

Article

Homotopy Analysis Method and Physics-Informed Neural Networks for Solving Volterra Integral Equations with Discontinuous Kernels

Samad Noeiaghdam ^{1,2,*} , Md Asadujjaman Miah ³ and Sanda Micula ^{4,5,*} ¹ Institute of Mathematics, Henan Academy of Sciences, Zhengzhou 450046, China² Department of Mathematical Sciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602105, India³ Department of Applied Mathematics, University of Dhaka, Dhaka 1000, Bangladesh; sifata425@gmail.com⁴ Department of Mathematics and Computer Science, Babes-Bolyai University, 400084 Cluj-Napoca, Romania⁵ Academy of Romanian Scientists, 50044 Bucharest, Romania

* Correspondence: snoei@hnas.ac.cn (S.N.); sanda.micula@ubbcluj.ro (S.M.)

Abstract

This paper addresses first- and second-kind Volterra integral equations (VIEs) with discontinuous kernels. A hybrid method combining the Homotopy Analysis Method (HAM) and Physics-Informed Neural Networks (PINNs) is developed. The convergence of the HAM is analyzed. Benchmark examples confirm that the proposed HAM-PINNs approach achieves high accuracy and robustness, demonstrating its effectiveness for complex kernel structures.

Keywords: Volterra integral equation; discontinuous kernel; homotopy analysis method; physics-informed neural networks

MSC: 45D05; 65R20; 68T07



Academic Editor: Hyun Soo Chung

Received: 15 August 2025

Revised: 23 September 2025

Accepted: 23 September 2025

Published: 25 September 2025

Citation: Noeiaghdam, S.; Miah, M.A.; Micula, S. Homotopy Analysis Method and Physics-Informed Neural Networks for Solving Volterra Integral Equations with Discontinuous Kernels. *Axioms* **2025**, *14*, 726. <https://doi.org/10.3390/axioms14100726>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

VIEs of the first and second kinds are fundamental tools for modeling processes with memory and hereditary effects. In particular, first-kind equations, where the unknown appears only under the integral, often arise in inverse problems and are generally ill-posed. Second-kind equations, in which the unknown function appears both inside and outside the integral, are typically well-posed and suitable for forward modeling. Two important areas where VIEs play a significant role are energy systems and biology. In energy systems, they provide effective models for load leveling in batteries, where the future state depends on both current demand and historical usage. In biology, they are widely used to describe population dynamics, incorporating the influence of past generations on current growth. A wide range of analytical and numerical approaches have been developed for solving VIEs. For instance, spectral and superconvergent techniques have been proposed for both linear and nonlinear systems [1,2], while quadrature-based formulations such as barycentric rational Hermite methods have been introduced for improved numerical integration [3]. Discretization and extrapolation strategies for multidimensional cases have also been investigated [4]. Earlier works employed wavelet-based approaches [5], variational iteration [6], and sinc-collocation schemes [7], alongside semi-analytical methods like the homotopy analysis method [8] and Tau collocation [9]. To address discontinuities

in kernels, the Adomian decomposition method has been applied [10], whereas reproducing kernel methods [11] and radial basis function approximations [12] offer alternative function-based formulations. More recently, artificial neural networks have been explored as flexible approximators for VIEs [13]. Together, these contributions highlight the richness of methodologies available for tackling both theoretical and application-oriented integral equation problems.

In this paper, we focus on solving first- and second-kind VIEs with discontinuous kernels because of their applications in load leveling problems in batteries. The general form of this problem can be presented as

$$\lambda u(x) + \int_a^x k(x, t) F(u(t)) dt = f(x), \quad 0 \leq t \leq T \leq x, \quad (1)$$

where $\lambda \in \{0, 1\}$ indicates whether the equation is of the first kind ($\lambda = 0$) or second kind ($\lambda = 1$). The function F can be linear or nonlinear and $f(x)$ is a known function. The kernel $k(x, t)$ is assumed to be discontinuous along a finite number of smooth curves $\rho_i(x)$, for $i = 1, \dots, m$, which partition the integration domain $[a, x]$ as follows:

$$k(x, t) = \begin{cases} k_1(x, t), & \rho_1(x) = a \leq t \leq \rho_2(x), \\ k_2(x, t), & \rho_2(x) \leq t \leq \rho_3(x), \\ \vdots & \\ k_m(x, t), & \rho_{m-1}(x) \leq t \leq \rho_m(x) = x. \end{cases}$$

In light of these discontinuities, Equation (1) can be rewritten in a piecewise integrated form as follows:

$$\lambda u(x) + \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) F(u(t)) dt = f(x). \quad (2)$$

It should be noted that the non-negativity of the solution in the load leveling problem for energy systems has already been assumed. The first kind of this problem is among inverse problems, which need some special methods for solving. Falaleev et al. and Sidorov et al. discussed the generalized solution of this problem in [14]. In [15], the existence of a solution for the first-kind problem has been discussed theoretically. In [16], the successive approximation method has been applied for finding the numerical solution for the non-regular case. The paper in [17] presents the parametric families of solutions for the first-kind problem with piecewise smooth kernel. And finally, more theory and basis of these problems can be found in the book in [18]. Among the methods for solving these problems in both the linear and nonlinear cases as well as for the first and second kinds with a prediction on load leveling in batteries, we can mention [19,20]. Also, other methods for solving IEs can be found in [21–23].

The HAM is a powerful semi-analytical technique for solving a wide range of linear and nonlinear problems in mathematics, physics, and engineering, which has been presented for the first time by Liao [24]. The main distinctions between the HAM and other methods such as collocation based methods, ADM, HPM, and others lie in the presence of the convergence control parameter $\hbar$ and two additional auxiliary components: the auxiliary function $H(t)$ and the auxiliary linear operator L . The function $H(t)$ introduces flexibility by allowing the construction of solutions using different functional forms, such as exponential or trigonometric functions, among others, depending on the nature of the problem. A notable feature of the HAM is the parameter $\hbar$, which provides direct control over the convergence of the approximate series solution. By plotting the so-called $\hbar$ -curves, one can determine the admissible convergence regions; as demonstrated by Liao in [24],

the valid region corresponds to the portion of the $\hbar$ -curve that remains parallel to the horizontal axis. Furthermore, the choice of the linear operator L improves the adaptability of the HAM. In this study, we consider the simplest case $L = I$; however, more sophisticated operators, such as the Laplace operator or other integral transforms, can be employed to obtain more accurate and efficient solutions. These flexibilities allow the HAM to handle nonlinear equations, integral equations, differential equations, integro-differential equations, and even systems with discontinuities or singularities [25]. It has been successfully applied in fluid dynamics, heat and mass transfer, nonlinear oscillations, wave propagation, biological systems, and financial modeling [26]. Its capability to provide accurate analytical approximations and to control the convergence region makes the HAM particularly useful in problems where purely numerical or perturbation approaches may fail or offer limited insight [27,28]. PINNs are one of the most popular research areas at this time, which allows us to integrate neural networks with the physical laws of the governing system, often expressed by differential equations or IEs [29]. This methodology creates a neural networks loss function from physical constraints. Later, we can minimize it, which makes the solution not only accurate but also physically consistent. Many researchers are working on improving numerical methods for different types of IEs, but still numerical methods are facing challenges with complex kernels, high dimensionality, and discontinuous kernels [30–32]. Neural networks allow us to solve these problems efficiently thanks to their universal approximation capability and by incorporating integral operators in the learning process. Numerous research studies demonstrate promising results in diverse fields, ranging from computational engineering and physics to finance and biology [29].

This paper is organized as follows: Section 2 provides the necessary preliminaries, where we recall fundamental concepts and notations and introduce the main idea of the HAM as the analytical foundation of our study. Section 3 is devoted to the main theoretical contributions, where we implement the HAM for solving first- and second-kind VIEs with discontinuous kernels and establish a rigorous convergence theorem to guarantee the validity of the proposed series solution. In Section 4, we present the framework of PINNs, define the augmented solution representation, and construct the residual and full loss functions that incorporate both data-driven learning and the governing equations. This section highlights the novelty of combining the HAM with PINNs to enhance accuracy and robustness. Section 5 contains several numerical examples, where the proposed hybrid HAM-PINN approach is applied to handle VIEs with discontinuities, and the results are compared with existing methods to demonstrate the efficiency, accuracy, and stability of the method. Finally, Section 6 concludes the paper with a summary of the main findings, discussion of the theoretical and practical implications, and suggestions for future research directions. The novelty of this work lies in the integration of an analytical series-based method (HAM) with a modern machine learning framework (PINNs) to develop a hybrid approach capable of solving discontinuous-kernel VIEs more effectively, while also providing theoretical convergence guarantees and practical numerical evidence of superiority over existing methods.

2. Preliminaries

Consider the nonlinear equation

$$N[\psi] = 0,$$

where N is a nonlinear operator and $\psi(t)$ is the unknown function to be determined. Based on the classical HAM, the corresponding zero-order deformation equation is formulated as

$$(1 - v)L[\Psi(t;v) - \psi_0(t)] = v\hbar H(t)N[\Psi(t;v)], \quad v \in [0, 1], \quad (3)$$

where $\hbar$ is the convergence-control parameter, L is an auxiliary linear operator, $\psi_0(t)$ is an initial guess for $\psi(t)$, $H(t) \neq 0$ is an auxiliary non-zero function, and $\Psi(t;v)$ is an unknown function that depends on the embedding parameter v .

Since $\hbar$ and $H(t)$ are non-zero, Equation (3) satisfies the boundary conditions

$$\begin{cases} \Psi(t;0) = \psi_0(t), \\ \Psi(t;1) = \psi(t), \end{cases} \quad (4)$$

indicating that as v continuously changes from 0 to 1, $\Psi(t;v)$ deforms from the initial approximation $\psi_0(t)$ to the exact solution $\psi(t)$. This continuous transformation is referred to as the homotopy deformation.

By applying Taylor series expansion around $v = 0$, the function $\Psi(t;v)$ can be expressed as

$$\Psi(t;v) = \psi_0(t) + \sum_{m=1}^{\infty} \psi_m(t)v^m, \quad (5)$$

where each component $\psi_m(t)$ is given by

$$\psi_m(t) = \frac{1}{m!} \left. \frac{\partial^m \Psi(t;v)}{\partial v^m} \right|_{v=0}.$$

If the auxiliary elements L , $\hbar$, $H(t)$, and $\psi_0(t)$ are chosen appropriately, the series (5) converges at $v = 1$, resulting in the solution

$$\psi(t) = \Psi(t;1) = \psi_0(t) + \sum_{m=1}^{\infty} \psi_m(t).$$

To systematically compute each term $\psi_m(t)$, we define the vector

$$\vec{\psi}_l(t) = \{\psi_0(t), \psi_1(t), \dots, \psi_l(t)\},$$

and derive the m -th order deformation equation by differentiating (3) m -times with respect to v , dividing by $m!$, and then setting $v = 0$. This yields

$$L[\psi_m(t) - \chi_m \psi_{m-1}(t)] = \hbar H(t) \mathcal{R}_m(\vec{\psi}_{m-1}(t)), \quad (6)$$

where the nonlinear operator $\mathcal{R}_m$ is defined as

$$\mathcal{R}_m(\vec{\psi}_{m-1}(t)) = \frac{1}{(m-1)!} \left. \frac{\partial^{m-1} N[\Psi(t;v)]}{\partial v^{m-1}} \right|_{v=0}, \quad (7)$$

and χ_m is given by

$$\chi_m = \begin{cases} 0, & m \leq 1, \\ 1, & m > 1. \end{cases} \quad (8)$$

Using the computed terms $\psi_m(t)$, the l -th order approximation of the exact solution is

$$\Psi_l(t) = \sum_{m=0}^l \psi_m(t),$$

and the exact solution is obtained as the limit

$$\psi(t) = \lim_{l \rightarrow \infty} \Psi_l(t).$$

Therefore, the accuracy and convergence of the solution heavily depend on the appropriate choice of auxiliary elements L , $\hbar$, $H(t)$, and the initial guess $\psi_0(t)$.

3. Main Idea

First, we focus on solving the IE (2) when $\lambda = 0$. It means we have a first-kind IE

$$\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) F(u(t)) dt = f(x), \quad (9)$$

and if for the nonlinear term, we have $F(u(t)) = U(t)$, then we get a linear problem like

$$\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U(t) dt = f(x). \quad (10)$$

We can define the operator N as follows:

$$N[u] = \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U(t) dt - f(x).$$

The n -th order deformation equation of the HAM can be expressed as

$$L \left[U_n(x) - \chi_n U_{n-1}(x) \right] = \hbar H(x) [\Re_n(U_{n-1}(x))], \quad (11)$$

where

$$\Re_n(U_{n-1}(x)) = \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_{n-1}(t) dt - (1 - \chi_n) f(x). \quad (12)$$

Substituting Equation (12) into Equation (11), we get

$$L \left[U_n(x) - \chi_n U_{n-1}(x) \right] = \hbar H(x) \left[\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_{n-1}(t) dt - (1 - \chi_n) f(x) \right]. \quad (13)$$

We know that, having freedom to choose the special parameters, operators, and functions of the HAM like L , $H(x)$, and $U_0(x)$ is one of the advantages of the HAM in comparison to the other semi-analytical methods. By taking $L = I$, $H(x) = 1$ in Equation (13) the following successive formula can be obtained:

$$\begin{cases} U_0(x) = \hbar(-f(x)), \\ U_1(x) = \hbar \left[\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_0(t) dt - f(x) \right], \\ U_2(x) = U_1(x) + \hbar \left[\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_1(t) dt \right], \\ U_n(x) = U_{n-1}(x) + \hbar \left[\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_{n-1}(t) dt \right]. \end{cases} \quad (14)$$

Applying the successive Formula (14) leads to the M -th order approximate solution of IE (10) using

$$U_M(x) = \sum_{r=0}^M U_r(x). \quad (15)$$

Then, applying the inverse substitution $u(t) = F^{-1}(U(t))$, we find the approximate solution of the nonlinear problem (9).

For the second-kind problem, we have $\lambda = 1$ in Equation (2) and we get

$$u(x) = f(x) + \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) F(u(t)) dt \quad (16)$$

and by the same transformation, we can change the problem to the linear form as

$$U(x) = f(x) + \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U(t) dt. \quad (17)$$

Repeating the same process as for the first-kind problem, we have the following successive relations:

$$\left\{ \begin{array}{l} U_0(x) = \hbar[-f(x)] \\ U_1(x) = \hbar \left[U_0(x) - \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_0(t) dt - f(x) \right] \\ U_2(x) = U_1(x) + \hbar \left[U_1(x) - \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_1(t) dt \right] \\ \vdots \\ U_n(x) = U_{n-1}(x) + \hbar \left[U_{n-1}(x) - \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_{n-1}(t) dt \right] \end{array} \right. \quad (18)$$

and the approximate solution can be found using Equation (15).

It is important to note that the presence of discontinuities in the kernel does not introduce instability in our solution process. This is because the integral equation is explicitly partitioned into subintervals at the discontinuity locations, as shown in Equation (1). Each subinterval is handled separately, which avoids the need for additional “gluing” conditions or smoothness assumptions at the discontinuity points. Consequently, the homotopy analysis framework applies uniformly across all subdomains, and the convergence behavior remains stable near the discontinuities. Our numerical results further confirm that the method produces bounded errors in the vicinity of discontinuity points, demonstrating that they do not disrupt the solution process.

The following flowchart (Figure 1) shows the HAM procedure for solving IEs.

Theorem 1. *If the solution of problem (2) is obtained using successive relations (14) and (20), and series (15), then the HAM converges to the exact solution.*

Proof. The series solution (16) can be written as follows:

$$U(x) = \sum_{r=0}^{\infty} U_r(x), \quad (19)$$

where $\lim_{r \rightarrow \infty} U_r(x) = 0$. Using Equation (6), we have

$$\sum_{m=1}^n [U_m(x) - \chi_m U_{m-1}(x)] = U_1 + (U_2 - U_1) + \dots + (U_n - U_{n-1}) = U_n(x), \quad (20)$$

and for the obtained equation, we get

$$\sum_{m=1}^n [U_m(x) - \chi_m U_{m-1}(x)] = \lim_{m \rightarrow \infty} U_m(x) = 0. \quad (21)$$

Applying Equation (21) and using the definition of the linear operator L , we have

$$\sum_{m=1}^{\infty} L[U_m(x) - \chi_m U_{m-1}(x)] = L\left[\sum_{m=1}^{\infty} (U_m(x) - \chi_m U_{m-1}(x))\right] = 0$$

Thus,

$$\sum_{m=1}^{\infty} L[U_m(x) - \chi_m U_{m-1}(x)] = \hbar H(x) \sum_{m=1}^{\infty} \Re_m(\vec{U}_{m-1}(x)) = 0.$$

In the obtained formula $\hbar, H(x) \neq 0$. So,

$$\sum_{m=1}^{\infty} \Re_m(\vec{U}_{m-1}(x)) = 0.$$

Now, we can write

$$\begin{aligned} \sum_{m=1}^{\infty} \Re_m(\vec{x}_{m-1}(x)) &= \sum_{m=1}^{\infty} \left[\sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U_{m-1}(t) dt - (1 - \chi_m) f(x) \right] \\ &= \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) \sum_{m=1}^{\infty} U_{m-1}(t) dt - f(x) \\ &= \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) \sum_{m=0}^{\infty} U_m(t) dt - f(x) \\ &= \sum_{i=1}^m \int_{\rho_{i-1}(x)}^{\rho_i(x)} k_i(x, t) U(t) dt - f(x), \end{aligned} \quad (22)$$

which shows that the series solution (19) must be the exact solution as that of problem (2).

In the HAM, $\hbar$ is known as the deformation parameter. Choosing an optimal $\hbar$ is very important to control error. We can choose an optimal $\hbar$ from the $\hbar$ curve where it is flat in our numerical illustration. $\square$

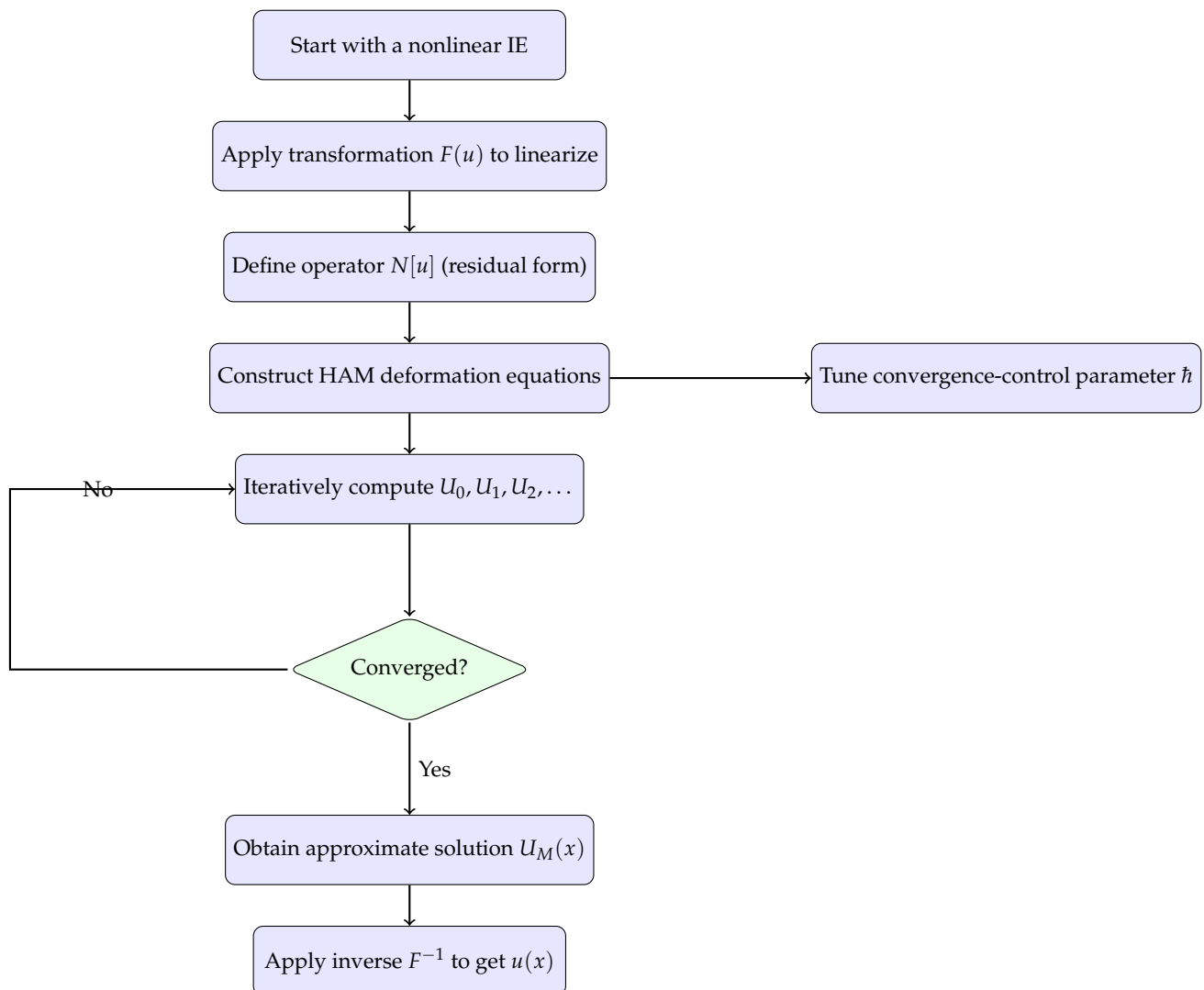


Figure 1. Flowchart of the HAM procedure for solving nonlinear integral equations.

4. Physics-Informed Neural Networks Integration

The proposed method integrates the HAM with PINNs. The combined algorithm is more powerful to capture the accurate solution pattern. The HAM captures the fundamental behavior of the solution, while neural networks make it more refined to the exact solution by minimizing the loss function of the governing integral equation.

The neural network of HAM-PINN is a fully connected feed-forward network (Figure 2). The network consists of a single input layer, three hidden layers, and one output layer. In the input layer, there is only a single neuron, which takes the independent variable t from $[0, 1]$. In the three hidden layers, there are 40 neurons, which are passed through the hyperbolic tangent activation function, which normalizes the output to the range $[-1, 1]$. The final layer is the output layer with the single neuron, which approximates the overall solution. To optimize the parameters, we use the Adam optimizer with an initial learning rate of 10^{-3} , and the model is trained for a total of 3000 epochs.

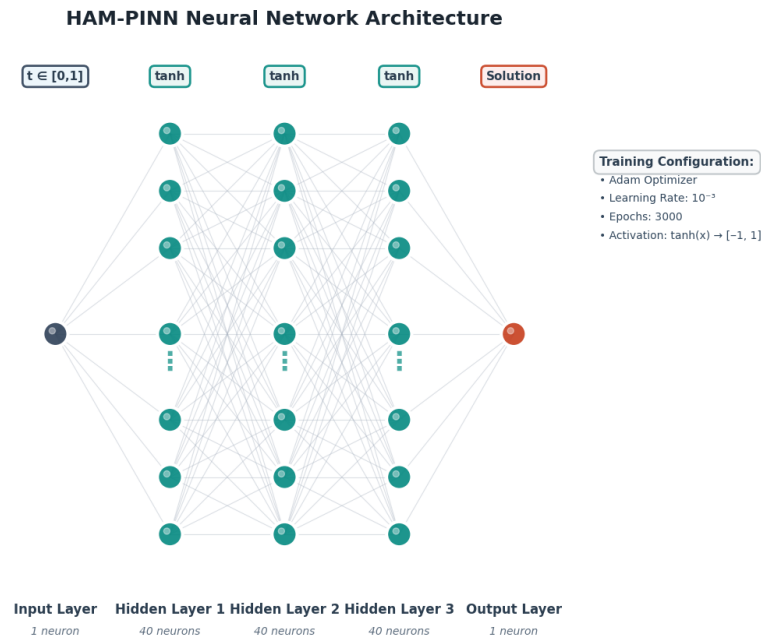


Figure 2. Schematic flow.

4.1. Augmented Solution

The total approximate solution $u_{\text{total}}(x)$ is the solution from the HAM series with optimized order and optimized h . After all, the additional neural network to the approximate solution is the extra correction term, which allows us to obtain a more refined result.

$$u_{\text{total}}(x; \theta) = u_{\text{HAM}}^{(N)}(x) + \mathcal{N}_{\theta}(x), \quad (23)$$

where $u_{\text{HAM}}^{(N)}(x)$ represents the sum of the first N terms of the HAM series for a specific or most optimized value of h , which is pre-computed and fixed during the neural networks training. $\mathcal{N}_{\theta}(x)$ denotes the output of a feed-forward neural networks where the goal is to achieve the most optimized θ , which minimizes the loss function, designed to approximate the remaining part of the solution.

4.2. PINN Residual Loss

The main idea of constructing any PINN algorithm lies in designing a physics-informed loss function of the corresponding governing equation of the system. For a VIE of the form (2), the residual loss function is the square of the difference between the left and right sides of the set of points for the integral equation.

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \left[u_{\text{total}}(x_i; \theta) - f(x_i) - \lambda \sum_{j=1}^{m+1} \mathcal{I}_j^{\text{total}}(x_i; \theta) \right]^2, \quad (24)$$

where

- N_c is the number of points x_i sampled uniformly or randomly across the domain.
- $u_{\text{total}}(x_i)$ is the augmented solution evaluated at x_i .
- $f(x_i)$ is the forcing term evaluated at x_i .
- λ is a constant coefficient from the integral equation.
- $\mathcal{I}_j^{\text{total}}(x_i)$ represents the numerical approximation of the j -th segment of the integral term at x_i . The integral is often split into multiple segments, especially for discontinuous kernels, as seen in the provided Python 3.12.11 code where the kernel changes over different sub-intervals.

Here, the numerical integration for each segment $\mathcal{I}_j^{\text{total}}(x_i)$ is performed using a quadrature rule over a grid of integration points t_k :

$$\mathcal{I}_j^{\text{total}}(x_i) \approx \sum_{k \in \mathcal{S}_j(x_i)} K_j(x_i, t_k, u_{\text{total}}(t_k)) w_k, \quad (25)$$

where $K_j(x_i, t_k, u_{\text{total}}(t_k))$ is the kernel function specific to the j -th integral segment, evaluated at x_i and integration point t_k with the current estimate of the total solution $u_{\text{total}}(t_k)$. w_k is the corresponding quadrature weights, and $\mathcal{S}_j(x_i)$ denotes the set of integration points t_k that fall within the bounds of the j -th integral segment for a given x_i .

4.3. Full Loss Function

The primary focus in PINNs is to minimize the physics-informed residual. PINN frameworks often include regularization terms to remove overfitting and improve training stability and generalization.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{physics}} + \beta_1 \mathcal{L}_{\text{regularization}}, \quad (26)$$

where $\mathcal{L}_{\text{regularization}}$ in this context refers to the sum of the weighted squared terms of the parameter θ . β_1 is a hyperparameter that controls the weighting of the regularization term relative to the physics loss. The training process focuses on optimizing θ to minimize this $\mathcal{L}_{\text{total}}$.

5. Numerical Illustration

In this section, we present several numerical examples involving linear and nonlinear VIEs of both the first and second kinds, characterized by discontinuous kernels. The proposed HAM-PINN method is employed to solve these problems. Our training strategy includes two hidden layers with the 40 neurons, and we used the following relation

$$\tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

as the activation function [33]. We take up to 200 equally independent points x .

Also, in order to compare the accuracy of the method, we focus on three metrics [34], including MSE (Mean Squared Error), MAE (Mean Absolute Error), and R-squared, as follows:

$$MAE = \frac{1}{K} \sum_{i=1}^K |u_i - \hat{u}_i|,$$

$$MSE = \frac{1}{K} \sum_{i=1}^K (u_i - \hat{u}_i)^2,$$

$$R^2 = 1 - \frac{\sum_{i=1}^K (u_i - \hat{u}_i)^2}{\sum_{i=1}^K (u_i - \bar{u})^2},$$

where u_i is to show the obtained numerical solution, $\hat{u}_i$ is the obtained solution value by PINN, $\bar{u}$ is the average value of all the solution values obtained using the numerical method, and K is the number of time data points. The results are summarized through tables comparing the approximate solutions with the exact ones at various values of t , including the corresponding errors. In addition, graphical illustrations such as the comparison of

exact and approximate solutions, loss curves across training epochs, error graphs, and the $\hbar$ -curves are provided for deeper insight.

Example 1. Consider the following linear Volterra integral equation of the first kind:

$$\int_0^{\frac{x}{4}} (1+x+t) u(t) dt + \int_{\frac{x}{4}}^{\frac{x}{2}} (2+xt) u(t) dt + \int_{\frac{x}{2}}^x (1+x+t) u(t) dt = \frac{271x^4}{8192} + \frac{1099x^5}{20480} + \frac{31x^6}{40960},$$

where the exact solution is $u(x) = \frac{x^3}{8}$. The HAM-PINN method is applied to solve this equation, using $\hbar = -1.3$, $M = 15$ and training the neural networks for 3000 epochs.

Table 1 presents the error metrics for the HAM-PINN solution in Example 1, including the MAE, MSE, and cumulative R^2 values. The MAE remains very small across all time points, reaching a maximum of approximately 3.46×10^{-3} at $t = 1.0$, indicating that the predicted solution closely aligns with the exact solution. The MSE values are consistently negligible, confirming the stability of the approximation. The cumulative R^2 values are all near 1, reflecting strong agreement between the HAM-PINN predictions and the reference solution, except at the initial point where R^2 is undefined. Overall, the results indicate that HAM-PINN provides accurate and reliable approximations for this example without significant variation in the error measures across the interval.

Table 1. Comparison between different errors for HAM-PINN in Example 1.

t	MAE	MSE	R^2 (Cumulative)
0.000000	0.000000	0.000000	NaN
0.250000	0.000085	0.000000	0.996288
0.500000	0.000123	0.000000	0.999850
0.750000	0.001747	0.000003	0.998266
1.000000	0.003464	0.000012	0.998630

Table 2 compares the exact solution with the approximate solutions obtained using HAM-PINN and HAM for Example 1, along with their corresponding absolute errors. Overall, HAM-PINN produces results that closely align with the exact solution at all time points, with absolute errors generally on the order of 10^{-3} or smaller. In comparison, HAM also provides reasonable approximations, but the errors are slightly higher at intermediate points, such as at $t = 0.5$ and $t = 0.75$, where HAM-PINN shows improved accuracy. At $t = 1.0$, HAM achieves a marginally smaller error than HAM-PINN; however, across the majority of the interval, HAM-PINN demonstrates more consistent performance. These results indicate that the hybrid HAM-PINN approach maintains a stable and reliable approximation throughout the domain while effectively handling variations in the solution.

Table 2. Comparison between the exact and approximate solutions and absolute errors in Example 1.

t	$u(t)$	$u_{15}(t)$ HAM-PINN	$u_{15}(t)$ HAM	$E_{15}(t)$ HAM-PINN	$E_{15}(t)$ HAM
0.00	0.000000	0.000000	0.000000	0.000000	0.000000
0.25	0.001983	0.001897	0.001716	0.000085	0.000267
0.50	0.015862	0.015985	0.016755	0.000123	0.000894
0.75	0.052470	0.050723	0.049975	0.001747	0.002494
1.00	0.125000	0.128464	0.125413	0.003464	0.000413

Figure 3a shows the comparative graphs between the approximate and exact solutions. Figure 3b contains the graphs of the absolute errors for HAM and PINN, which shows that PINN

has better and accurate results against normal HAM. Figure 3c illustrates the evolution of the loss function during training, highlighting how the networks parameters θ are optimized to minimize the error and enhance model performance.

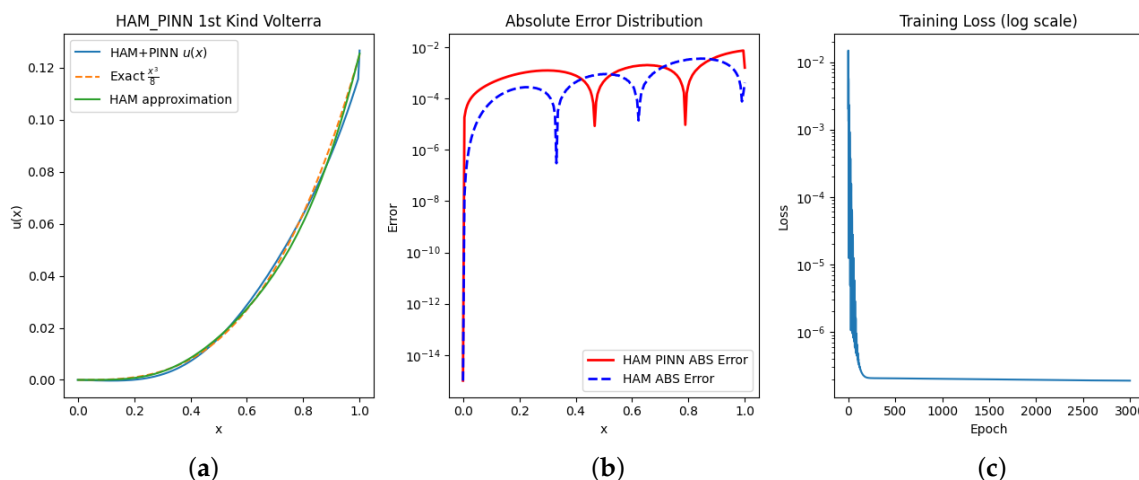


Figure 3. (a) Exact and approximate solutions. (b) Absolute error functions. (c) Loss vs. epoch.

Figure 4 illustrates the convergence region of the auxiliary parameter $\hbar$, which plays a crucial role in ensuring the validity of the HAM. By plotting the so-called $\hbar$ -curves, we are able to identify the intervals where the approximate solution converges to the exact one. For this particular example, the convergence region is observed within the interval $-2 \leq \hbar \leq -1$. Inside this range, the series solution generated by the HAM remains stable and reliable, whereas values of $\hbar$ outside this interval lead to divergence or loss of accuracy. This clearly demonstrates the importance of the convergence-control parameter in the HAM, since it not only guarantees convergence but also provides flexibility in adjusting the solution process without altering the original problem formulation.

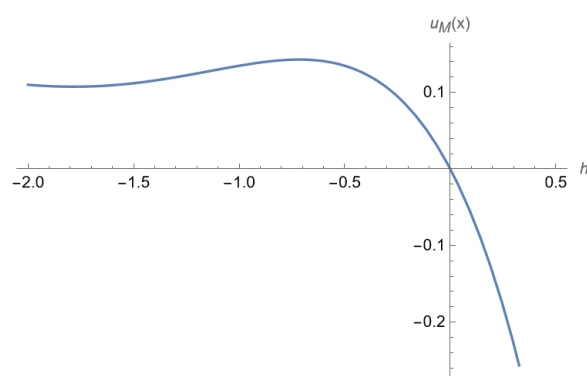


Figure 4. $\hbar$ -curve for Example 1 for $M = 15$.

Example 2. Consider the following Volterra integral equation of the first kind:

$$\int_0^{\frac{x}{3}} (t-1)u(t)^2 dt + \int_{\frac{x}{3}}^x x u(t)^2 dt = -\frac{x^9}{177147} + \frac{196823 x^{10}}{1771470},$$

where the exact solution is given by $u(x) = x^4$. This equation is characterized by a linear structure and a discontinuous kernel. To solve this problem, we applied a hybrid approach combining the PINN method with the HAM, using $\hbar = -1$, $M = 20$ and training the neural networks up to 3000 epochs.

Table 3 presents the error analysis of the HAM-PINN method for Example 2. At $t = 0$, all error measures are zero, as expected from the initial condition. For $t = 0.25$, the method shows a relatively small mean absolute error (MAE = 0.002350), though the R^2 value (0.304838) indicates limited

agreement with the exact solution at this early stage. As t increases to 0.50 and 0.75, the errors grow moderately ($MAE = 0.016805$ and 0.022351), but the R^2 values improve significantly, suggesting that the approximation follows the overall trend of the exact solution more closely. Finally, at $t = 1.0$, the method achieves a very low error ($MAE = 0.001786$) with an R^2 of 0.998903, showing strong agreement at the final point.

Table 3. Comparison between different errors for HAM-PINN in Example 2.

t	MAE	MSE	R^2 (Cumulative)
0.00	0.000000	0.000000	NaN
0.25	0.002350	0.000006	0.304838
0.50	0.016805	0.000282	0.887166
0.75	0.022351	0.000500	0.988135
1.00	0.001786	0.000003	0.998903

Table 4 compares the exact solution with the HAM-PINN and HAM approximations for Example 2, together with their absolute errors. At $t = 0$, all methods start from the exact value, giving zero error, as expected. For $t = 0.25$ and $t = 0.50$, both HAM-PINN and HAM provide close approximations, though HAM-PINN shows slightly smaller errors (0.002350 and 0.016805) compared to the HAM (0.002504 and 0.020320). At $t = 0.75$, the HAM-PINN error is 0.022351, which remains lower than the HAM's error of 0.035844, indicating a closer alignment with the exact solution in this interval. Finally, at $t = 1.00$, HAM-PINN achieves a very small error of 0.001786, while the HAM has a larger deviation of 0.012523.

Table 4. Comparison between the exact and approximate solutions and absolute errors in Example 2.

t	$u(t)$	$u_{20}(t)$ HAM-PINN	$u_{20}(t)$ HAM	$E_{20}(t)$ HAM-PINN	$E_{20}(t)$ HAM
0.00	0.000000	0.000000	0.000000	0.000000	0.000000
0.25	0.003985	0.001636	0.001482	0.002350	0.002504
0.50	0.063766	0.046960	0.043446	0.016805	0.020320
0.75	0.314292	0.291940	0.278448	0.022351	0.035844
1.00	1.000000	1.001786	0.987477	0.001786	0.012523

Figure 5 illustrates different aspects of the results. In panel (a), the solution obtained by the HAM-PINN method shows a close agreement with the exact solution. Panel (b) presents a comparison of the errors from the HAM and HAM-PINN, highlighting their relative accuracy. Panel (c) depicts the convergence trend of the loss function during training, reflecting how the optimization of the neural network parameters θ contributes to improving the solution accuracy.

Figure 6 presents the admissible interval of the convergence-control parameter $\hbar$ for this example. The results indicate that the convergence region is obtained within the range $-1.5 \leq \hbar \leq -1$ for $M = 20$. This interval corresponds to the nearly horizontal portion of the h -curve, which is known to characterize the region where the HAM series solution remains valid and reliable. Identifying such a range is essential, since values of $\hbar$ outside this interval may lead to divergence or reduced accuracy of the approximation.

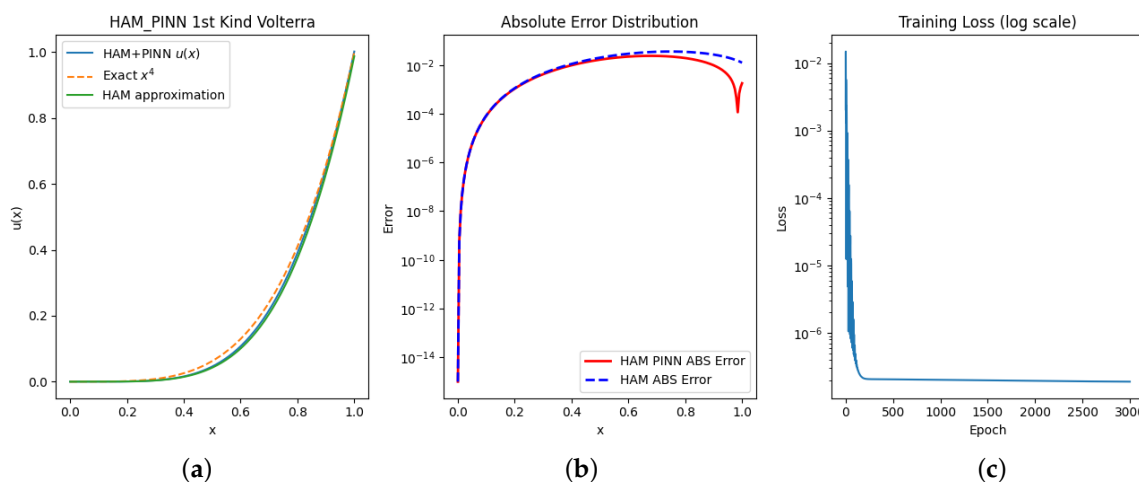


Figure 5. (a) Exact and approximate solutions. (b) Absolute error functions. (c) Loss vs. epoch.

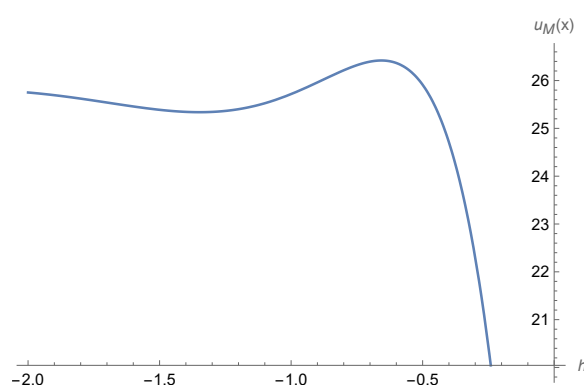


Figure 6. Convergence region for h in Example 3.

Example 3. Consider the following nonlinear VIE

$$u(x) = f(x) + \int_0^{x/2} x u(t)^2 dt + \int_{x/2}^x (x-t) u(t) dt, \quad (27)$$

with the exact solution $u(x) = e^x$ where the source term is

$$f(x) = \frac{1}{2} \left(x - e^x x + e^{x/2} (2 + x) \right)$$

where the exact solution is $u(x) = e^x$. Here, we show our result after combining the neural networks training up to 3000 epochs and the HAM for $h = -1$ and $M = 10$.

Table 5 presents the error metrics for HAM-PINN in Example 3. At $t = 0$, the MSE is zero. For $t = 0.25$ and $t = 0.50$, the mean absolute error remains small (0.001459 and 0.002712, respectively), with corresponding MSE values on the order of 10^{-6} to 10^{-7} . At later points, $t = 0.75$ and $t = 1.00$, the errors increase slightly, reaching a maximum MAE of 0.013723. Despite this increase, the R^2 values remain close to unity (above 0.9998), suggesting that the HAM-PINN solution maintains a strong agreement with the exact solution throughout the interval. The results indicate stable performance of the method, with errors growing moderately as t increases.

Table 6 compares the exact solution with the approximations obtained by HAM-PINN and pure HAM in Example 3, together with their absolute errors. At $t = 0$, both methods reproduce the initial condition, but HAM-PINN achieves a smaller error (0.000387) compared to the HAM (0.010050). As t increases, HAM-PINN consistently provides lower errors than HAM. For instance, at $t = 0.25$ and $t = 0.50$, the absolute errors of HAM-PINN are 0.001459 and 0.002712, respectively, while the HAM errors are noticeably larger (0.008264 and 0.011896). At later points ($t = 0.75$ and $t = 1.00$),

the difference becomes more pronounced: HAM-PINN maintains errors below 0.014, whereas the HAM errors increase to 0.035116 and 0.094190. The results indicate that HAM-PINN achieves closer agreement with the exact solution across the whole interval, with a clear improvement in accuracy compared to the HAM, especially at larger values of t .

Table 5. Comparison between different errors for HAM-PINN in Example 3.

t	MAE	MSE	R^2 (Cumulative)
0.00	0.000387	0.000000	NaN
0.25	0.001459	0.000002	0.999939
0.50	0.002712	0.000007	0.999953
0.75	0.009643	0.000093	0.999851
1.00	0.013723	0.000188	0.999842

Table 6. Comparison between the exact and approximate solutions and absolute errors in Example 3.

t	$u(t)$	$u_{10}(t)$ HAM-PINN	$u_{10}(t)$ HAM	$E_{10}(t)$ HAM-PINN	$E_{10}(t)$ HAM
0.000000	1.010050	1.009663	1.000000	0.000387	0.010050
0.250000	1.284025	1.282567	1.275761	0.001459	0.008264
0.500000	1.648721	1.646009	1.636825	0.002712	0.011896
0.750000	2.117000	2.107357	2.081884	0.009643	0.035116
1.000000	2.718282	2.704559	2.624091	0.013723	0.094190

Figure 7a shows that the HAM-PINN approach closely aligned with the exact solution, Figure 7b gives a comparison between the errors, and Figure 7c demonstrates how the loss function is optimized per epoch to obtain the promising result by determining θ , which is the parameter of the neural networks.

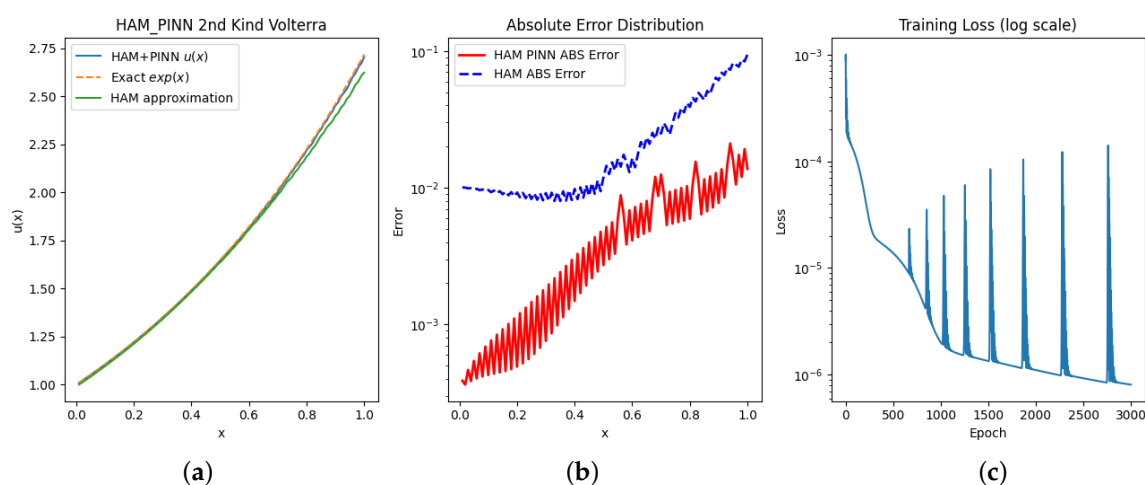


Figure 7. (a) Exact and approximate solutions. (b) Absolute error functions. (c) Loss vs. epoch.

Figure 8 shows the $\hbar$ -curve, illustrating the range of auxiliary parameter $\hbar$ for which the HAM solution converges. For this particular example, convergence is achieved when $\hbar$ lies within the interval $-1.5 \leq \hbar \leq -0.5$. Within this range, the curve exhibits a nearly flat portion that runs parallel to the x -axis, which defines the stable convergence region. This parallel segment indicates that small variations in $\hbar$ do not significantly affect the accuracy of the solution, providing flexibility in selecting the parameter. Outside this interval, the curve deviates sharply, suggesting that the

solution may diverge or lose accuracy. Therefore, choosing $\hbar$ from the parallel region ensures both convergence and numerical stability of the HAM approximation.

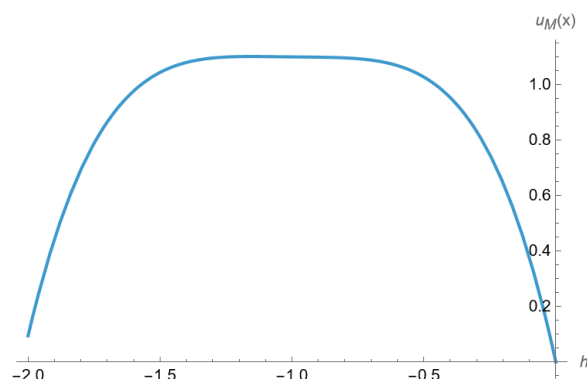


Figure 8. $\hbar$ Curve of Example 3 for $M = 10$.

Example 4. Consider the integral equation:

$$u(x) = f(x) + \int_0^{x/8} 2x u(t) dt + \int_{x/8}^{3x/8} (x-t) u(t) dt + \int_{3x/8}^x u(t) dt$$

with

$$f(x) = -x + \frac{55x^2}{128} + \frac{815x^3}{768} - \frac{4015x^4}{16384} - \frac{289x^5}{8192}$$

where the exact solution is $u(x) = x^3 - x$. This is a linear Volterra IE of the second kind with a discontinuous kernel.

Table 7 presents the error metrics for the HAM-PINN solution in Example 4, including MAE, MSE, and cumulative R^2 values. The MAE remains consistently small across all time points, with the highest value being 1.702×10^{-3} at $t = 1.0$, indicating that the predicted solution closely follows the exact solution. The MSE values are negligible throughout, reflecting the stability and consistency of the method. The cumulative R^2 values are all very close to 1, confirming strong agreement between the HAM-PINN predictions and the reference solution, except at the initial point where R^2 is undefined. The results demonstrate that HAM-PINN provides accurate and reliable approximations with stable performance over the entire interval.

Table 7. Comparison between different errors for HAM-PINN in Example 4.

t	MAE	MSE	R^2 (Cumulative)
0.00	0.000271	0.000000	NaN
0.25	0.001269	0.000002	0.999939
0.50	0.000983	0.000001	0.999963
0.75	0.001304	0.000002	0.999948
1.00	0.001702	0.000003	0.999943

Table 8 compares the exact solution with the approximate solutions obtained using HAM-PINN and HAM, along with their absolute errors. The HAM-PINN method maintains a close alignment with the exact solution across all time points, with absolute errors ranging from 6.0×10^{-5} at $t = 0$ to 2.003×10^{-3} at $t = 1.0$. In comparison, the HAM shows slightly larger deviations, particularly at $t = 0.75$ and $t = 1.0$, where the absolute error reaches 6.965×10^{-3} and 2.0265×10^{-2} , respectively. These results indicate that HAM-PINN provides more consistent and accurate approximations, especially in regions affected by rapid changes or near the endpoints, while HAM performs adequately in smoother regions but exhibits larger errors in critical points.

Table 8. Comparison between the exact and approximate solutions and absolute errors in Example 4.

t	$u(t)$	$u_{10}(t)$ HAM-PINN	$u_{10}(t)$ HAM	$E_{10}(t)$ HAM-PINN	$E_{10}(t)$ HAM
0.00	0.000000	0.000060	0.000000	0.000060	0.000000
0.25	−0.235395	−0.236284	−0.235734	0.000890	0.000339
0.50	−0.375619	−0.376198	−0.374198	0.000579	0.001421
0.75	−0.328985	−0.329910	−0.322020	0.000925	0.006965
1.00	0.000000	0.002003	0.020265	0.002003	0.020265

Figure 9 illustrates different aspects of the HAM-PINN performance. Figure 9a shows that the HAM-PINN solution is closely aligned with the exact solution across the domain. Figure 9b presents a comparison of the absolute errors obtained from HAM-PINNs and pure HAM, highlighting the improved accuracy of the hybrid approach. Finally, Figure 9c displays the evolution of the loss function during training, demonstrating how the neural network parameters θ are progressively optimized to enhance solution accuracy.

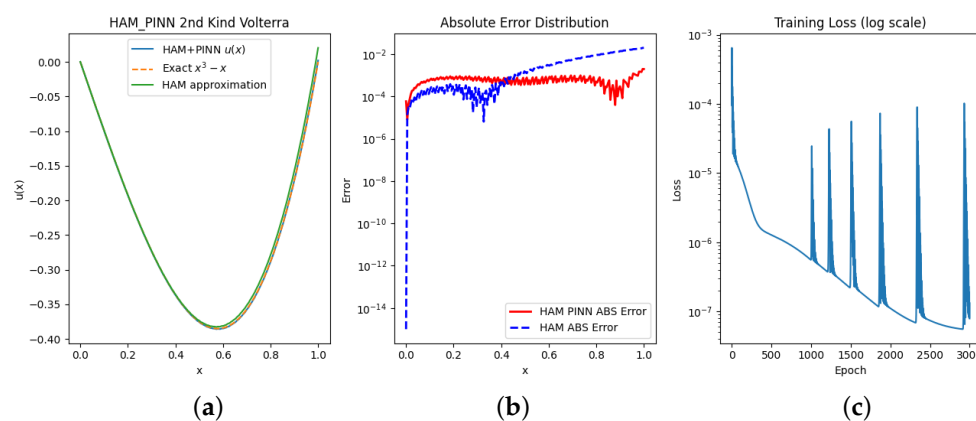
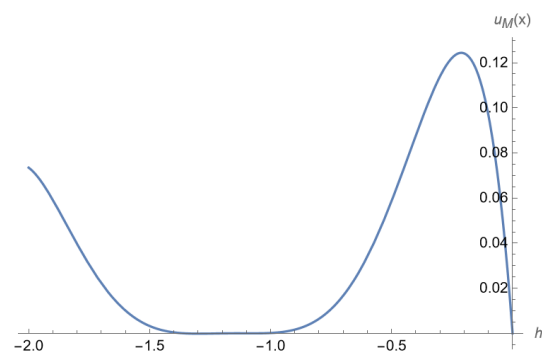

Figure 9. (a) Exact and approximate solutions. (b) Absolute error functions. (c) Loss vs. epoch.

Figure 10 illustrates the admissible range of values for the auxiliary parameter $\hbar$ that ensures a convergent solution. For this problem, convergence is obtained when $-1.5 \leq \hbar \leq -1$. Within this interval, the $\hbar$ -curve exhibits a nearly flat portion that runs parallel to the x -axis, indicating a stable convergence region. This parallel behavior suggests that the solution remains insensitive to small variations in $\hbar$, providing flexibility in its selection without compromising accuracy. Outside this range, however, the curve deviates, leading to divergence and loss of reliability in the approximation. Hence, choosing $\hbar$ from the parallel region ensures both convergence and numerical stability of the solution.


Figure 10. $\hbar$ curve Example 4 for $M = 10$.

6. Conclusions

In this study, we investigated the numerical solution of first- and second-kind VIEs with discontinuous kernels, which are fundamental in modeling systems with memory effects, such as load leveling in battery energy systems. The discontinuities in the kernels in the load leveling problem introduce significant challenges, requiring robust numerical strategies. To address this, we proposed a hybrid framework that combines the HAM, with its analytical flexibility and adjustable convergence control, and PINNs, which leverage data-driven learning guided by physical laws.

The comparative analysis between the HAM and HAM-PINNs demonstrates that the hybrid method achieves improvements in accuracy. For example, the absolute errors in HAM-PINNs were consistently lower than those of the HAM alone. However, these improvements come at the cost of additional computational effort due to the training phase of PINNs, which requires optimization over neural network parameters. Despite this, the enhanced accuracy and stability make HAM-PINNs a promising trade-off between precision and computational cost.

When compared with existing methods in the literature, the HAM-PINN approach demonstrates strong competitiveness, yielding results that are consistently closer to the exact solutions across various test problems. One of the main advantages of the HAM-based family of methods is the presence of auxiliary functions, auxiliary parameters, and the freedom in selecting a linear operator. These features provide a high degree of flexibility and control over the convergence of the series solution, which is not available in most traditional methods. In particular, the $\hbar$ -curve offers a practical mechanism for identifying convergence regions and ensuring solution stability, allowing the method to be systematically tuned for improved accuracy.

Several limitations of the proposed method should also be noted. The scalability of HAM-PINNs to high-dimensional domains remains challenging due to increased computational demand in training PINNs. Additionally, while the current study focused primarily on linear examples, future work should address nonlinear integral equations and coupled systems to further validate the robustness of the framework.

Looking ahead, future research directions include (i) extending the approach to stochastic kernels and uncertainty quantification, and (ii) developing adaptive real-time learning strategies for online control and energy management. These extensions would enhance the practicality and versatility of HAM-PINNs in solving real-world problems where memory effects and discontinuities play a dominant role.

Author Contributions: Methodology, S.N., M.A.M. and S.M.; Software, S.N. and M.A.M.; Validation, S.N., M.A.M. and S.M.; Formal analysis, S.N. and S.M.; Investigation, S.M.; Writing—original draft, S.N. and M.A.M.; Writing—review & editing, S.N., M.A.M. and S.M. All authors have read and agreed to the published version of the manuscript.

Funding: The work of S. Noeiaghdam was funded by the High-Level Talent Research Start-up Project Funding of Henan Academy of Sciences (Project No. 241819246).

Data Availability Statement: Data are contained within the article.

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

References

1. Kumar, R.; Kant, K.; Kumar, B.V.R. Superconvergent spectral methods for system of modified Volterra Integral Equations. *Math. Comput. Simul.* **2026**, *239*, 115–134. [[CrossRef](#)]
2. Chakraborty, S.; Agrawal, S.K.; Nelakanti, G. Spectral approximated superconvergent methods for system of nonlinear Volterra Hammerstein integral equations. *Chaos Solitons Fractals* **2025**, *192*, 116008. [[CrossRef](#)]

3. Abdi, A.; Hormann, K.; Hosseini, S.A. Linear barycentric rational Hermite quadrature and its application to Volterra integral equations. *J. Comput. Appl. Math.* **2026**, *474*, 117009. [\[CrossRef\]](#)
4. Bazm, S.; Lima, P.; Nemati, S. Discretization methods and their extrapolations for two-dimensional nonlinear Volterra–Urysohn integral equations. *Appl. Numer. Math.* **2025**, *208*, 323–337. [\[CrossRef\]](#)
5. Bahmanpour, M.; Araghi, M.A.F. Numerical solution of Fredholm and Volterra integral equations of the first kind using wavelets bases. *J. Math. Comput. Sci.* **2012**, *5*, 337–345. [\[CrossRef\]](#)
6. Araghi, M.A.F.; Behzadi, S.S. Solving nonlinear Volterra–Fredholm integro-differential equations using He’s variational iteration method. *Int. J. Comput. Math.* **2011**, *88*, 829–838. [\[CrossRef\]](#)
7. Fahim, A.; Araghi, M.A.F.; Rashidinia, J.; Jalalvand, M. Numerical solution of Volterra partial integro-differential equations based on sinc-collocation method. *Adv. Differ. Equ.* **2017**, *2017*, 362. [\[CrossRef\]](#)
8. Behzadi, S.S.; Abbasbandy, S.; Allahviranloo, T.; Yildirim, A. Application of homotopy analysis method for solving a class of nonlinear Volterra–Fredholm integro-differential equations. *J. Appl. Anal. Comput.* **2012**, *2*, 127–136.
9. Gouyandeh, Z.; Allahviranloo, T.; Armand, A. Numerical solution of nonlinear Volterra–Fredholm–Hammerstein integral equations via Tau-collocation method with convergence analysis. *J. Comput. Appl. Math.* **2016**, *308*, 435–446. [\[CrossRef\]](#)
10. Amirkhizi, S.A.; Mahmoudi, Y.; Shamloo, A.S. Solution of Volterra integral equations of the first kind with discontinuous kernels by using the Adomian decomposition method. *Comput. Methods Differ. Equ.* **2024**, *12*, 189–195.
11. Amoozad, T.; Allahviranloo, T.; Abbasbandy, S.; Malkhalifeh, M.R. Application of the reproducing kernel method for solving linear Volterra integral equations with variable coefficients. *Phys. Scr.* **2024**, *99*, 025246. [\[CrossRef\]](#)
12. Firouzidor, R.; Asari, S.S.; Amirfakhrian, M. Application of radial basis function to approximate functional integral equations. *J. Interpolat. Approx. Sci. Comput.* **2016**, *2*, 77–86. [\[CrossRef\]](#)
13. Jafarian, A.; Measoomy, S.; Abbasbandy, S. Artificial neural networks based modeling for solving Volterra integral equations system. *Appl. Soft Comput.* **2015**, *27*, 391–398. [\[CrossRef\]](#)
14. Sidorov, N.A.; Falaleev, M.V.; Sidorov, D.N. Generalized solutions of Volterra integral equations of the first kind. *Bull. Malays. Math. Sci. Soc.* **2006**, *28*, 101–109.
15. Sidorov, N.A.; Sidorov, D.N. Existence and construction of generalized solutions of nonlinear Volterra integral equations of the first kind. *Differ. Equ.* **2006**, *42*, 1312–1316. [\[CrossRef\]](#)
16. Sidorov, N.A.; Sidorov, D.N.; Krasnik, A.V. Solution of Volterra operator-integral equations in the nonregular case by the successive approximation method. *Differ. Equ.* **2010**, *46*, 882–891. [\[CrossRef\]](#)
17. Sidorov, D.N. On parametric families of solutions of Volterra integral equations of the first kind with piecewise smooth kernel. *Differ. Equ.* **2013**, *49*, 210–216. [\[CrossRef\]](#)
18. Sidorov, D. *Integral Dynamical Models: Singularities. Signals & Control*, T. 87; World Scientific Series on Nonlinear Science Series A; World Scientific Publications Pte Ltd.: Singapore, 2015.
19. Noeiaghdam, S.; Sidorov, D.; Dreglea, A. A novel numerical optimality technique to find the optimal results of Volterra integral equation of the second kind with discontinuous kernel. *Appl. Numer. Math.* **2023**, *186*, 202–212. [\[CrossRef\]](#)
20. Noeiaghdam, S.; Sidorov, D.; Muftahov, I.; Zhukov, A.V. Control of Accuracy on Taylor-Collocation Method for Load Leveling Problem. *Bull. Irkutsk. State Univ. Ser. Math.* **2019**, *30*, 59–72. [\[CrossRef\]](#)
21. Noeiaghdam, S.; Araghi, M.A.F. Homotopy regularization method to solve the singular Volterra integral equations of the first kind. *Jordan J. Math. Stat.* **2018**, *11*, 1–12.
22. Noeiaghdam, S.; Araghi, M.A.F. Valid implementation of the Sinc-collocation method to solve the linear integral equations by CADNA library. *J. Math. Model.* **2019**, *7*, 63–84.
23. Mikaeilvand, N.; Noeiaghdam, S. Mean value theorem for integrals and its application on numerically solving of Fredholm integral equation of second kind with Toeplitz plus Hankel Kernel. *Int. J. Ind. Math.* **2014**, *6*, 351–360.
24. Liao, S.J. The Proposed Homotopy Analysis Techniques for the Solution of Nonlinear Problems. Ph.D. Thesis, Shanghai Jiao Tong University, Shanghai, China, 1992. (In English)
25. Liao, S.J. *Beyond Perturbation: Introduction to Homotopy Analysis Method*; Chapman & Hall/CRC Press: Boca Raton, FL, USA, 2003.
26. Sahihi, H.; Abbasbandy, S.; Allahviranloo, T. Homotopy analysis method and its application for solving singularly perturbed differential–difference equation with boundary layer behavior and dela. *J. New Res. Math.* **2020**, *6*, 29–38.
27. Liao, S.J. On the homotopy analysis method for nonlinear problems. *Appl. Math. Comput.* **2004**, *147*, 499–513. [\[CrossRef\]](#)
28. Liao, S.J. *Homotopy Analysis Method in Nonlinear Differential Equations*; Higher Education Press: Beijing, China; Springer: Berlin/Heidelberg, Germany, 2012.
29. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
30. Mao, Z.; Jagtap, A.D.; Karniadakis, G.E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* **2020**, *360*, 112789. [\[CrossRef\]](#)

31. Effati, S.; Buzhabadi, R. A neural networks approach for solving Fredholm integral equations of the second kind. *Neural Comput. Appl.* **2012**, *21*, 843–852. [[CrossRef](#)]
32. Lu, L.; Meng, Y.; Mao, Z.; Wang, J.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations and integral equations. *SIAM J. Sci. Comput.* **2021**, *43*, S613–S642. [[CrossRef](#)]
33. Vo, V.T.; Noeiaghdam, S.; Sidorov, D.; Dreglea, A.; Wang, L. Solving Nonlinear Energy Supply and Demand System Using Physics-Informed Neural Networks. *Computation* **2025**, *13*, 13. [[CrossRef](#)]
34. Wang, X.; Liu, X.; Wang, Y.; Kang, X.; Geng, R.; Li, A.; Xiao, F.; Zhang, C.; Yan, D. Investigating the Deviation Between Prediction Accuracy Metrics and Control Performance Metrics in the Context of an Ice-Based Thermal Energy Storage System. *J. Energy Storage* **2024**, *91*, 112–126. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.