):

$$\begin{aligned} u_0(x) &= f_{1,0}(x) = 1, \\ v_0(x) &= f_{2,0}(x) = 1, \\ u_1(x) &= f_{1,1}(x) + \int_0^x (A_0^{11} + A_0^{12})dt = -x + \int_0^x (u_0^2(t) + v_0^2(t))dt = x, \\ v_1(x) &= f_{2,1}(x) + \int_0^x (A_0^{21} - A_0^{22})dt = -x + \int_0^x (u_0^2(t) - v_0^2(t))dt = -x, \\ u_2(x) &= f_{1,2}(x) + \int_0^x (A_1^{11} + A_1^{12})dt = \frac{1}{2}x^2 + \int_0^x (2u_0u_1 + 2v_0v_1)dt = \frac{1}{2}x^2, \\ v_2(x) &= f_{2,2}(x) + \int_0^x (A_1^{21} - A_1^{22})dt = -\frac{3}{2}x^2 + \int_0^x (2u_0u_1 - 2v_0v_1)dt = \frac{1}{2}x^2, \\ u_3(x) &= f_{1,3}(x) + \int_0^x (A_2^{11} + A_2^{12})dt = -\frac{7}{6}x^3 + \int_0^x \left[(2u_0u_2 + u_1^2) + (2v_0v_2 + v_1^2)\right]dt = \frac{1}{6}x^3, \\ v_3(x) &= f_{2,3}(x) + \int_0^x (A_2^{21} - A_2^{22})dt = -\frac{1}{6}x^3 + \int_0^x \left[(2u_0u_2 + u_1^2) - (2v_0v_2 + v_1^2)\right]dt = -\frac{1}{6}x^3, \\ u_4(x) &= f_{1,4}(x) + \int_0^x (A_3^{11} + A_3^{12})dt = \frac{1}{24}x^4 + \int_0^x \left[(2u_0u_3 + 2u_1u_2) + (2v_0v_3 + 2v_1v_2)\right]dt = \frac{1}{24}x^4, \\ v_4(x) &= f_{2,4}(x) + \int_0^x (A_3^{21} - A_3^{22})dt = -\frac{5}{8}x^4 + \int_0^x \left[(2u_0u_3 + 2u_1u_2) - (2v_0v_3 + 2v_1v_2)\right]dt = \frac{1}{24}x^4. \end{aligned}$$

$$u_5(x) = f_{1,5}(x) + \int_0^x (A_4^{11} + A_4^{12})dt$$
$$= -\frac{31}{120}x^5 + \int_0^x [(2u_0u_4 + 2u_1u_3 + u_2^2) + (2v_0v_4 + 2v_1v_3 + v_2^2)]dt = \frac{1}{120}x^5,$$
$$v_5(x) = f_{2,5}(x) + \int_0^x (A_4^{21} - A_4^{22})dt$$
$$= -\frac{1}{120}x^5 + \int_0^x [(2u_0u_4 + 2u_1u_3 + u_2^2) - (2v_0v_4 + 2v_1v_3 + v_2^2)]dt = -\frac{1}{120}x^5.$$

The approximate solution is:

$$\begin{cases} u(x) = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + \cdots = e^x, \\ v(x) = 1 - x + \frac{1}{2}x^2 - \frac{1}{6}x^3 + \frac{1}{24}x^4 - \frac{1}{120}x^5 + \cdots = e^{-x}. \end{cases}$$

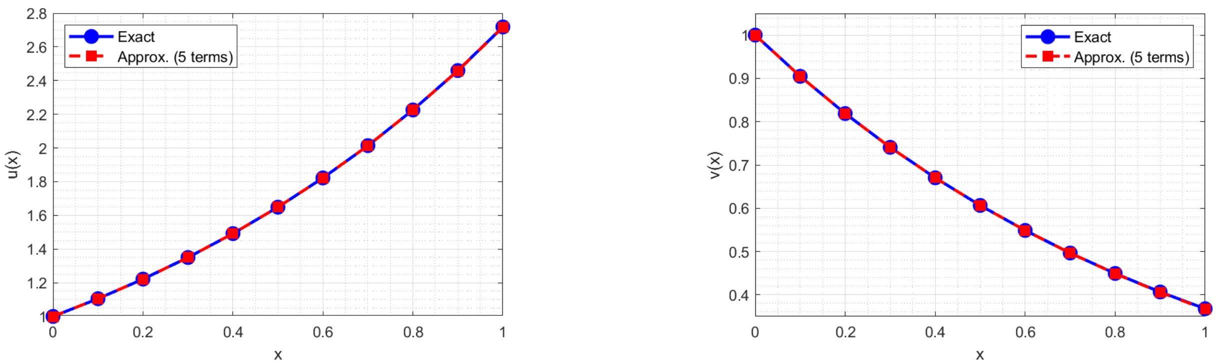


Figure 4.3: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.2 by MADM-II.

Figure 4.3 presents the comparison between the exact solutions and the approximate solutions obtained by applying MADM-II with five terms. In both subfigures, it is observed that the approximate solutions closely match the exact solutions across the entire interval $x \in [0, 1]$, demonstrating the robustness and high precision of the MADM-II method.

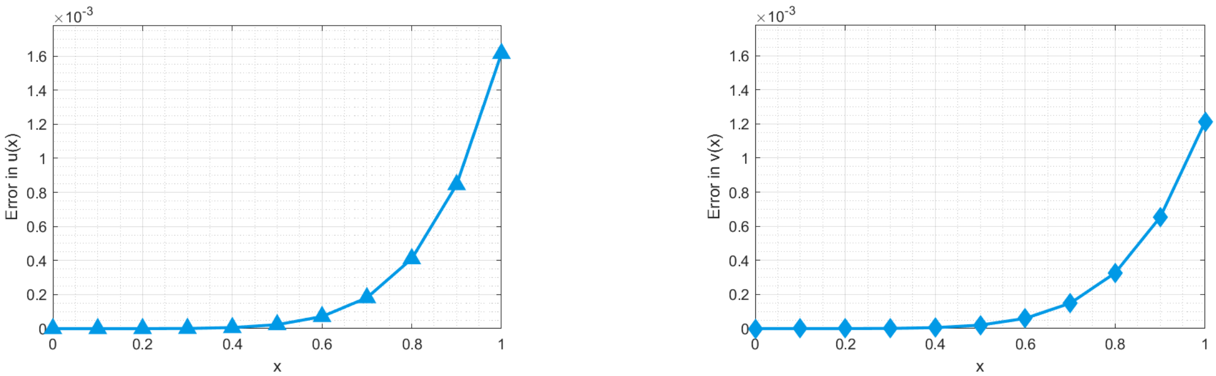


Figure 4.4: Absolute error of $u(x)$ and $v(x)$ for Example 4.2 by MADM-II.

The absolute error analysis displayed in Figure 4.4 indicates that the maximum absolute error for both $u(x)$ and $v(x)$ is approximately 1.6×10^{-3}. The error remains relatively small and stable over most of the domain, with a slight increase near $x = 1$, confirming the efficiency of the method even for nonlinear and more complex forms of the NVIEs-II.

4.2 Application of the H-JM

Example 4.3. Consider the following system of NVIEs-II:

$$\begin{cases} u(x) = \cosh(x) - x + \int_0^x \left(u^2(t) - v^2(t)\right)dt, \\ v(x) = \sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\left(u^2(t) + v^2(t)\right)dt, \end{cases} \tag{4.27}$$

with exact solution:

$$\begin{cases} u(x) = \cosh(x), \\ v(x) = \sinh(x). \end{cases}$$

When we apply the algorithm of the new technique H-JM to Equation (4.27), we obtain:

$$\begin{cases} \sum_{i=0}^{\infty} u_i(x) = \cosh(x) - x + \int_0^x \Big(\sum_{i=0}^{\infty} u_i^2(t) - \sum_{i=0}^{\infty} v_i^2(t)\Big)dt, \\ \sum_{i=0}^{\infty} v_i(x) = \sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\Big(\sum_{i=0}^{\infty} u_i^2(t) + \sum_{i=0}^{\infty} v_i^2(t)\Big)dt. \end{cases} \tag{4.28}$$

$$\begin{aligned}
u_0 &= 1, \\
v_0 &= 0, \\
u_1 &= xD_x\Big(\cosh(x) - x + \int_0^x (1^2 - 0^2)dt\Big)_{x=0} = 0, \\
v_1 &= xD_x\Big(\sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)(1+0)dt\Big)_{x=0} = x, \\
u_2 &= \frac{x^2}{2!}D_x^2\Big(\cosh(x) - x + \int_0^x (1^2 - t^2)dt\Big)_{x=0} = \frac{x^2}{2!}, \\
v_2 &= \frac{x^2}{2!}D_x^2\Big(\sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)(1+t^2)dt\Big)_{x=0} = 0, \\
u_3 &= \frac{x^3}{3!}D_x^3\Big(\cosh(x) - x + \int_0^x \big((1+\frac{t^2}{2!})^2 - t^2\big)dt\Big)_{x=0} = 0, \\
v_3 &= \frac{x^3}{3!}D_x^3\Big(\sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\big((1+\frac{t^2}{2!})^2 + t^2\big)dt\Big)_{x=0} = \frac{x^3}{3!}, \\
u_4 &= \frac{x^4}{4!}D_x^4\Big(\cosh(x) - x + \int_0^x \big((1+\frac{t^2}{2!})^2 - (t+\frac{t^3}{3!})^2\big)dt\Big)_{x=0} = \frac{x^4}{4!}, \\
v_4 &= \frac{x^4}{4!}D_x^4\Big(\sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\big((1+\frac{t^2}{2!})^2 + (t+\frac{t^3}{3!})^2\big)dt\Big)_{x=0} = 0.
\end{aligned}$$

The general recurrence relations are:

$$u_{i+1}(x) = \frac{x^{i+1}}{(i+1)!}D_x^{i+1}\Big(\cosh(x) - x + \int_0^x \Big(\sum_{j=0}^{i} u_j^2(t) - \sum_{j=0}^{i} v_j^2(t)\Big)dt\Big)_{x=0} = \begin{cases} \frac{x^{i+1}}{(i+1)!} & \text{if } i+1 \text{ is even}, \\ 0 & \text{if } i+1 \text{ is odd}, \end{cases}$$

$$v_{i+1}(x) = \frac{x^{i+1}}{(i+1)!}D_x^{i+1}\Big(\sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\Big(\sum_{j=0}^{i} u_j^2(t) + \sum_{j=0}^{i} v_j^2(t)\Big)dt\Big)_{x=0} = \begin{cases} 0 & \text{if } i+1 \text{ is e} \\ \frac{x^{i+1}}{(i+1)!} & \text{if } i+1 \text{ is o} \end{cases}$$

The approximate solution of Equation (4.27) is therefore:

$$\begin{cases} u(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots = \sum_{i=0}^{\infty} \frac{x^{2i}}{(2i)!}, \\ v(x) = x + \frac{x^3}{3!} + \cdots = \sum_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)!}. \end{cases}$$

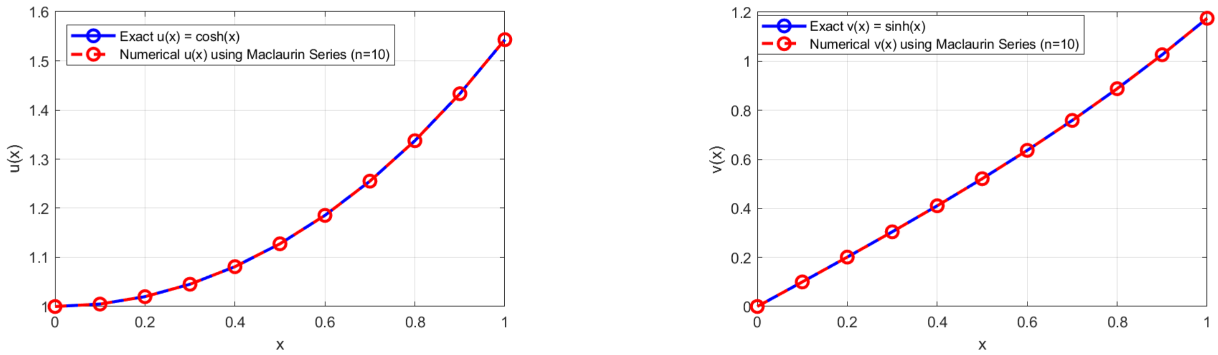


Figure 4.5: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.3 by H-JM.

Figure 4.5 shows the comparison of the exact and approximate solutions using H-JM with ten terms of the Maclaurin series expansion. The numerical results match the exact solutions perfectly across the entire domain, demonstrating the very high precision results that can be obtained with H-JM for this problem.

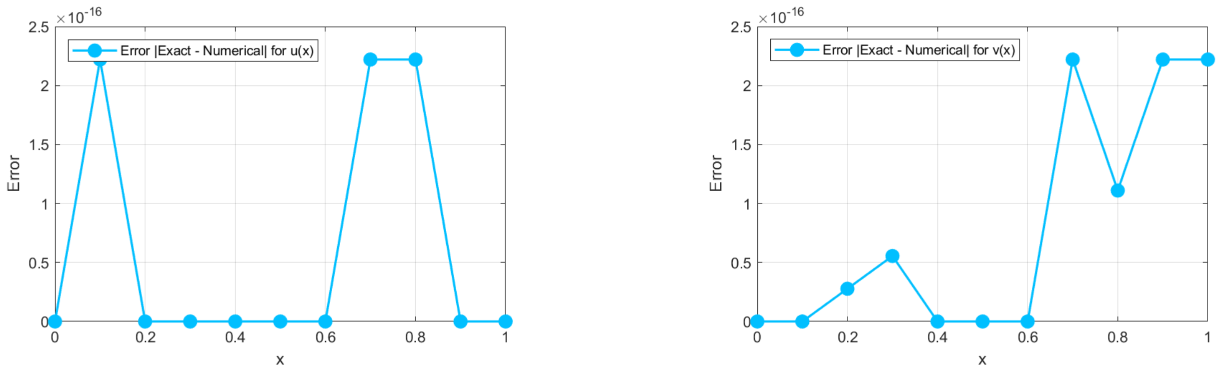


Figure 4.6: Absolute error of $u(x)$ and $v(x)$ for Example 4.3 by H-JM.

The absolute error plots demonstrate minimal errors, on the order of 10^{-16}, indicating that the approximate solutions are almost identical to the exact solutions. This confirms the outstanding accuracy and effectiveness of the H-JM method for solving the system of NVIEs-II in this example.

Example 4.4. Consider the following system of NVIEs-II:

$$\begin{cases} u(x) = e^x - \sinh(2x) + \int_0^x \left(u^2(t) + v^2(t)\right)dt, \\ v(x) = e^{-x} + 1 - \cosh(2x) + \int_0^x \left(u^2(t) - v^2(t)\right)dt, \end{cases} \tag{4.29}$$

with exact solution:

$$\begin{cases} u(x) = e^x, \\ v(x) = e^{-x}. \end{cases}$$

When we use the algorithm of the new technique H-JM on Equation (4.29), we get:

$$\begin{cases} \sum_{i=0}^{\infty} u_i(x) = e^x - \sinh(2x) + \int_0^x \left(\sum_{i=0}^{\infty} u_i^2(t) + \sum_{i=0}^{\infty} v_i^2(t)\right)dt, \\ \sum_{i=0}^{\infty} v_i(x) = e^{-x} + 1 - \cosh(2x) + \int_0^x \left(\sum_{i=0}^{\infty} u_i^2(t) - \sum_{i=0}^{\infty} v_i^2(t)\right)dt. \end{cases} \tag{4.30}$$

$$
\begin{aligned}
u_0 &= 1,\\
v_0 &= 1,\\
u_1 &= xD_x\Big(e^x - \sinh(2x) + \int_0^x (1+1)dt\Big)_{x=0} = x,\\
v_1 &= xD_x\Big(e^{-x} + 1 - \cosh(2x) + \int_0^x (1-1)dt\Big)_{x=0} = -x,\\
u_2 &= \frac{x^2}{2!}D_x^2\Big(e^x - \sinh(2x) + \int_0^x ((1+t)^2 + (1-t)^2)dt\Big)_{x=0} = \frac{x^2}{2!},\\
v_2 &= \frac{x^2}{2!}D_x^2\Big(e^{-x} + 1 - \cosh(2x) + \int_0^x ((1+t)^2 - (1-t)^2)dt\Big)_{x=0} = \frac{x^2}{2!},\\
u_3 &= \frac{x^3}{3!}D_x^3\Big(e^x - \sinh(2x) + \int_0^x ((1+t+\frac{t^2}{2!})^2 + (1-t+\frac{t^2}{2!})^2)dt\Big)_{x=0} = \frac{x^3}{3!},\\
v_3 &= \frac{x^3}{3!}D_x^3\Big(e^{-x} + 1 - \cosh(2x) + \int_0^x ((1+t+\frac{t^2}{2!})^2 - (1-t+\frac{t^2}{2!})^2)dt\Big)_{x=0} = -\frac{x^3}{3!},\\
u_4 &= \frac{x^4}{4!}D_x^4\Big(e^x - \sinh(2x) + \int_0^x ((1+t+\frac{t^2}{2!}+\frac{t^3}{3!})^2 + (1-t+\frac{t^2}{2!}-\frac{t^3}{3!})^2)dt\Big)_{x=0} = \frac{x^4}{4!},\\
v_4 &= \frac{x^4}{4!}D_x^4\Big(e^{-x} + 1 - \cosh(2x) + \int_0^x ((1+t+\frac{t^2}{2!}+\frac{t^3}{3!})^2 - (1-t+\frac{t^2}{2!}-\frac{t^3}{3!})^2)dt\Big)_{x=0} = \frac{x^4}{4!}.
\end{aligned}
$$

The general recurrence relations are:

$$
\begin{aligned}
u_{i+1}(x) &= \frac{x^{i+1}}{(i+1)!}D_x^{i+1}\Big(e^x - \sinh(2x) + \int_0^x \Big(\sum_{j=0}^{i} u_j^2(t) + \sum_{j=0}^{i} v_j^2(t)\Big)dt\Big)_{x=0} = \frac{x^{i+1}}{(i+1)!},\\
v_{i+1}(x) &= \frac{x^{i+1}}{(i+1)!}D_x^{i+1}\Big(e^{-x} + 1 - \cosh(2x) + \int_0^x \Big(\sum_{j=0}^{i} u_j^2(t) - \sum_{j=0}^{i} v_j^2(t)\Big)dt\Big)_{x=0} = (-1)^{i+1}\frac{x^{i+1}}{(i+1)!}.
\end{aligned}
$$

The approximate solution is therefore:

$$
\begin{cases}
u(x) = 1 + x + \dfrac{x^2}{2!} + \dfrac{x^3}{3!} + \dfrac{x^4}{4!} + \cdots = \displaystyle\sum_{i=0}^{\infty} \frac{x^i}{i!},\\
v(x) = 1 - x + \dfrac{x^2}{2!} - \dfrac{x^3}{3!} + \dfrac{x^4}{4!} + \cdots = \displaystyle\sum_{i=0}^{\infty} (-1)^i \frac{x^i}{i!}.
\end{cases}
$$

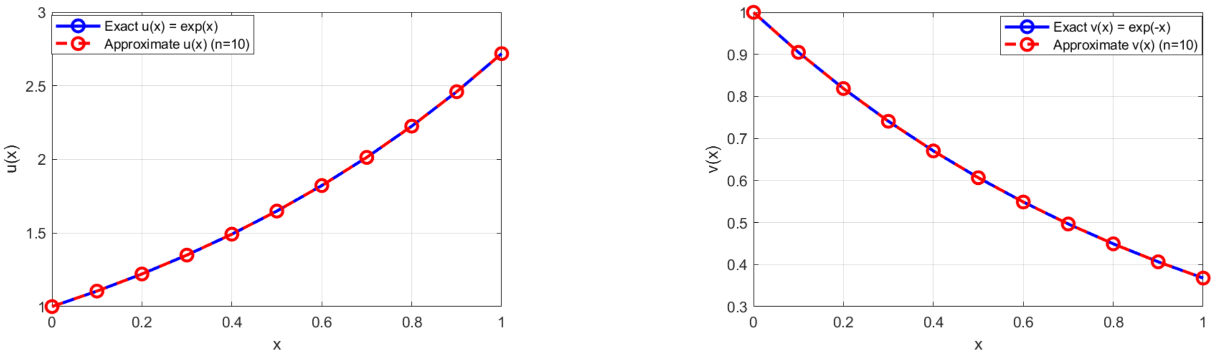


Figure 4.7: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.4by H-JM.

Figure 4.7 represents the comparison of the exact and numerical solutions using H-JM with ten terms of the Maclaurin series expansion. The approximate solutions are in agreement with the exact solutions perfectly over the interval $x \in [0, 1]$, confirming the method's strong convergence.

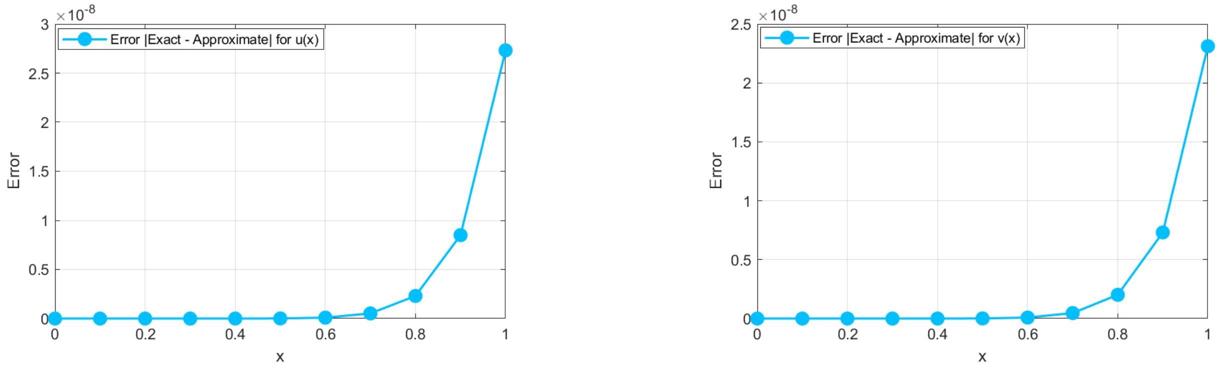


Figure 4.8: Absolute error of $u(x)$ and $v(x)$ for Example 4.4 by H-JM.

The absolute error plots confirm the very high accuracy of H-JM, as the maximum errors for both $u(x)$ and $v(x)$ are extremely small, approaching 10^{-8}, demonstrating the capability of the method to provide highly accurate results for solving NVIEs-II.

4.3 Application of the CNPSFM

Example 4.5. Consider the following nonlinear system of Volterra integral equations of the second kind:

$$\begin{cases} u(x) = \cosh(x) - x + \int_0^x \left(u^2(t) - v^2(t)\right)dt, \\ v(x) = \sinh(x) - \frac{1}{2}\sinh^2(x) + \int_0^x (x-t)\left(u^2(t) + v^2(t)\right)dt, \end{cases} \tag{4.31}$$

which has the exact solution:

$$\begin{cases} u(x) = \cosh(x), \\ v(x) = \sinh(x). \end{cases}$$

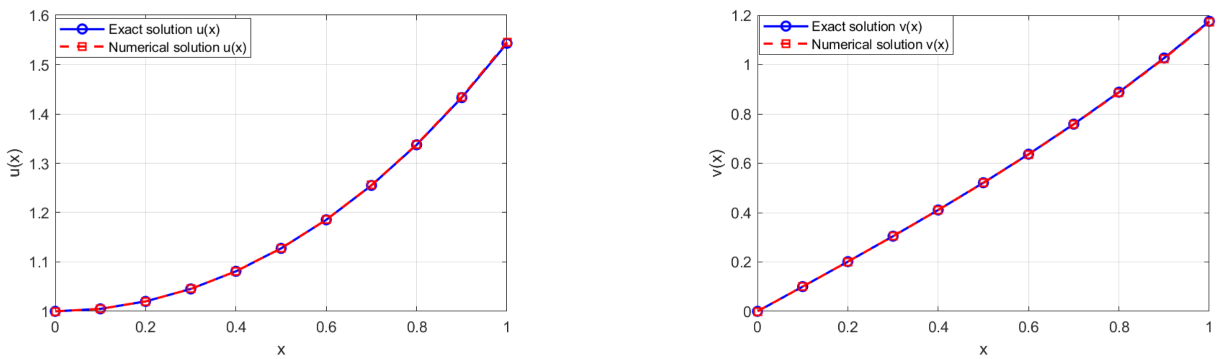


Figure 4.9: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.5 by CNPSFM.

The graphical illustration in Figure 4.9 clearly shows the high agreement between the exact solutions $u(x) = \cosh(x)$ and $v(x) = \sinh(x)$, and the corresponding approximate solutions obtained by the CNPSFM approach. The approximate solutions closely track the exact curves across the entire domain $x \in [0, 1]$, with only minor deviations visible at the upper boundary, reflecting the maximum errors obtained in the numerical computations.

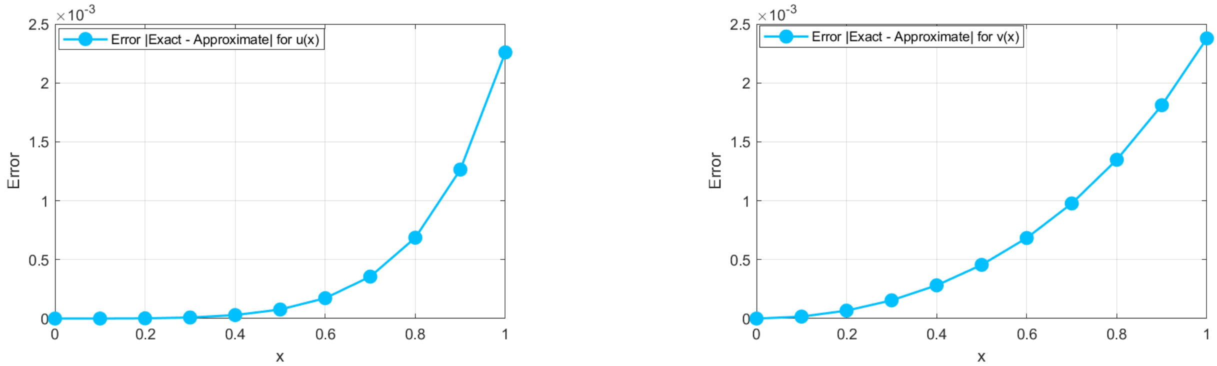


Figure 4.10: Absolute error of $u(x)$ and $v(x)$ for Example 4.5 by CNPSFM.

The absolute error plots shown in Figure 4.10 further demonstrate the accuracy of the CNPSFM method. The maximum absolute error for $u(x)$ is approximately 2.26×10^{-3} and for $v(x)$ it is approximately 2.38×10^{-3}. These small errors confirm the effectiveness of CNPSFM in solving the system of NVIEs-II with remarkable precision.

Example 4.6. Consider the following system of NVIEs-II:

$$\begin{cases} u(x) = e^x - \sinh(2x) + \int_0^x \left(u^2(t) + v^2(t)\right)dt, \\ v(x) = e^{-x} + 1 - \cosh(2x) + \int_0^x \left(u^2(t) - v^2(t)\right)dt, \end{cases} \tag{4.32}$$

which has the exact solution:

$$\begin{cases} u(x) = e^x, \\ v(x) = e^{-x}. \end{cases}$$

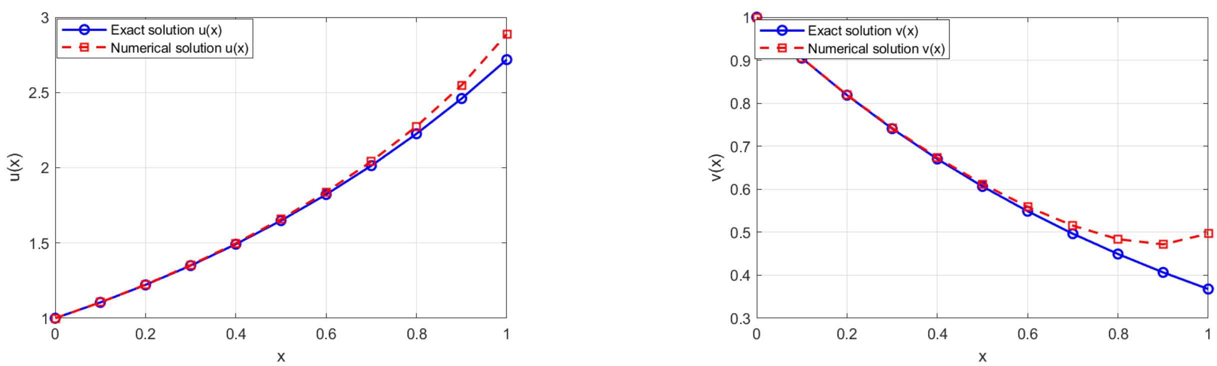
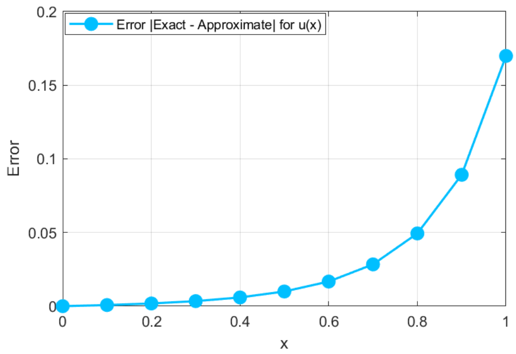


Figure 4.11: The exact and numerical solutions of $u(x)$ and $v(x)$ for Example 4.6by CNPSFM.

Figure 4.11 illustrates the comparative behavior of the exact and approximate solutions for the functions $u(x)$ and $v(x)$ in Example 4.6 using the CNPSFM. While the method maintains a reasonable level of accuracy for lower values of x, a noticeable deviation occurs as x increases. This is particularly evident in the solution for $u(x)$, which exhibits greater divergence due to the rapid exponential growth of the exact solution. Nonetheless, the method successfully captures the overall behavior and trend of the solution profiles.



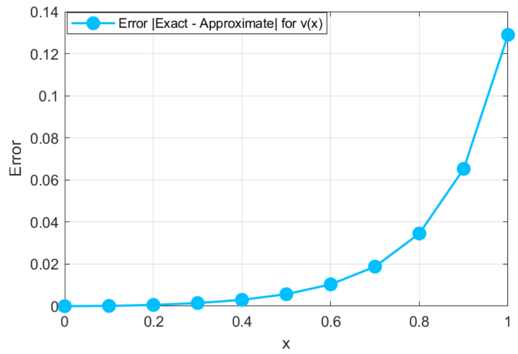


Figure 4.12: Absolute error of $u(x)$ and $v(x)$ for Example 4.6by CNPSFM.

The absolute error plots shown in Figure 4.12 further demonstrate the numerical accuracy of the CNPSFM method. The maximum absolute error for $u(x)$ is approximately 1.70×10^{-1}, while for $v(x)$ it is approximately 1.29×10^{-1}. Although the errors increase gradually towards the end of the interval, the overall magnitudes remain reasonably low, confirming the effectiveness of CNPSFM in providing reliable approximations for the system of NVIEs-II. The method performs particularly well near the initial region, highlighting its strength in capturing smooth solution behaviors.

5 Comparative analysis of methods

5.1 Consolidated numerical results tables

Following the presentation of analytical and numerical solutions alongside their corresponding graphical representations for each example, Tables 5.1 and 5.2 (for Example 4.1) and Tables 5.3 and 5.4 (for Example 4.2) provide a comprehensive, side-by-side comparison of the performance of the three methods (MADM-II, H-JM, CNPSFM) in approximating both functions $u(x)$ and $v(x)$ over the domain $[0, 1]$. These tables display exact and approximate values concurrently, along with a detailed analysis of the absolute error for each method at discrete x-values. This unified comparison enables a clear assessment of the accuracy and reliability of each method across the full domain of x and for both studied examples.

Table 5.1: Comparison of exact and numerical solutions for Example 4.1 by MADM-II, H-JM, and CNPSFM.

x	u_{exact}	$u_{\text{Approximate}}$			v_{exact}	$v_{\text{Approximate}}$		
		MADM-II	H-JM	CNPSFM		MADM-II	H-JM	CNPSFM
0	1	1	1	1	0	0	0	0
0.1	1.005	1.005	1.005	1.005	0.10017	0.10017	0.10017	0.10015
0.2	1.0201	1.0201	1.0201	1.0201	0.20134	0.20134	0.20134	0.20127
0.3	1.0453	1.0453	1.0453	1.0453	0.30452	0.30452	0.30452	0.30437
0.4	1.0811	1.0811	1.0811	1.0811	0.41075	0.41075	0.41075	0.41047
0.5	1.1276	1.1276	1.1276	1.1277	0.5211	0.5211	0.5211	0.52064
0.6	1.1855	1.1855	1.1855	1.1856	0.63665	0.63665	0.63665	0.63597
0.7	1.2552	1.2552	1.2552	1.2555	0.75858	0.75858	0.75858	0.75761
0.8	1.3374	1.3374	1.3374	1.3381	0.88811	0.88811	0.88811	0.88676
0.9	1.4331	1.4331	1.4331	1.4344	1.0265	1.0265	1.0265	1.0247
1	1.5431	1.5431	1.5431	1.5453	1.1752	1.1752	1.1752	1.1728

Table 5.1 shows that all three methods give approximations that are very close to the exact solution, especially for the $u(x)$ component, where the errors are very small. There are small differences in the $v(x)$ component visible as x increases, especially with the CNPSFM method. However, all methods are stable and consistent across the domain.

Table 5.2: Comparison of absolute errors for Example 4.1 by MADM-II, H-JM, and CNPSFM.

x	$\|u_{\text{exact}} - u_{\text{approx}}\|$			$\|v_{\text{exact}} - v_{\text{approx}}\|$		
	MADM-II	H-JM	CNPSFM	MADM-II	H-JM	CNPSFM
0	0	0	0	0	0	0
0.1	2.2204×10^{-16}	2.2204×10^{-16}	1.8583×10^{-7}	2.7756×10^{-17}	2.7756×10^{-17}	1.6689×10^{-5}
0.2	0	0	1.926×10^{-6}	1.1102×10^{-16}	1.1102×10^{-16}	6.7455×10^{-5}
0.3	1.1102×10^{-15}	0	9.1407×10^{-6}	1.0547×10^{-15}	0	0.00015454
0.4	3.5083×10^{-14}	0	2.9575×10^{-5}	1.9651×10^{-14}	0	0.00028194
0.5	5.1026×10^{-13}	0	7.6736×10^{-5}	2.1005×10^{-13}	0	0.00045655
0.6	4.5532×10^{-12}	0	0.00017317	2.2204×10^{-12}	0	0.00068389
0.7	3.8974×10^{-11}	2.2204×10^{-16}	0.00035998	1.5594×10^{-12}	2.2204×10^{-16}	0.00097716
0.8	1.4397×10^{-10}	2.2204×10^{-16}	0.00068616	8.8555×10^{-12}	1.1102×10^{-16}	0.0012481
0.9	5.9225×10^{-10}	0	0.0012645	4.0978×10^{-11}	2.2204×10^{-16}	0.0018106
1.0	2.0992×10^{-9}	0	0.0022592	1.6136×10^{-10}	2.2204×10^{-16}	0.0023778

The error analysis reveals that both MADM-II and H-JM exhibit very low absolute errors for $u(x)$, typically less than machine precision. The CNPSFM method has slightly larger errors, but it is also very accurate, especially for smaller x. For $v(x)$, the errors again have similar trends, with CNPSFM showing relatively larger errors for larger x, but still in the acceptable range for engineering practice.

Table 5.3: Comparison of exact and numerical solutions for Example 3.2 by MADM-II, H-JM, and CNPSFM.

x	u_{exact}	$u_{\text{Approximate}}$			v_{exact}	$v_{\text{Approximate}}$		
		MADM-II	H-JM	CNPSFM		MADM-II	H-JM	CNPSFM
0	1	1	1	1	1	1	1	1
0.1	1.1052	1.1052	1.1052	1.1059	0.90484	0.90484	0.90484	0.90498
0.2	1.2214	1.2214	1.2214	1.2232	0.81873	0.81873	0.81873	0.81932
0.3	1.3499	1.3499	1.3499	1.3533	0.74082	0.74082	0.74082	0.74229
0.4	1.4918	1.4918	1.4918	1.4978	0.67032	0.67032	0.67032	0.67334
0.5	1.6487	1.6487	1.6487	1.6587	0.60653	0.60653	0.60653	0.61222
0.6	1.8221	1.8220	1.8221	1.8388	0.54881	0.54881	0.54881	0.55918
0.7	2.0138	2.0136	2.0138	2.0421	0.49659	0.49659	0.49659	0.51538
0.8	2.2255	2.2251	2.2255	2.2749	0.44933	0.44933	0.44933	0.48385
0.9	2.4596	2.4588	2.4596	2.5488	0.40657	0.40657	0.40657	0.47183
1.0	2.7183	2.7167	2.7183	2.8882	0.36788	0.36788	0.36788	0.49683

Table 5.3 shows that all methods approximate the exact solutions accurately for smaller x values. As x increases, some deviation appears, particularly with the CNPSFM method. Nonetheless, all methods demonstrate reliable performance across the domain.

Table 5.4: Comparison of absolute errors for Example 3.2 by MADM-II, H-JM, and CNPSFM.

x	$\lvert u_{\text{exact}} - u_{\text{approx}} \rvert$			$\lvert v_{\text{exact}} - v_{\text{approx}} \rvert$		
	MADM-II	H-JM	CNPSFM	MADM-II	H-JM	CNPSFM
0	0	0	0	0	0	0
0.1	1.409×10^{-9}	4.4409×10^{-16}	0.0007682	1.3693×10^{-9}	1.1102×10^{-16}	0.00013918
0.2	9.1494×10^{-8}	6.6613×10^{-16}	0.0018361	8.641×10^{-8}	5.5511×10^{-16}	0.00059078
0.3	1.0576×10^{-6}	4.5741×10^{-14}	0.0034347	9.7068×10^{-7}	4.3188×10^{-14}	0.0014698
0.4	6.031×10^{-6}	1.0871×10^{-12}	0.0059353	5.3794×10^{-6}	1.0166×10^{-12}	0.0030158
0.5	2.3354×10^{-5}	1.2763×10^{-11}	0.0099783	2.0243×10^{-5}	1.1742×10^{-11}	0.0056889
0.6	7.08×10^{-5}	9.5652×10^{-11}	0.016724	5.9636×10^{-5}	8.6545×10^{-11}	0.010371
0.7	0.00018129	5.259×10^{-10}	0.028382	0.00014839	4.6795×10^{-10}	0.018796
0.8	0.00041026	2.3048×10^{-9}	0.049382	0.0003263	2.0168×10^{-9}	0.034522
0.9	0.00084486	8.4948×10^{-9}	0.089194	0.00065291	7.3103×10^{-9}	0.065259
1	0.0016152	2.7313×10^{-8}	0.16987	0.0012128	2.3114×10^{-8}	0.12895

As shown in Table 5.4, the error analysis shows that both MADM-II and H-JM methods give extremely small absolute errors for $u(x)$ and $v(x)$, remaining close to machine precision for most values of x. Observed errors in the CNPSFM method show larger errors as the values of x increase, but still give acceptable accuracy levels for the types of computational applications that would be typically conducted.

5.2 Comparative performance analysis

Building upon the systematically presented numerical results in Tables 5.1–5.4 and Figures 1–12, this section provides a comprehensive comparative analysis of three numerical methods:

the MADM, the H-JM, and the CNPSFM, for solving systems of NVIEs-II. The analysis is structured into two complementary components:

(i) a systematic presentation of numerical performance through tables and figures, and

(ii) an in-depth analytical discussion highlighting convergence behavior, accuracy, stability, and computational efficiency.

5.2.1 Numerical results summary

Two benchmark examples were employed to evaluate the three methods. Absolute errors were computed and compared against exact analytical solutions. Tables 5.1–5.4 and Figures 1–12 illustrate the numerical behavior across the domain $[0, 1]$:

- **MADM:** Achieved absolute errors ranging from 10^{-16} to 2.0992×10^{-9}, exhibiting excellent agreement up to $x = 0.8$, followed by mild error accumulation near $x = 1$.
- **H-JM:** Consistently maintained machine-precision accuracy (10^{-16}–10^{-8}), with exact agreement between numerical and analytical solutions across the entire interval.
- **CNPSFM:** Produced moderate accuracy (10^{-3}–10^{-1}) with relatively stable performance in the interior domain; however, significant endpoint deviations were observed, reaching 0.16987 in Example 2.

5.2.2 Analytical discussion

Accuracy and error propagation

- **MADM:** Near-machine-precision accuracy for $x < 0.7$; small residual errors arise near the endpoint due to truncation of higher-order Adomian terms (Figures 1–2).
- **H-JM:** Exhibits uniformly negligible errors across all nodes (Tables 5.2–5.4). Its Maclaurin-based recursive structure ensures exact agreement even for systems exhibiting exponential growth (Figures 5–7).
- **CNPSFM:** Acceptable accuracy near $x = 0$; however, increasing boundary errors indicate limited robustness for solutions with sharp gradients (Figures 10–12).

Convergence behavior

- **MADM:** Demonstrates rapid initial convergence with slight stagnation near $x = 1$.
- **H-JM:** Achieves machine-precision convergence with relatively few terms, confirming global stability.
- **CNPSFM:** Converges satisfactorily for smooth solutions (e.g., hyperbolic cosine functions); however, performance degrades when solutions exhibit steep variations.

5.2.3 Stability and computational efficiency

Table 5.5 presents a qualitative comparison of stability and computational characteristics:

Table 5.5: Stability and computational efficiency comparison.

Criterion	MADM-II	H-JM	CNPSFM
Endpoint stability	Moderate	Excellent	Limited
Computational cost	Low	Moderate	High
Handling nonlinearity	Efficient	Exceptional	Variable

Key observations:

- H-JM avoids numerical integration entirely, thereby minimizing error propagation;
- MADM provides an optimal balance between computational efficiency and acceptable precision;
- CNPSFM requires refined meshes to mitigate boundary artifacts.

5.2.4 Comparative insights from benchmark examples

- **Example 1:** H-JM achieved near-zero errors; MADM produced negligible errors ($\sim 10^{-9}$); CNPSFM maintained errors within $\sim 10^{-3}$.
- **Example 2:** H-JM and MADM preserved excellent accuracy (10^{-16}–10^{-8}); CNPSFM errors increased to 10^{-1}, confirming instability for solutions with steep gradients.

5.2.5 Guidelines for method selection

This comparative study demonstrates that no single method universally dominates; rather, the optimal choice depends on problem-specific characteristics:

- **For absolute accuracy and global stability** $\rightarrow$ H-JM represents the gold standard.
- **For computational efficiency with high central-domain accuracy** $\rightarrow$ MADM offers the optimal trade-off.
- **For smooth problems with limited variations** $\rightarrow$ CNPSFM is viable, provided endpoint refinement strategies are employed.

5.2.6 Limitations and future research directions

Despite their respective strengths, all three methods exhibit certain limitations:

- **MADM:** Susceptible to endpoint error accumulation.
- **CNPSFM:** Sensitive to boundary artifacts; requires adaptive or fractional spline enhancements.
- **Common limitation:** None of the methods fully addresses singular kernels or multi-dimensional extensions.

Proposed future research directions:

1. Development of hybrid MADM/H-JM frameworks combining computational efficiency with machine-precision accuracy;
2. Implementation of adaptive CNPSFM schemes employing variable knot placement and fractional spline bases to reduce endpoint instability;
3. Design of kernel-optimized algorithms for higher-dimensional NVIEs-II systems.

5.2.7 Summary

This study reveals a clear performance hierarchy among the three methods:

- **H-JM** establishes itself as the benchmark approach, offering unmatched global accuracy and numerical stability.
- **MADM** provides an optimal balance between computational efficiency and precision, rendering it particularly suitable for engineering applications requiring rapid computation.
- **CNPSFM**, while exhibiting lower accuracy, remains valuable for smooth systems and serves as a foundation for future spline-based enhancements.

Ultimately, method selection should align with problem-specific characteristics including solution smoothness, nonlinearity structure, and boundary sensitivity. The proposed hybrid and adaptive strategies represent promising avenues for advancing the numerical treatment of nonlinear Volterra integral systems.

6 Conclusion

This study presents a comprehensive comparative analysis of three innovative numerical techniques for solving systems of NVIEs-II: the MADM, the H-JM, and the CNPSFM. Each method was rigorously evaluated using benchmark problems with known exact solutions, and their accuracy, convergence rates, and numerical stability were systematically investigated.

Both theoretical analysis and numerical experiments confirmed the convergence behavior of all three methods; however, the computational results revealed distinct performance characteristics for each approach:

- **MADM and H-JM:** Both methods demonstrated clear advantages in achieving superior numerical accuracy with minimal errors, often approaching machine precision, as evidenced in Tables 5.2 and 5.4. MADM exhibited rapid convergence, while H-JM provided exceptional numerical stability and straightforward implementation through a simple recursive Maclaurin series framework. These attributes render both methods highly suitable for applications demanding high analytical precision and reliability, such as quantum mechanical systems and electromagnetic modeling.
- **CNPSFM:** While this method displayed robust numerical stability and accurately captured the overall solution behavior, the numerical results (Tables 5.1–5.3, Figures 9–12) indicated that it produced larger errors relative to the other two methods, particularly

for larger values of the independent variable x. Nevertheless, CNPSFM remains appropriate for applications prioritizing smooth local approximations, such as viscoelasticity problems.

Based on these findings, the following practical recommendations are offered:

- **MADM** is ideal for problems requiring high precision and rapid convergence in the initial stages (e.g., population dynamics models);
- **H-JM** is well-suited for scenarios demanding stable, computationally efficient implementations with minimal overhead;
- **CNPSFM** is viable for problems exhibiting smooth solution behavior, where moderate accuracy trade-offs are acceptable.

Although all three methods demonstrate strong performance across diverse test cases, challenges persist, particularly when addressing systems with singular kernels or requiring long-time integration. Consequently, future research should focus on extending these methods to more complex configurations, including systems with delay terms, integro-differential formulations, or hybrid computational frameworks that integrate machine learning techniques for adaptive refinement and enhanced efficiency.

In conclusion, this study provides both theoretical insights and evidence-based practical guidance for selecting robust and appropriate numerical solvers for NVIEs-II, thereby reinforcing the computational viability of these approaches across a broad spectrum of scientific and engineering applications.

References

[1] A. E. Abaoub, A. S. Shkheam, and S. M. Zali, *The Adomian decomposition method of the Volterra integral equation of the second kind*, American Journal of Applied Mathematics **6**(4) (2018), 142–148. DOI

[2] E. Babolian and Z. Masouri, *Direct method to solve the Volterra integral equation of the first kind using the operational matrix with block-pulse functions*, Journal of Computational and Applied Mathematics **220**(1-2) (2008), 51–57. DOI

[3] A. Baitiche and H. Brunner, *A collocation method for nonlinear Volterra integral equations with weakly singular kernels*, Journal of Integral Equations and Applications **33**(3) (2021), 401–423. DOI

[4] H. Brunner, *Non-polynomial spline collocation for Volterra equations with weakly singular kernels*, SIAM Journal on Numerical Analysis **20**(6) (1983), 1106–1119. DOI

[5] H. Brunner, *Volterra Integral Equations: An Introduction to Theory and Applications*, Cambridge University Press, Cambridge, 2017. DOI

[6] S. Gupta and K. Kumar, *Variational iteration method and homotopy analysis for nonlinear Volterra integral equations*, Symmetry **13**(6) (2021), 1087. DOI

[7] I. Hamza and A. S. Al-Atabi, *Algorithms for solving Volterra integral equations using non-polynomial spline functions*, Wasit Journal of Pure Sciences **2**(1) (2023), 164–172. DOI

[8] S. H. Harbi, M. A. Murad, and S. N. Majeed, *A solution of the second kind of Volterra integral equations using a third-order non-polynomial spline function*, Baghdad Science Journal **12**(2) (2015), 406–411. DOI

[9] M. A. Husseini, H. K. Jassim, and A. K. Jassim, *An innovative iterative approach to solving Volterra integral equations of the second kind*, Acta Polytechnica **64**(2) (2024), 87–102. DOI

[10] E. A. E. Ibrahim, A. R. A. G. A. Abdel Rahman, N. Y. Mohammed, and N. A. H. Haroun, *Comparison between the Adomian decomposition method and numerical solutions of linear Volterra integral equations of the second kind by using the fifth order of non-polynomial spline functions*, International Journal of Applied Mathematics and Theoretical Physics **7**(3) (2021), 68–79. DOI

[11] B. F. Jumaa and A. H. Taqi, *Non-polynomial spline method for the solution of the system of two nonlinear Volterra integral equations*, Kirkuk University Journal for Scientific Studies **11**(3) (2016), 15–25. DOI

[12] A. J. Khalaf and B. A. Taha, *Numerical study of the system of nonlinear Volterra integral equations by using the spline method*, Journal of Al-Qadisiyah for Computer Science and Mathematics **13**(3) (2021), 34–43. DOI

[13] A. J. Khalaf and B. A. Taha, *The numerical solution of the Volterra integral equation of the second kind using a quartic non-polynomial spline function*, Basrah Journal of Science **40**(1) (2022), 1–9. DOI

[14] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, 1985.

[15] K. Maleknejad, P. Torabi, and S. Sauter, *Numerical solution of a non-linear Volterra integral equation*, Vietnam Journal of Mathematics **44**(1) (2016), 5–28. DOI

[16] S. Mirmostafaee and R. Eslami, *Numerical treatment of nonlinear weakly singular Volterra integral equations using cubic splines*, Applied Mathematics Letters **139** (2023), 108804. DOI

[17] M. M. Mustafa, *Volterra Integral Equations Using Non-Polynomial Spline Functions*, Master's Thesis, University of Baghdad, Iraq, 2009.

[18] A. D. Polyanin and A. V. Manzhirov, *Handbook of Integral Equations*, CRC Press, Boca Raton, 1998.

[19] R. Rach, J. S. Duan, and A. M. Wazwaz, *A new modification of the Adomian decomposition method for solving boundary value problems*, Applied Mathematics and Computation **244** (2014), 403–412. DOI

[20] A. H. Taqi and B. F. Jumaa, *Spline approaches for solving nonlinear Volterra integral equations*, Kirkuk Journal of Science **48**(1) (2021), 21–36. DOI

[21] A. M. Wazwaz, *Linear and Nonlinear Integral Equations: Methods and Applications*, Springer, Berlin, 2011.

[22] L. Xie, *A new modification of the Adomian decomposition method for Volterra integral equations of the second kind*, Journal of Applied Mathematics **2013** (2013), 795015. DOI