Analytical and Numerical Solutions for Nonlinear Equations
©Available online at https://ansne.du.ac.ir/
Online ISSN: 3060–785X
2027, Volume 12, Issue 2, pp. xx–xx
Research article

A Chebyshev-Enhanced Physics-Informed Neural Network for Duffing Equations

Fatemeh Nikzaddinan[1], Hassan Dana Mazraeh[1,*]

[1] School of Mathematics and Computer Science, Damghan University, Damghan, Iran

* **Corresponding author(s):** dana@du.ac.ir

Received: 16/10/2026 **Revised:** 12/11/2026 **Accepted:** 17/11/2026 **Published:** xx/xx/2026 10.22128/ansne.2026.3512.1251

Abstract

The Duffing equation is a fundamental model for nonlinear vibrations in mechanical structures, micro-electromechanical systems, and structural dynamics; however, obtaining accurate solutions under strong nonlinear conditions remains challenging. This paper proposes a Chebyshev-enhanced physics-informed neural network that replaces the conventional activation function in the second hidden layer with a Chebyshev polynomial basis to mitigate the spectral bias associated with oscillatory solutions. The proposed method is validated on four Duffing equation configurations, including damped and undamped, as well as forced and unforced cases, with both known and unknown analytical solutions.The Chebyshev polynomial degree is selected separately for each Duffing configuration, rather than being fixed at a single value across all examples.Numerical results demonstrate that the proposed Chebyshev-enhanced physics-informed neural network consistently outperforms conventional physics-informed neural networks, achieving one to three orders of magnitude improvements in solution accuracy and residual minimization. These findings establish Chebyshev polynomial activation as an effective enhancement for physics-informed neural networks and a promising approach for solving nonlinear dynamical systems encountered in engineering applications.

Keywords: Physics-informed neural networks, Chebyshev polynomial, Duffing equation, Nonlinear oscillation, Engineering

Mathematics Subject Classification (2020): 68T07, 34A34

1 Introduction

Ordinary differential equations (ODEs), particularly nonlinear dynamical systems, constitute one of the most important and computationally challenging mathematical frameworks in modern science and engineering applications. Such equations naturally arise in numerous disciplines, including nonlinear mechanics, fluid dynamics, control systems, vibration analysis, electrical circuits, biological processes, and applied physics, where accurate characterization of time-dependent dynamical behaviors is of considerable practical importance [1]. In contrast to linear models, nonlinear ODEs are capable of describing highly intricate phenomena such as nonlinear oscillations, instability, bifurcation, resonance, and chaotic dynamics that frequently emerge in realistic physical and engineering systems. Due to their broad applicability and significant practical relevance, nonlinear ODEs have attracted substantial attention in modern scientific computing and applied mathematics. However, despite considerable progress in analytical and numerical methodologies, obtaining exact closed-form

COPYRIGHTS: ©2027, Analytical and Numerical Solutions for Nonlinear Equations published by Damghan University.
This is an open access article published under the *CC BY 4 DEED* license

solutions for many nonlinear ODEs remains highly difficult or even impossible because of their inherent nonlinearity and complex dynamical structures [2,3]. This limitation has motivated increasing interest in the development of efficient and reliable approximation methods within scientific computing and modern computational mathematics.

As a canonical representative of the nonlinear dynamical systems described above, the Duffing equation has long been recognized as one of the most fundamental and representative nonlinear oscillators in applied mathematics, physics, and engineering. The equation provides a mathematical model for a wide range of nonlinear vibration phenomena arising in mechanical structures, electrical circuits, microelectromechanical systems, structural dynamics, nonlinear springs, aeroelastic systems, and nonlinear wave propagation. Owing to the presence of cubic nonlinear stiffness terms, the Duffing equation is capable of exhibiting rich and complex dynamical behaviors including hardening and softening responses, jump phenomena, hysteresis, bifurcation, resonance, quasi-periodicity, and chaotic motion [4,5]. These characteristics make the Duffing oscillator an important benchmark problem for evaluating the effectiveness, robustness, and accuracy of modern computational and numerical methodologies for nonlinear differential equations. Furthermore, while certain simplified forms of the Duffing equation admit exact analytical solutions, many practically relevant cases involving damping, external forcing, or strong nonlinearities do not possess closed-form analytical solutions and therefore require advanced approximation techniques. Consequently, the Duffing equation represents an excellent testbed for investigating the capabilities of Physics-Informed Neural Networks (PINNs) [6], particularly in assessing their ability to approximate highly nonlinear dynamical responses under both analytically solvable and analytically intractable conditions.

The computational intractability of exact solutions for nonlinear ODEs such as the Duffing equation has motivated the development of data-driven and machine learning approaches capable of capturing complex dynamical behaviors without relying on analytical closed forms. In this context, neural networks have emerged as powerful universal function approximators for nonlinear mappings, offering a mesh-independent and continuous representation of solution operators associated with differential equations [7, 8]. Their ability to approximate nonlinear, high-dimensional, and time-dependent functions makes neural networks particularly suitable for modeling complex dynamical systems, where classical numerical methods may suffer from discretization errors and computational inefficiency [9,10]. Building upon the universal approximation capability of neural networks, physics-informed neural networks, previously identified as a promising framework for the Duffing oscillator, offer a principled mechanism to embed the governing equations of nonlinear dynamical systems directly into the learning process. PINNs extend this paradigm by incorporating governing differential equations directly into the loss function, promoting physical consistency of the learned solutions. Nevertheless, the effectiveness of PINNs is still influenced by the complexity of the underlying optimization landscape in nonlinear regimes. In particular, insufficient representational capacity of the neural network may lead to increased approximation error and larger residual error in satisfying the governing differential equations, which can adversely affect both solution accuracy and physics consistency [11]. To address these limitations, recent studies have explored enhanced neural representations that improve approximation quality and training stability [12]. In this direction, structured basis-enhanced architectures have demonstrated promising capabilities in improving function representation within scientific machine learning frameworks. Motivated by this perspective, this study proposes a Chebyshev-enhanced physics-informed neural network for improved solution accuracy and convergence behavior in nonlinear dynamical systems.

While the limitations of standard PINNs have been partially addressed through enhanced architectures, as noted above, the role of activation function design in improving function representation for nonlinear and oscillatory ordinary differential equations remains relatively underexplored. In particular, most existing PINNs studies primarily focus on loss function design and training strategies, while representation-level improvements through activation function engineering have received limited attention. This limitation becomes more pronounced in nonlinear dynamical systems exhibiting strong oscillatory behavior, such as the Duffing equation, where accurate function representation is critical for capturing complex solution dynamics. Moreover, polynomial-based activations, such as Chebyshev polynomials, have rarely been investigated within PINNs frameworks for solving nonlinear ordinary differential equations with highly nonlinear and oscillatory responses [13].

Motivated by the theoretical gap identified above, this study proposes a Chebyshev polynomial-enhanced physics-informed neural network (ChebyPINN) for solving nonlinear ordinary differential equations, with focused application to the Duffing oscillator. The scientific rationale for integrating Chebyshev-based activation functions is grounded in classical approximation theory: Chebyshev polynomials of the first kind constitute a complete orthogonal basis on the interval $[-1,1]$ and are known to deliver near-minimax optimal polynomial approximation of smooth functions, achieving exponential convergence rates for analytic targets—a property that fundamentally outperforms standard monomial bases, which are susceptible to Runge's phenomenon under high-degree interpolation. Because the core

Article in press

objective of a PINN is to approximate the solution of a governing differential equation through residual minimization, the representational capacity of the underlying network architecture directly governs both solution accuracy and physics consistency. Replacing conventional pointwise activations—such as tanh or sigmoid—with structured Chebyshev basis functions therefore constitutes a theoretically principled enhancement to network expressivity, one that is particularly well-suited to functions exhibiting oscillatory and strongly nonlinear character, as commonly arise in the Duffing system. Furthermore, the bounded domain definition of Chebyshev polynomials on $[-1, 1]$ naturally constrains activation magnitudes during forward propagation, contributing to gradient stability and mitigating the stiffness-induced training pathologies frequently encountered when applying standard PINNs to nonlinear dynamical systems. The primary contribution of this work therefore lies in the systematic integration of Chebyshev activation mechanisms within a PINN residual minimization framework, and in the rigorous empirical evaluation of the resulting architecture across both analytically tractable and analytically intractable configurations of the Duffing equation. Numerical results demonstrate that ChebyPINN achieves measurably improved solution accuracy, reduced physics-informed residuals, and more stable convergence behavior relative to standard PINNs baselines, providing empirical evidence that activation function design constitutes an effective and theoretically grounded degree of freedom for scientific machine learning applied to nonlinear dynamical systems.

The remainder of this paper is organized as follows. Section 2 reviews related work on neural-network-based and physics-informed approaches for solving the Duffing equation, and identifies the limitations that motivate the present study. Section 3 presents the proposed methodology, beginning with the conventional PINN formulation for differential equations, followed by the mathematical foundations of Chebyshev polynomials, and finally the detailed architecture and training procedure of the proposed ChebyPINN. Section 4 reports the experimental results, in which the proposed architecture is evaluated on four representative Duffing oscillator configurations, including damped and undamped as well as forced and unforced cases, with both known and unknown analytical solutions, and compared against a conventional PINN baseline under identical training conditions. Section 5 discusses the interpretation of the numerical findings, the mechanisms underlying the observed improvements, and the sensitivity of the proposed model to key hyperparameters. Finally, Section 6 summarizes the main contributions and outlines directions for future work.

2 Related Works

Neural-network-based approaches for the Duffing–harmonic oscillator have been explored within metaheuristic optimization frameworks to construct approximate solution representations of the system dynamics. Khan et al. [14] proposed a hybrid computational scheme combining an artificial neural network with the firefly algorithm, in which the solution of the –harmonic oscillator is approximated through a parameterized neural representation whose weights are optimized via a population-based evolutionary search. In this formulation, the governing differential equation is transformed into an optimization problem by defining a residual-based objective function, while a cosine-based activation structure is employed to enhance the representation of oscillatory behavior. The firefly algorithm is then utilized to search globally for optimal network parameters that minimize the discrepancy between the neural approximation and the governing equation constraints. Although the method provides accurate approximate solutions and captures the frequencyamplitude relationship of the oscillator, it remains a metaheuristic neural approximation framework rather than a physics-constrained learning approach. Nevertheless, the study highlights the effectiveness of coupling neural function approximation with evolutionary optimization for nonlinear oscillatory systems such as the Duffing equation.

While these early studies relied heavily on evolutionary and metaheuristic optimization to tune neural representations, an alternative line of research explored structured approximation frameworks aimed at improving representational efficiency without explicitly embedding physical laws. In this direction, Lazovskaya et al. [15] investigated polynomial neural network approximations for the Duffing equation by constructing structured functional representations capable of capturing nonlinear dynamical behavior, demonstrating improved approximation accuracy over standard multilayer perceptron baselines. Verma et al. [16] proposed orthogonal-polynomial and radial-basis-function-inspired architectures for solving nonlinear fractional Liénard and Duffing equations, employing both Chebyshev- and Legendre-based expansions to construct trial solutions and optimizing the resulting residual-based objective functions using simulated annealing. These Chebyshev and Legendre formulations demonstrated that compact single-layer neural architectures could effectively approximate complex nonlinear oscillatory responses while reducing computational complexity compared with conventional multilayer networks, and the use of orthogonal basis functions further improved representational efficiency and facilitated the modeling of fractional-order dynamics. Together, these studies indicate that structured neural approximators can effectively capture complex nonlinear

Article in press

behavior through enriched functional representations. Nevertheless, the resulting models remained primarily approximation-driven and did not explicitly incorporate the governing physical laws into the learning process, thereby limiting their connection to physics-informed learning paradigms.

The emergence of PINNs marked a significant shift in neural-network-based solution strategies for nonlinear dynamical systems. Unlike earlier approximation-oriented approaches, PINNs incorporate the governing differential equations directly into the training objective, enabling the simultaneous enforcement of physical constraints and data-driven function approximation. In the context of the Duffing oscillator, [17] employed PINNs-based formulations to approximate system responses under different forcing conditions, demonstrating strong accuracy in capturing nonlinear dynamics. The study also systematically investigated the influence of activation function selection on convergence behavior and solution quality, revealing that periodic activation functions provide improved performance for oscillatory systems. Despite these advantages, Naveen Raj [17] also highlighted the persistence of spectral bias, whereby high-frequency solution components are learned more slowly than low-frequency modes due to the frequency preference of smooth activation functions, limiting approximation accuracy in highly oscillatory regimes. These findings established PINNs as a promising framework for solving Duffing-type nonlinear systems while simultaneously revealing challenges that motivate further architectural and representational improvements.

The existing studies on neural approaches for solving Duffing-type nonlinear systems span a range of strategies, including metaheuristic-optimized neural networks [14], structured polynomial approximators [15], orthogonal-polynomial-based architectures [16], and Physics-Informed Neural Networks [17]. However, two key limitations persist across these approaches. First, standard PINNs formulations rely on smooth activation functions that are subject to spectral bias, which slows the convergence of high-frequency solution components and reduces approximation accuracy in strongly oscillatory regimes – a particularly consequential limitation for the Duffing oscillator, whose response is characterized by rich high-frequency content. Second, orthogonal polynomial-based neural representations improve functional expressiveness but are not embedded within a physics-informed learning framework, precluding the direct enforcement of governing differential equations during training. To address these limitations, this study proposes a modified PINN in which a Chebyshev polynomial-based activation function is incorporated specifically in the second hidden layer of the network, while the first layer employs standard activation functions.This layer-selective architectural modification is motivated by the well-established approximation properties of Chebyshev polynomials for oscillatory functions and is designed to enhance the representation of high-frequency components without altering the physics-informed loss formulation. The proposed model maintains the core PINN structure while improving its capacity to represent strongly nonlinear oscillatory dynamics. Experimental results presented in Section 4 confirm that the proposed architecture achieves lower residual error under identical hyperparameter settings compared with a conventional PINNs baseline, while incurring only a small increase in computational cost.

3 Proposed Architecture

3.1 Conventional PINN for Solving Differential Equations

PINNs provide a mesh-free framework for solving differential equations – ordinary or partial – by embedding the governing physics directly into the training objective of a neural network (Raissi, Perdikaris, & Karniadakis, 2019). In the conventional formulation, a feedforward neural network with parameters θ approximates the unknown solution u as $\hat{u}(\cdot;\theta)$, with the network inputs corresponding to the independent variables of the governing equation. For the initial-value problem considered in this work, namely the Duffing oscillator, the relevant independent variable is time t, and the network approximates $u(t)$ as $\hat{u}(t)$. Derivatives of $\hat{u}$ with respect to t are obtained through automatic differentiation, enabling direct evaluation of the differential operator $\mathcal{N}$ without temporal discretization (Baydin, Pearlmutter, Radul, & Siskind, 2018).

From a broader perspective, any neural-network-based approach that formulates and minimizes a residual loss associated with governing equations or physical constraints may be viewed as belonging to PINNs paradigm. Accordingly, the essence of a PINN lies in the enforcement of physics through residual minimization rather than in any particular network architecture. For the Duffing oscillator, the network is trained by minimizing a composite loss

$$\mathcal{L}(\theta) = \lambda_{IC}\mathcal{L}_{IC}(\theta) + \lambda_{ODE}\mathcal{L}_{ODE}(\theta), \tag{1}$$

where $\mathcal{L}_{IC}$ penalizes deviations from the prescribed initial conditions and $\mathcal{L}_{ODE}$ is the mean-squared residual of the governing equation

Article in press

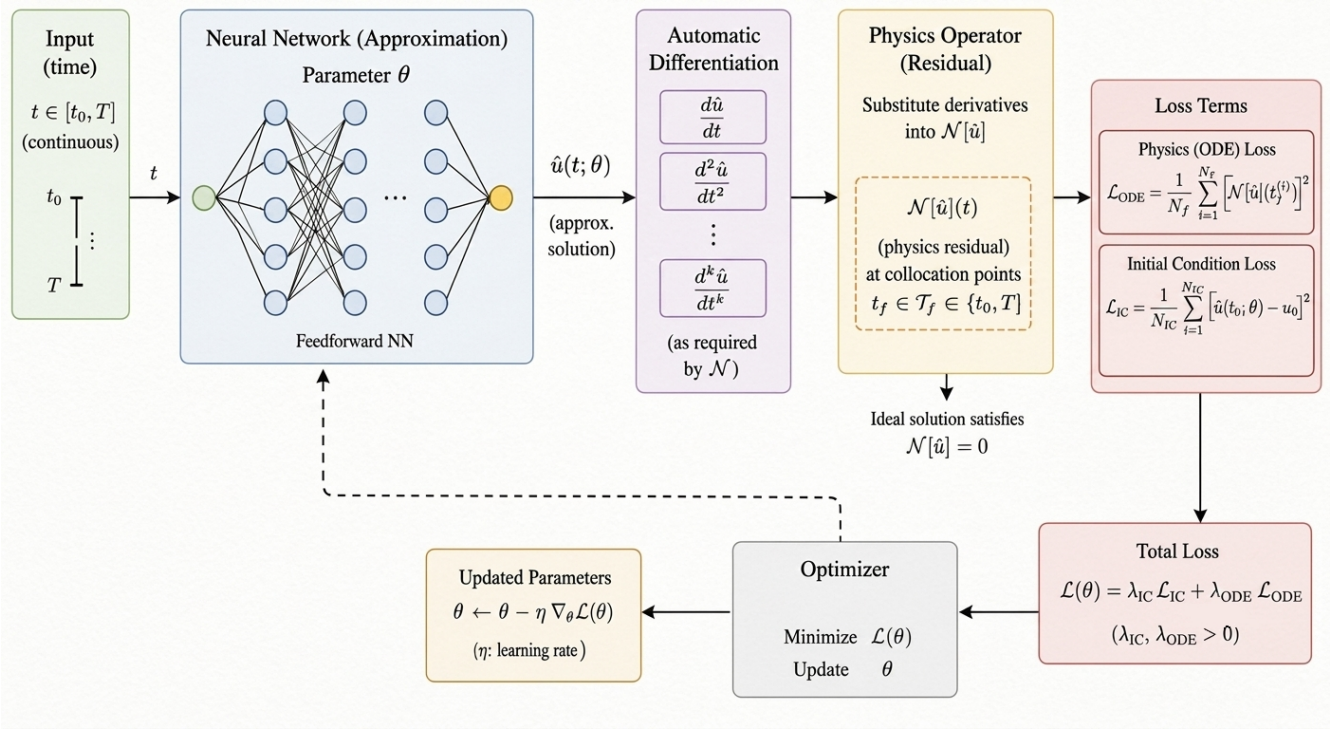


Figure 1. Conventional PINNs for Solving ODEs

evaluated at collocation points sampled over the temporal domain. The weighting coefficients λ_{IC} and λ_{ODE} may be fixed or adaptively tuned, and optimization proceeds via gradient-based methods such as Adam and L-BFGS.

Figure 1 illustrates the schematic architecture of a conventional PINN for a generic initial-value problem. The input t propagates through the hidden layers to produce $\hat{u}(t;\theta)$; automatic differentiation supplies the derivatives required to form $\mathcal{N}[\hat{u}]$, and the resulting residual is combined with the initial-condition loss into the total objective minimized through the optimizer loop. The specific form of $\mathcal{N}$ for the Duffing oscillator is introduced in Section 4.

3.2 Chebyshev Polynomials

Chebyshev polynomials constitute one of the most important classes of orthogonal polynomials in approximation theory, spectral methods, and scientific computing. Owing to their excellent approximation characteristics, numerical stability, and orthogonality properties, they have been extensively employed for solving ordinary and partial differential equations, integral equations, and nonlinear dynamical systems [18–20]. In recent years, Chebyshev-based representations have also attracted increasing attention in machine learning and neural-network-based scientific computing due to their ability to efficiently approximate highly nonlinear and oscillatory functions [21–23].

The Chebyshev polynomials of the first kind, denoted by $T_n(x)$, are defined on the interval $[-1,1]$ through the trigonometric representation

$$T_n(x) = \cos\left(n \arccos(x)\right), \qquad x \in [-1,1],$$

where n denotes the polynomial order. An equivalent and computationally efficient recursive formulation is given by

$$T_0(x) = 1, \tag{2}$$

$$T_1(x) = x, \tag{3}$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n \geq 2. \tag{4}$$

Using the recurrence relations in Eqs. (2), the first few Chebyshev polynomials of the first kind can be generated explicitly as Table 1. These low-order polynomials illustrate the oscillatory and orthogonal structure of the Chebyshev basis functions that are later incorporated into the proposed neural-network architecture.

Article in Press

Table 1. First few Chebyshev polynomials of the first kind

n	$T_n(x)$
0	1
1	x
2	$2x^2-1$
3	$4x^3-3x$
4	$8x^4-8x^2+1$
5	$16x^5-20x^3+5x$

The above recurrence relation enables stable computation of high-order polynomial terms while avoiding the severe numerical instability commonly encountered in standard monomial expansions [24]. This property becomes particularly important in neural-network training, where large polynomial coefficients may lead to gradient explosion and unstable optimization trajectories.

Chebyshev polynomials are orthogonal with respect to the weight function $w(x)=\frac{1}{\sqrt{1-x^2}}$, satisfying the orthogonality condition

$$\int_{-1}^{1} T_m(x)T_n(x)\frac{1}{\sqrt{1-x^2}}\,dx=\begin{cases}0, & m\neq n,\\ \pi, & m=n=0,\\ \frac{\pi}{2}, & m=n\neq 0.\end{cases}$$

This orthogonality property plays a fundamental role in minimizing redundancy among basis functions and improving representational efficiency in approximation tasks. In classical spectral methods, orthogonal polynomial bases are known to significantly improve numerical conditioning and convergence behavior compared with conventional polynomial representations [25, 26].

One of the most remarkable theoretical properties of Chebyshev polynomials is their minimax characteristic. Among all monic polynomials of degree n, the scaled Chebyshev polynomial minimizes the maximum approximation error over the interval $[-1,1]$. Consequently, Chebyshev expansions provide near-optimal uniform approximations and substantially reduce Runge-type oscillatory interpolation errors. Furthermore, for sufficiently smooth target functions, truncated Chebyshev expansions exhibit spectral convergence, meaning that the approximation error decreases exponentially as the polynomial order increases. These properties have made Chebyshev-based spectral approaches highly successful in solving nonlinear differential equations on bounded and semi-infinite domains.

The effectiveness of Chebyshev-based approximations has been demonstrated in numerous scientific computing applications. Motivated by these theoretical and computational advantages, the present study incorporates Chebyshev polynomial activation mechanisms into the hidden representation of the proposed Physics-Informed Neural Network. The rationale behind this integration is that nonlinear oscillatory systems such as the Duffing equation require expressive functional representations capable of efficiently modeling high-frequency and strongly nonlinear solution components. Conventional activation functions, including sigmoid and hyperbolic tangent functions, may exhibit limited representational efficiency in such regimes due to spectral bias and slow learning of oscillatory modes. By contrast, Chebyshev polynomial activations naturally provide a structured orthogonal basis capable of representing oscillatory behaviors more efficiently.

Another important advantage of Chebyshev-based activations is their bounded behavior on the interval $[-1,1]$, which contributes to improved gradient stability during training. Since PINNs optimization involves repeated differentiation of the neural representation through automatic differentiation, stable activation dynamics are highly desirable for preventing optimization instability and stiffness-related training difficulties. Consequently, incorporating Chebyshev polynomial structures into the neural architecture can simultaneously improve approximation quality, physics residual minimization, and convergence stability.

In the proposed ChebyPINN framework, Chebyshev polynomial activations are selectively embedded within the hidden layers of the network to enhance the expressive capacity of the model while preserving the original physics-informed learning formulation. This integration combines the strong approximation properties of orthogonal polynomial bases with the physics-constrained optimization capability of PINNs, providing a theoretically grounded architecture for solving nonlinear oscillatory differential equations such as the Duffing system.

Article in press

3.3 Proposed Chebyshev-Based PINN Framework for the Duffing Oscillator

The proposed ChebyPINN architecture combines the approximation capability of physics-informed neural networks with the spectral representation power of Chebyshev polynomial basis functions. Unlike conventional multilayer perceptrons that learn arbitrary nonlinear feature representations through multiple fully connected layers, the proposed architecture first extracts nonlinear latent features using a standard fully connected layer followed by a hyperbolic tangent activation. These features are then transformed through a second fully connected layer, whose outputs are explicitly expanded into a finite Chebyshev polynomial basis. This explicit basis expansion enables the network to represent highly oscillatory nonlinear solutions more efficiently while preserving the stability of the optimization process.

Let $t \in \mathbb{R}$ denote an arbitrary temporal input. During training, the network is evaluated at a set of collocation points $T = \{t_i\}_{i=1}^{N_f}$, where N_f denotes the total number of collocation points used for enforcing the governing differential equation.

The first hidden layer performs a linear transformation,

$$Z^{(1)}(t) = W^{(1)}t + b^{(1)}, \tag{5}$$

where

$$W^{(1)} \in \mathbb{R}^{m_1 \times 1}, \qquad b^{(1)} \in \mathbb{R}^{m_1},$$

and m_1 denotes the number of neurons in the first hidden layer.

The nonlinear hidden representation is then obtained using the hyperbolic tangent activation function,

$$H^{(1)}(t) = \tanh\left(Z^{(1)}(t)\right), \tag{6}$$

where $H^{(1)}(t) \in \mathbb{R}^{m_1}$. The use of the hyperbolic tangent activation is motivated by its bounded output interval $(-1,1)$, which provides a bounded nonlinear representation in the first hidden layer. This representation is subsequently passed through the second fully connected layer before the Chebyshev polynomial expansion.

The second hidden layer performs another linear transformation,

$$G^{(2)}(t) = W^{(2)}H^{(1)}(t) + b^{(2)}, \tag{7}$$

where

$$W^{(2)} \in \mathbb{R}^{m_2 \times m_1}, \qquad b^{(2)} \in \mathbb{R}^{m_2},$$

and m_2 denotes the number of neurons in the second hidden layer before Chebyshev expansion. Consequently, $G^{(2)}(t) \in \mathbb{R}^{m_2}$. The Chebyshev polynomials of the first kind are defined by

$$T_0(x) = 1, \tag{8}$$

$$T_1(x) = x, \tag{9}$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \qquad n \geq 2. \tag{10}$$

Let d denote the maximum polynomial degree. To explicitly define the polynomial expansion of each neuron in the second hidden layer, write

$$G^{(2)}(t) = \begin{bmatrix} g_1^{(2)}(t) \\ g_2^{(2)}(t) \\ \vdots \\ g_{m_2}^{(2)}(t) \end{bmatrix}.$$

For each polynomial order $n = 0, 1, \ldots, d$, define the corresponding Chebyshev feature vector as

$$\mathbf{T}_n(G^{(2)}(t)) = \begin{bmatrix} T_n(g_1^{(2)}(t)) \\ T_n(g_2^{(2)}(t)) \\ \vdots \\ T_n(g_{m_2}^{(2)}(t)) \end{bmatrix} \in \mathbb{R}^{m_2}. \tag{11}$$

Article in press

The complete Chebyshev feature vector is then constructed by vertically concatenating the polynomial feature vectors of all orders:

$$\Phi_d(G^{(2)}(t)) = \mathrm{col}\Big(\mathbf{T}_0(G^{(2)}(t)), \mathbf{T}_1(G^{(2)}(t)), \ldots, \mathbf{T}_d(G^{(2)}(t))\Big), \tag{12}$$

where col denotes vertical concatenation of the corresponding column vectors.

Since each of the m_2 neurons is expanded into $d+1$ polynomial terms, the resulting Chebyshev feature vector has dimension

$$\dim\Big(\Phi_d(G^{(2)}(t))\Big) = m_2(d+1). \tag{13}$$

Collecting all polynomial features yields the Chebyshev feature vector

$$Z^{(2)}(t) = \Phi_d(G^{(2)}(t)), \tag{14}$$

where

$$Z^{(2)}(t) \in \mathbb{R}^{m_2(d+1)}.$$

Finally, the network output is computed through a linear combination of the Chebyshev features,

$$\hat{u}(t) = \Big(W^{(o)}\Big)^T Z^{(2)}(t) + b^{(o)}, \tag{15}$$

where

$$W^{(o)} \in \mathbb{R}^{m_2(d+1)}, \qquad b^{(o)} \in \mathbb{R}.$$

The proposed architecture therefore operates as a structured nonlinear basis-expansion network. Rather than relying solely on learned nonlinear hidden representations, it explicitly incorporates orthogonal Chebyshev polynomial bases into the approximation process. Owing to the orthogonality of Chebyshev polynomials on the interval $[-1,1]$, different polynomial orders capture complementary characteristics of the target solution while reducing redundancy among learned features. This property improves approximation efficiency and facilitates the representation of highly nonlinear oscillatory responses arising in Duffing systems.

Algorithm 1 details the principal computational steps of the proposed ChebyPINN architecture. Furthermore, Figure 2 illustrates the flowchart of the proposed architecture in this paper.

4 Experimental Results

In this section, we evaluate the performance of the proposed ChebyPINN on four representative cases of the Duffing oscillator. These cases are carefully selected to consider a broad spectrum of dynamical behaviors: the first two admit known exact analytical solutions, enabling direct quantitative assessment of approximation accuracy, while the latter two lack closed-form solutions and therefore represent more challenging, practically cases. For the latter two cases, performance is evaluated primarily through residual minimization and qualitative consistency with expected physical behavior. For each test case, the proposed architecture is systematically compared against a conventional PINN baseline under identical training configurations. Solution quality is assessed using the mean absolute error (MAE) with respect to the exact solution (when available) and the MAE of the governing equation residual (when the exact solution is unavailable). The numerical results demonstrate that the incorporation of Chebyshev polynomial activations consistently improves both solution accuracy and training stability for nonlinear oscillatory systems.

The general form of the Duffing equation considered throughout this section is

$$u''(t) + k_1 u'(t) + k_2 u(t) + k_3 u^3(t) = f(t), \tag{16}$$

$$u(0) = a, \quad u'(0) = b.$$

where $a, b \in \mathbb{R}$, $u(t)$ denotes the displacement as a function of time t, k_1 is the damping coefficient, k_2 is the linear stiffness coefficient, k_3 is the nonlinear (cubic) stiffness coefficient, and $f(t)$ represents the external forcing term.

Article in press

Algorithm 1 Training Procedure of the Proposed ChebyPINN

Require: Collocation points $T = \{t_i\}_{i=1}^{N_f}$, initial conditions (u_0, v_0), number of neurons in the first and second hidden layers m_1 and m_2, maximum Chebyshev polynomial degree d, maximum training iterations $K_{\max}$, loss weights λ_{IC} and λ_{ODE}

Ensure: Optimized network parameters

$$\theta = \left\{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W^{(o)}, b^{(o)}\right\}$$

Randomly initialize $W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W^{(o)}, b^{(o)}$

for $k = 1, 2, \ldots, K_{\max}$ **do**

  for each collocation point $t_i \in T$ **do**

    Compute the first hidden layer

$$Z^{(1)}(t_i) = W^{(1)} t_i + b^{(1)}$$

    Apply the nonlinear activation

$$H^{(1)}(t_i) = \tanh\left(Z^{(1)}(t_i)\right)$$

    Compute the second hidden layer

$$G^{(2)}(t_i) = W^{(2)} H^{(1)}(t_i) + b^{(2)}$$

    Initialize the Chebyshev basis

$$\Phi_0(t_i) = \mathbf{1}_{m_2}, \qquad \Phi_1(t_i) = G^{(2)}(t_i)$$

    for $n = 2, 3, \ldots, d$ **do**

      Compute the Chebyshev recurrence

$$\Phi_n(t_i) = 2G^{(2)}(t_i) \odot \Phi_{n-1}(t_i) - \Phi_{n-2}(t_i)$$

    end for

    Form the Chebyshev feature vector

$$Z^{(2)}(t_i) = \mathrm{col}\left(\Phi_0(t_i), \ldots, \Phi_d(t_i)\right)$$

$$Z^{(2)}(t_i) \in \mathbb{R}^{m_2(d+1)}$$

    Compute the network prediction

$$\hat{u}(t_i) = \left(W^{(o)}\right)^T Z^{(2)}(t_i) + b^{(o)}$$

  end for

  Compute $\hat{u}'(t)$ and $\hat{u}''(t)$ using automatic differentiation

  Compute the Duffing equation residual

$$R(t) = \hat{u}''(t) + k_1 \hat{u}'(t) + k_2 \hat{u}(t) + k_3 \hat{u}^3(t) - f(t)$$

  Compute the initial-condition loss

$$\mathscr{L}_{IC} = \left(\hat{u}(0) - u_0\right)^2 + \left(\hat{u}'(0) - v_0\right)^2$$

  Compute the physics loss

$$\mathscr{L}_{ODE} = \frac{1}{N_f} \sum_{i=1}^{N_f} R(t_i)^2$$

  Compute the total loss

$$\mathscr{L} = \lambda_{IC} \mathscr{L}_{IC} + \lambda_{ODE} \mathscr{L}_{ODE}$$

  Update θ

end for

return $\hat{u}(t)$

Article in press

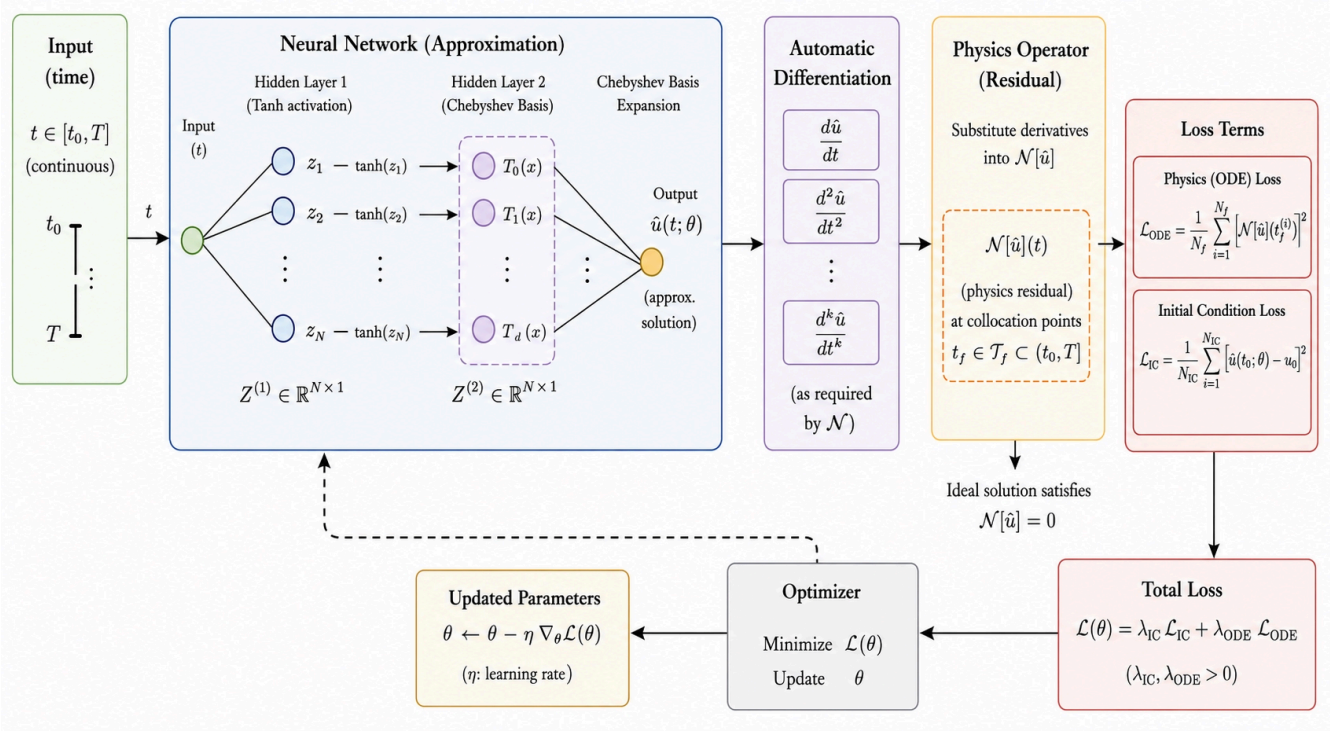


Figure 2. ChebyPINN architecture.

Example 1. In the first example, we consider a Duffing equation case with parameters that admit a known exact analytical solution. This case serves as a baseline to quantitatively assess the approximation accuracy of the proposed architecture against the conventional PINNs. The equation under consideration is as follows [27]:

$$u''(t) + 2u'(t) + u(t) + 8u^3(t) = e^{-3t}, \tag{17}$$
$$u(0) = \frac{1}{2}, \quad u'(0) = -\frac{1}{2}.$$

The exact solution for Equation (17) is $u(t) = \frac{1}{2}e^{-t}$.

In this example, all computations were performed on a GPU-accelerated runtime via Google Colab. Both architectures were trained over the domain $t \in [0, 1]$ using a sequential optimization strategy consisting of Adam iterations followed by L-BFGS refinement steps. For the conventional PINN architecture, the network consisted of two fully connected hidden layers with 64 neurons each. In contrast, the proposed Chebyshev neural network employed two hidden layers with 8 neurons each, where the first hidden layer was a fully connected layer and the second hidden layer was a Chebyshev polynomial-based layer. This controlled experimental setup ensures that any observed differences in approximation accuracy are attributable primarily to the architectural modification introduced through the Chebyshev-based hidden representation.

The results confirm that the proposed framework maintains stable and consistent performance across a broad range of configurations, with optimal accuracy achieved at $d = 8$ and $N_f = 2000$.

Figure 3 presents the predicted solution profiles of both architectures alongside the exact solution $u(t) = \frac{1}{2}e^{-t}$. Both models successfully reproduce the monotonically decaying behavior of the damped Duffing oscillator and remain in close visual agreement with the exact solution throughout the entire computational domain. The solution curves of the two architectures are nearly overlapping at this scale, reflecting the overall adequacy of the physics-informed training procedure for this configuration. The quantitative distinction between the two models, which is not fully discernible from the solution plot alone, is examined more rigorously through the pointwise error analysis presented in Figure 4.

Figure 4 displays the pointwise absolute error of both architectures on a logarithmic scale, revealing a clear distinction between the two architectures. The error profile of the ChebyPINN remains predominantly within the range of $[2 \times 10^{-8}, 4 \times 10^{-7}]$, exhibiting several

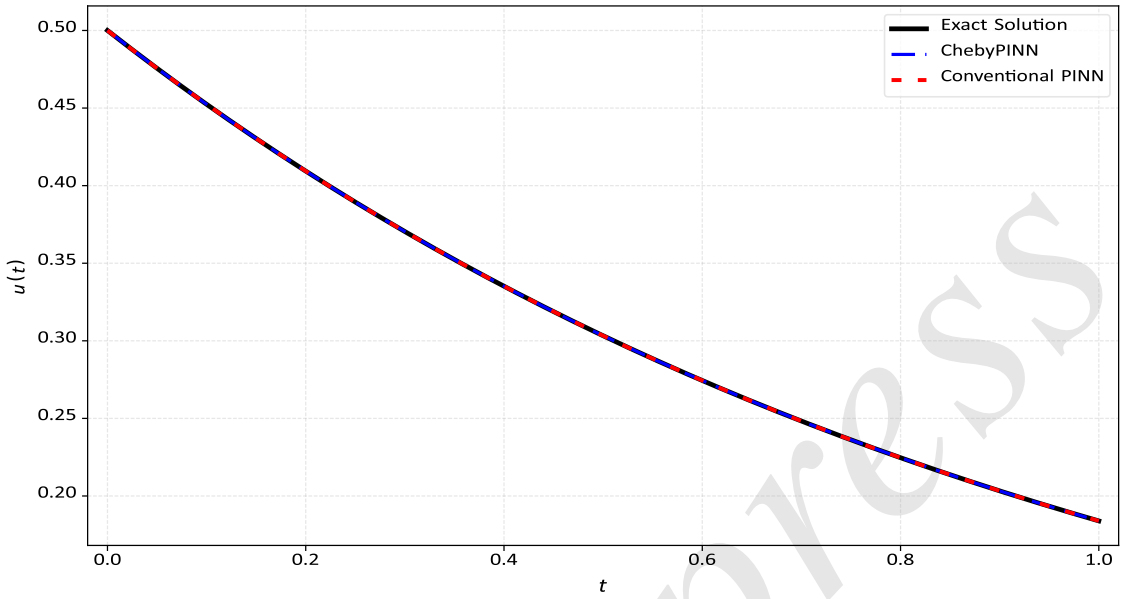


Figure 3. Predicted solution profiles of the conventional PINN and the proposed ChebyPINN compared with the exact solution for Example 1.

oscillatory patterns with multiple local maxima and minima throughout the computational domain. Several pronounced reductions in the error can be observed at different locations, where the error decreases by more than one order of magnitude. Apart from these local variations, the error remains consistently low over the entire interval, indicating the high approximation capability of the proposed architecture. The error of the conventional PINN, by contrast, remains mainly within the range of $[2 \times 10^{-7}, \times 10^{-5}]$, displaying larger fluctuations and significantly higher error values over most of the domain. Consequently, the conventional PINN error is approximately one to two orders of magnitude higher than the ChebyPINN in most regions, demonstrating the improved approximation accuracy achieved by incorporating the Chebyshev-based activation mechanism.

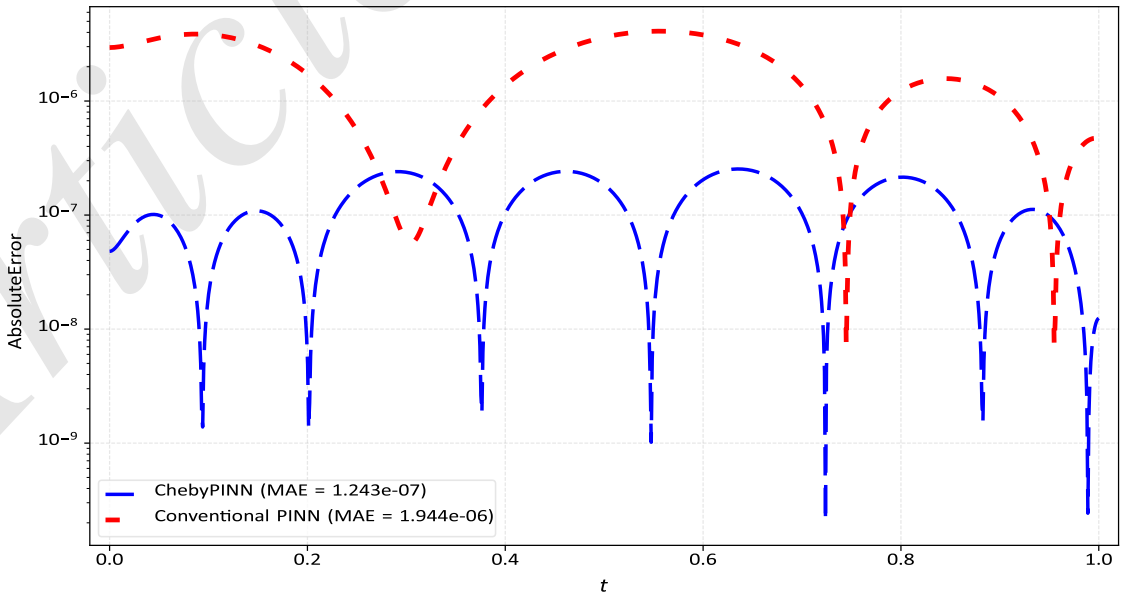


Figure 4. Pointwise absolute error distributions of the conventional PINN and the proposed ChebyPINN on a logarithmic scale.

Article in press

Table 2 investigates the effect of the number of Adam training epochs on approximation accuracy and computational cost for Example 1. The results indicate that increasing the number of epochs initially improves the solution accuracy, reducing the MAE from 1.750×10^{-7} at 100 epochs to a minimum value of 1.594×10^{-7} at 3000 epochs. Beyond this point, additional training leads to a gradual deterioration in performance together with a substantial increase in computational time, suggesting the onset of overfitting or optimization instability. Consequently, 3000 Adam epochs provide the best balance between accuracy and efficiency for the present problem.

Table 2. Effect of Adam training epochs on MAE and training time for Example 1.

Adam epochs	MAE	Training time (s)
100	1.750×10^{-7}	2.78
500	1.623×10^{-7}	8.99
1000	1.722×10^{-7}	19.73
3000	$\mathbf{1.594 \times 10^{-7}}$	41.51
5000	2.033×10^{-7}	69.04
7000	1.020×10^{-6}	143.23

Table 3 analyzes the effect of the number of LBFGS refinement iterations on model performance. A substantial reduction in MAE is observed when increasing the LBFGS iterations from 1 to 500, demonstrating the importance of second-order optimization in refining the PINN solution after the Adam pretraining stage. However, further increases beyond 500 iterations produce almost no measurable improvement in accuracy while maintaining a comparable computational cost. This indicates that the optimization process has effectively converged by approximately 500 LBFGS iterations.

Table 3. Effect of LBFGS refinement iterations on MAE and training time.

LBFGS iterations	MAE	Time (s)
1	1.779×10^{-3}	46.51
100	2.760×10^{-6}	51.08
500	$\mathbf{1.323 \times 10^{-7}}$	47.55
1000	1.323×10^{-7}	47.03
2000	1.323×10^{-7}	51.81
3000	1.323×10^{-7}	47.65

Table 4 examines the influence of the Chebyshev polynomial degree on prediction accuracy and training time. The results demonstrate that low polynomial degrees are insufficient to accurately capture the nonlinear dynamics of the Duffing oscillator, whereas increasing the polynomial degree significantly improves approximation quality up to an optimal range. Beyond the optimal degree, the improvement becomes marginal while the computational cost increases, indicating diminishing returns from further expansion of the polynomial basis. These observations confirm that an appropriate polynomial degree is essential for balancing approximation capability and computational efficiency.

Table 5 presents the effect of the number of collocation points on approximation accuracy and computational time. Increasing the collocation density improves the enforcement of the governing differential equation and gradually reduces the MAE, with the best performance achieved at $N_f = 2000$. Beyond this value, the accuracy improvement becomes negligible, whereas the computational cost continues to grow significantly. These results suggest that moderate collocation densities are sufficient to achieve highly accurate solutions for the present Duffing oscillator problem.

Table 6 summarizes the quantitative comparison between the conventional PINN and the proposed ChebyPINN for Example 1. Under identical training conditions, the ChebyPINN achieves a mean absolute error of 1.243×10^{-7}, compared with 1.944×10^{-6} for the conventional PINN, corresponding to an improvement of more than one order of magnitude in approximation accuracy. Although the proposed architecture requires a longer training time due to the additional computational complexity introduced by the Chebyshev activation

Article in press

Table 4. Effect of the Chebyshev polynomial degree on MAE and training time for Example 1.

Degree d	MAE	Training time (s)
3	1.657×10^{-6}	58.14
4	2.413×10^{-6}	61.79
5	6.624×10^{-6}	73.01
6	6.122×10^{-7}	82.56
7	1.594×10^{-7}	41.51
8	$\mathbf{1.375 \times 10^{-7}}$	47.19
9	6.969×10^{-7}	108.47
10	2.834×10^{-6}	111.99

Table 5. Effect of number of collocation points (N_f) on MAE and computational time. Fixed parameters: epochs = 3000, Chebyshev degree = 8, IC/ODE weight = 10, learning rate (Adam) = 10^{-5}, and LBFGS iterations = 500.

N_f	MAE	Time (s)
100	2.028×10^{-7}	52.90
300	1.392×10^{-7}	53.42
400	1.323×10^{-7}	47.55
600	1.305×10^{-7}	54.07
800	1.291×10^{-7}	54.83
1000	1.287×10^{-7}	49.75
2000	$\mathbf{1.243 \times 10^{-7}}$	59.55
4000	1.292×10^{-7}	73.55
6000	1.286×10^{-7}	97.67
8000	1.282×10^{-7}	128.41

layer, the resulting gain in solution fidelity demonstrates the effectiveness of the proposed spectral-inspired hidden representation for nonlinear Duffing systems.

Table 6. Quantitative comparison of approximation accuracy and computational cost for Example 1 between the conventional PINN and the proposed ChebyPINN.

Model	MAE	Training time (s)
Conventional PINN	1.944×10^{-6}	27.96
ChebyPINN (proposed)	$\mathbf{1.243 \times 10^{-7}}$	60.63

Example 2. The second test case examines a Duffing oscillator with different parameter values that also possesses a closed-form analytical solution. This example allows us to further validate the performance of the proposed architecture. The equation is as follows:

$$u''(t) + 3u(t) - 2u^3(t) = \cos(t)\sin(2t), \tag{18}$$

$$u(0) = 0, \qquad u'(0) = 1.$$

The exact solution for Equation (18) is $u(t) = \sin(t)$. Unlike the damped, forced Duffing oscillator considered in Example 1, the present example investigates an undamped, forced Duffing oscillator.

Figure 5 presents the predicted solution for both architectures (conventional PINNs and the proposed architecture in this paper) alongside the exact solution $u(t) = \sin(t)$. Both models accurately reproduce the smooth behavior of the undamped forced Duffing oscillator across

Article in press

the entire computational domain, with the predicted curves nearly indistinguishable from the exact solution at this scale. This close visual agreement confirms that both physics-informed formulations satisfy the governing differential equation and initial conditions; nevertheless, the substantial difference in approximation quality between the two models is not discernible from the solution profiles alone and is examined through the pointwise error analysis in Figure 6.

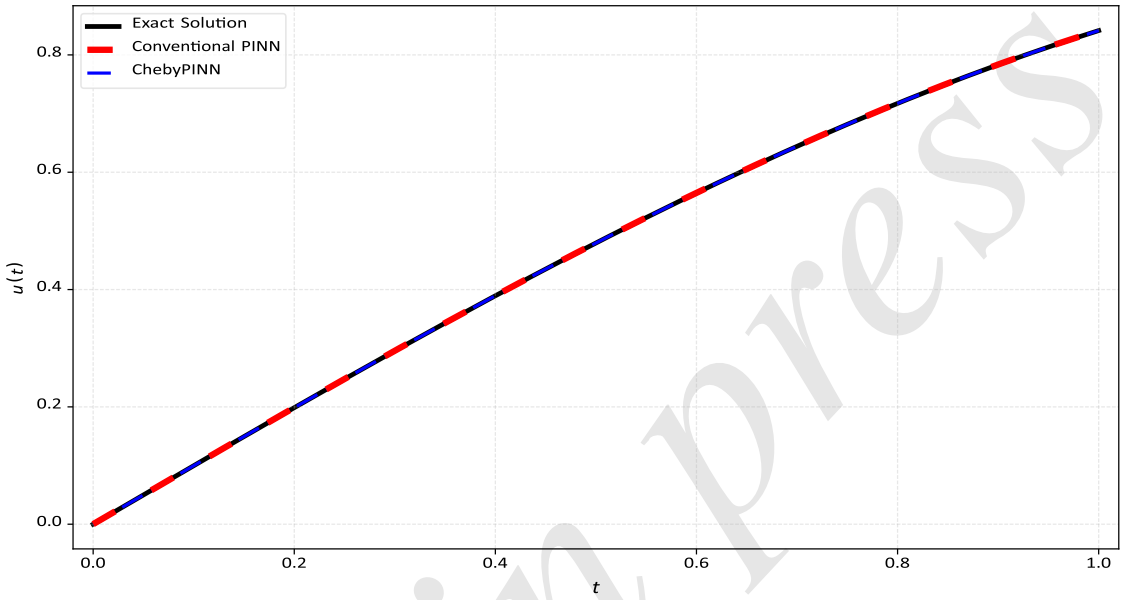


Figure 5. Predicted solution profiles of the conventional PINN and the proposed ChebyPINN compared with the exact solution for Example 2.

Figure 6 displays the pointwise absolute error of both architectures on a logarithmic scale. The error profile of the ChebyPINN oscillates predominantly within the range $[2 \times 10^{-8}, 10^{-6}]$, exhibiting several local maxima together with a pronounced minimum located near $t \approx 0.30$, where the error drops by more than one order of magnitude. Apart from this localized minimum, the error remains relatively stable throughout the computational domain, indicating consistently high approximation accuracy. The error of the conventional PINN, by contrast, remains on the order of 10^{-4} across the entire domain, displaying only mild variations while gradually increasing toward larger values of t. Consequently, the conventional PINN error remains approximately two to three orders of magnitude above the ChebyPINN throughout most of the computational domain, demonstrating the superior approximation capability of the proposed Chebyshev-based architecture.

Table 7 investigates the influence of the number of Adam training epochs on the approximation accuracy and computational cost for Example 2. Increasing the number of epochs from 500 to 4000 leads to a substantial reduction in the mean absolute error, indicating that the network progressively learns a more accurate representation of the solution. The minimum MAE is achieved at 4000 epochs, after which further training causes a noticeable degradation in accuracy despite the increased computational time. This behavior suggests that excessive optimization no longer improves generalization and may introduce optimization instability. Consequently, 4000 Adam epochs were selected as the optimal compromise between solution accuracy and training efficiency for the remainder of the experiments.

Table 8 examines the effect of the Chebyshev polynomial degree on the performance of the proposed architecture. Increasing the polynomial degree generally enhances the approximation capability of the network up to an optimal value, with the lowest MAE obtained at degree $d = 12$. Beyond this point, the approximation accuracy becomes less stable, and higher polynomial degrees can even produce a significant deterioration in performance while continuing to increase the computational cost. These results demonstrate that an appropriate polynomial degree is essential for achieving an effective balance between expressive power, numerical stability, and computational efficiency.

Table 9 evaluates the influence of the number of collocation points on the solution accuracy. Increasing the number of collocation points from 1000 to 3000 substantially improves the enforcement of the equation and reduces the approximation error by several orders of magnitude, with the best performance achieved at $N_f = 3000$. However, further increasing the collocation density to 4000 and 5000

Article in press

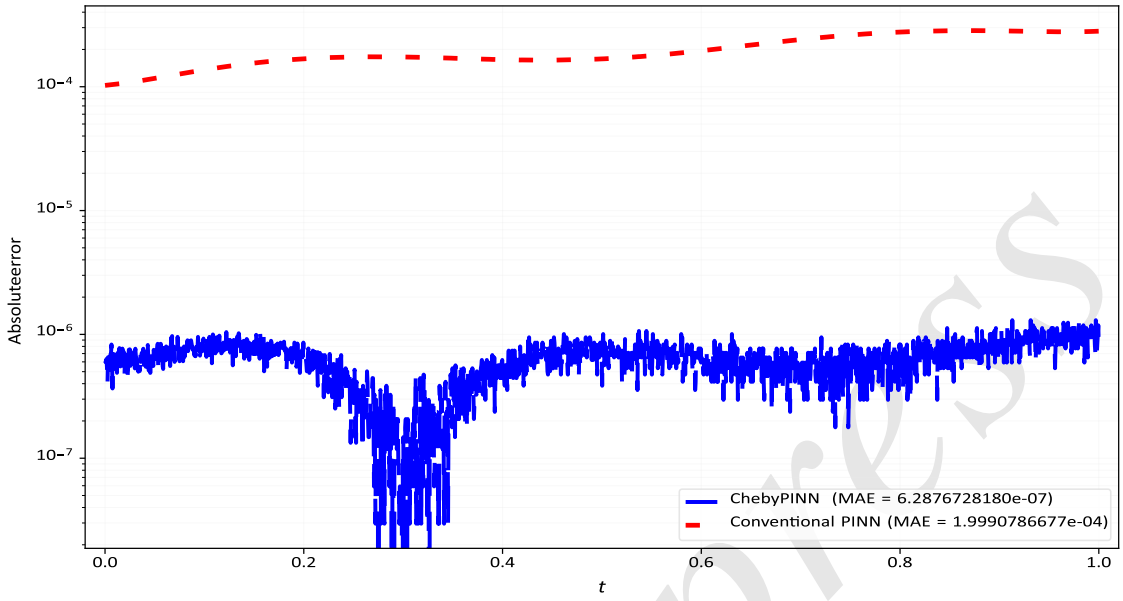


Figure 6. Pointwise absolute errors of the conventional PINN and the proposed ChebyPINN for Example 2, shown on a logarithmic scale.

Table 7. Effect of the number of Adam training epochs on MAE and training time for Example 2.

Adam epochs	MAE	Training time (s)
500	$1.1473431950 \times 10^{-3}$	7.54
1000	$1.1396248738 \times 10^{-4}$	15.44
2000	$1.1409282706 \times 10^{-5}$	30.87
3000	$1.0756386473 \times 10^{-5}$	45.77
4000	$\mathbf{1.0190218745 \times 10^{-5}}$	46.96
5000	$3.6705412640 \times 10^{-5}$	76.11
6000	$2.6669187355 \times 10^{-4}$	91.28
7000	$4.4936145423 \times 10^{-4}$	105.52
8000	$3.4531604033 \times 10^{-3}$	121.53

results in a marked deterioration in accuracy while providing little change in computational time. These results indicate that an excessively dense collocation set does not necessarily improve the learned solution and that a moderate number of collocation points provides the most effective training configuration for this problem.

Table 10 presents the quantitative comparison between the conventional PINN and the proposed ChebyPINN under identical training conditions. The proposed architecture achieves a mean absolute error of 6.288×10^{-7}, whereas the conventional PINN attains 1.999×10^{-4}, corresponding to an improvement of more than two orders of magnitude in approximation accuracy. Although the ChebyPINN requires a longer training time due to the additional computations associated with the Chebyshev polynomial layer, the substantial improvement in solution accuracy demonstrates the effectiveness of incorporating Chebyshev polynomial representations into the hidden-layer architecture for solving nonlinear Duffing equations.

Example 3. In the third example, we investigate the undamped, unforced Duffing equation, which does not have a known closed-form analytical solution. This example tests the ability of the proposed architecture to minimize the residual while producing physically consistent dynamics.

$$u''(t) + u(t) + u^3(t) = 0, \tag{19}$$

Article in press

Table 8. Effect of the Chebyshev polynomial degree on MAE and training time for Example 2. Fixed parameters: Adam epochs = 4000, $\eta_{\text{Adam}} = 10^{-3}$, $N_f = 2000$, and $\lambda_{\text{ODE}} = \lambda_{\text{IC}} = 1$.

Degree (d)	MAE	Training time (s)
3	$5.5226060795 \times 10^{-5}$	34.91
4	$4.1501083615 \times 10^{-5}$	36.54
5	$2.5121730869 \times 10^{-4}$	41.53
6	$3.3224845538 \times 10^{-5}$	47.76
7	$2.5528599508 \times 10^{-5}$	52.84
8	$1.0190218745 \times 10^{-5}$	62.03
9	$9.2785645393 \times 10^{-5}$	62.93
10	$6.3547595346 \times 10^{-5}$	69.75
11	$7.2271400131 \times 10^{-4}$	73.10
12	$\mathbf{5.9563494688 \times 10^{-6}}$	79.01
13	$1.0258057591 \times 10^{-4}$	83.56
14	$8.5523670350 \times 10^{-6}$	90.27
15	$3.5938250221 \times 10^{-5}$	94.67
16	$4.8359902576 \times 10^{-3}$	99.81

Table 9. Effect of the number of collocation points on MAE and training time for Example 2. Fixed parameters: Adam epochs = 4000, $d = 12$, $\eta_{\text{Adam}} = 10^{-3}$, and $\lambda_{\text{ODE}} = \lambda_{\text{IC}} = 1$.

N_f	MAE	Training time (s)
1000	$2.1937236306 \times 10^{-4}$	83.38
2000	$9.1782402478 \times 10^{-7}$	82.88
3000	$\mathbf{6.2876733864 \times 10^{-7}}$	83.17
4000	$8.8686160743 \times 10^{-7}$	83.93
5000	$3.9781235158 \times 10^{-6}$	84.60

Table 10. Quantitative comparison of approximation accuracy and computational cost between the conventional PINN and the proposed ChebyPINN.

Model	MAE	Training time (s)
Conventional PINN	$1.9990786677 \times 10^{-4}$	25.36
ChebyPINN (proposed)	$\mathbf{6.2876733864 \times 10^{-7}}$	91.30

with the initial conditions:

$$u(0) = 1, \qquad u'(0) = 5. \tag{20}$$

For Eq. (19), M. A. Al-Jawary et al have presented a fifth-order Taylor series expansion about $t = 0$ was given as follows [28]:

$$u(t) \approx 1 + 5t - t^2 - \frac{10}{3}t^3 - \frac{25}{4}t^4 + \frac{25}{4}t^5. \tag{21}$$

It should be emphasized that this truncated series constitutes a local approximation valid only in a neighborhood of the initial point and is not a global solution of Eq. (19); consequently, deviations between the network predictions and this reference at larger t reflect the limited radius of validity of the Taylor expansion rather than a deficiency of the learned solution. Accordingly, the physics-informed residual, rather than the distance to the Taylor reference, is adopted as the primary indicator of solution accuracy in this example.

The absence of a known analytical solution, together with the comparatively large initial slope $u'(0) = 5$ and the conservative, oscillatory

Article in press

character of Eq. (19), makes this example particularly sensitive to any residual inaccuracy in the imposed initial conditions, as such inaccuracy would otherwise be indistinguishable from genuine solution error in the absence of a ground-truth reference. To remove this potential source of ambiguity and to ensure that the reported physics residual reflects solely the network's ability to satisfy the governing equation, this example additionally incorporates a hard-constraint formulation of the initial conditions.

To eliminate the reliance on soft-constrained penalty terms, the initial conditions are enforced as hard constraints through a tailored ansatz. The network output is constructed as

$$u(t) = 1 + 5t + t^2 N(t), \tag{22}$$

where $N(t)$ denotes the raw output of the two-hidden-layer architecture. By construction, this parameterization satisfies $u(0) = 1$ and $u'(0) = 5$ identically, regardless of the network parameters. As a result, the initial-condition residual is reduced to zero, and the corresponding penalty term is removed from the objective function, so that the total loss depends solely on the governing-equation residual:

$$\text{loss}_{\text{total}} = \text{loss}_{\text{ode}}. \tag{23}$$

Figure 7 presents the predicted solution of both architectures alongside the fifth-order Taylor reference. Near $t = 0$, the predicted profiles of both architectures lie close to the Taylor curve before gradually deviating from it as t increases. The Taylor reference follows the PINN predictions in the early part of the domain; however, the discrepancy becomes more noticeable for larger values of t, where the truncated Taylor approximation loses accuracy away from the expansion point. The solution profiles of the two PINN architectures remain close to one another and follow a similar trend throughout $t \in [0, 1]$, both reproducing the initial growth and subsequent turning behavior characteristic of the conservative Duffing oscillator, indicating that the choice of second-hidden-layer activation function does not qualitatively alter the learned solution trajectory even though the Taylor reference is only reliable near $t = 0$.

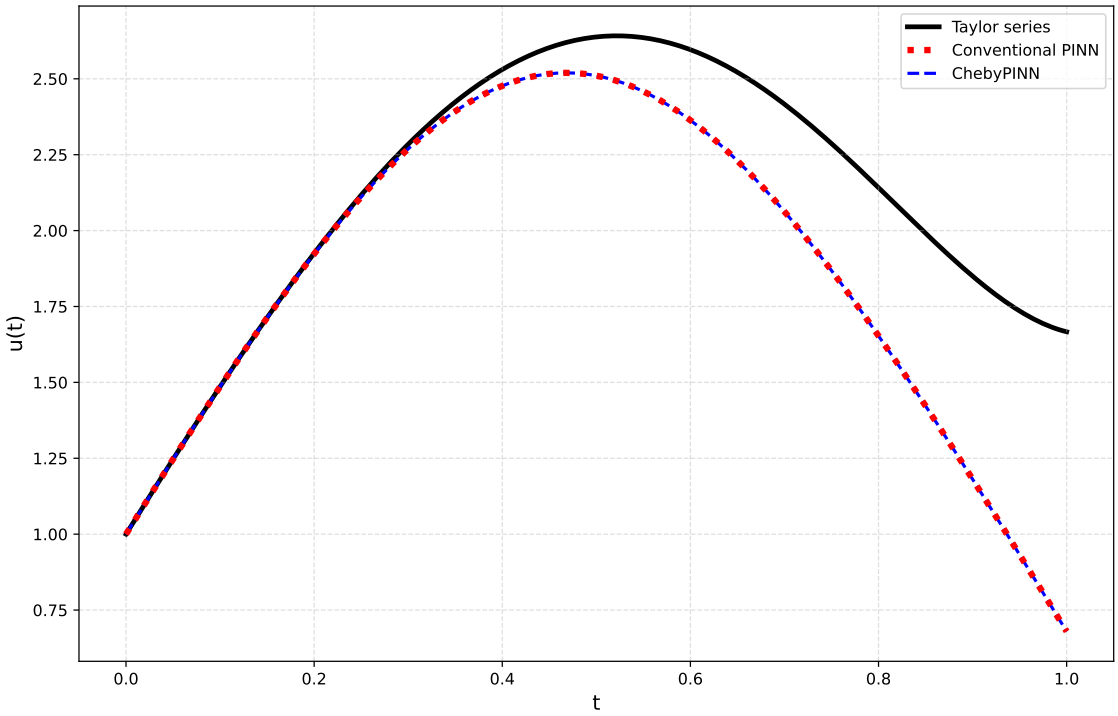


Figure 7. Comparison of the predicted solution profiles $u(t)$ obtained by the ChebyPINN and the conventional PINN, along with the fifth-order Taylor series expansion used as a local reference, for Example 3.

Figure 8 displays the pointwise absolute difference between each model's prediction and the Taylor reference on a logarithmic scale.

A more discriminating assessment is obtained from the physics-informed residual, shown in Figure 9 on a logarithmic scale. The ChebyPINN exhibits a regular oscillatory residual profile bounded within approximately $[5 \times 10^{-6}, 1.6 \times 10^{-3}]$. The conventional PINN

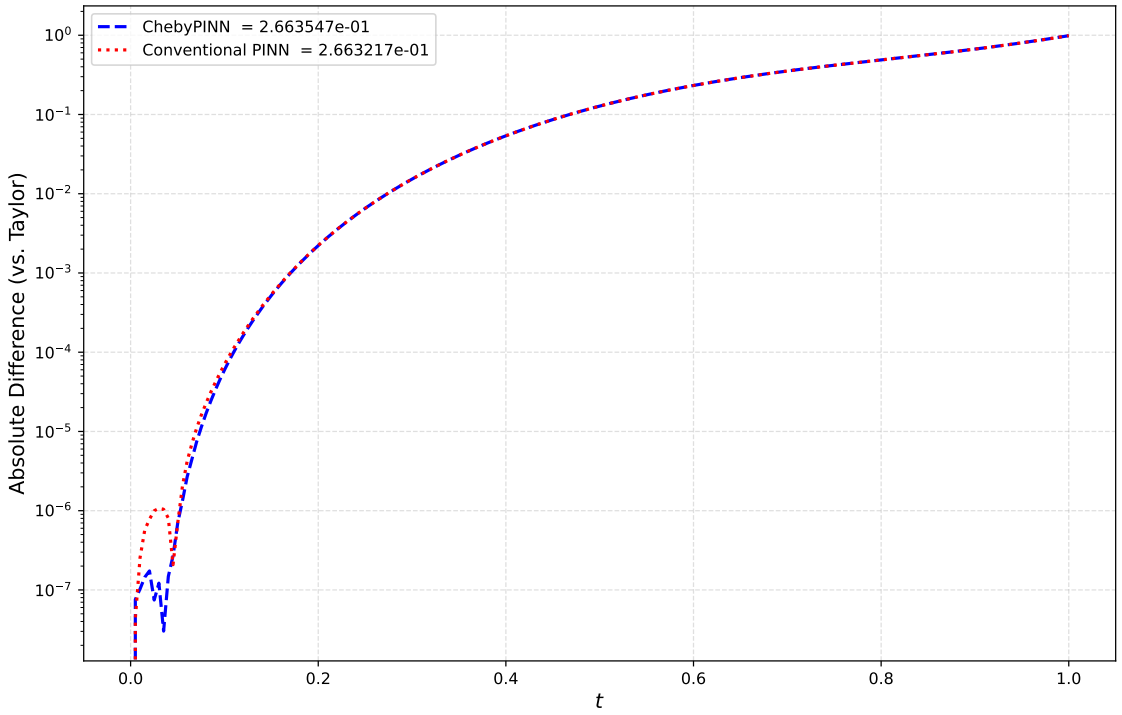


Figure 8. Pointwise absolute error between model predictions and the fifth-order Taylor reference on a logarithmic scale for Example 3.

spans a comparably shaped but systematically elevated range of approximately $[6 \times 10^{-6}, 8 \times 10^{-3}]$, remaining generally above the ChebyPINN curve over most of the domain.

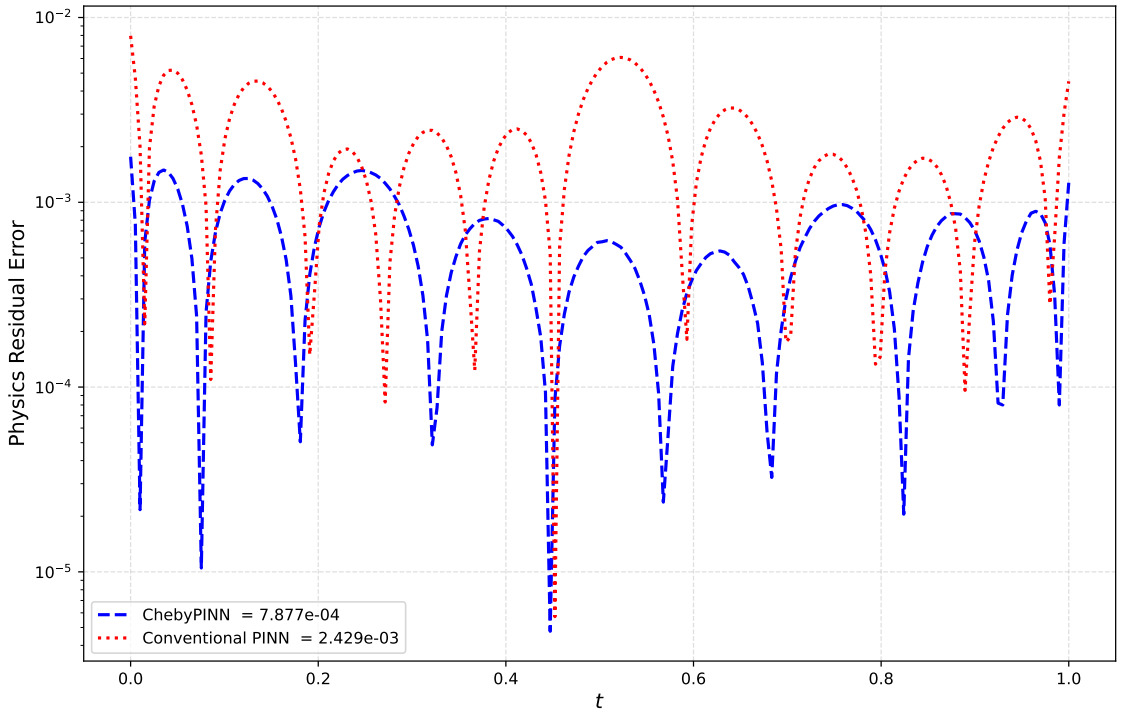


Figure 9. Physics-informed residual of the Duffing equation for both models on a logarithmic scale for Example 3.

The mean absolute residuals are $7.87713332 \times 10^{-4}$ and $2.42905458 \times 10^{-3}$ for the ChebyPINN and the conventional PINN, respectively, corresponding to an improvement of approximately a factor of three. Since no exact closed-form solution is available for this example, the governing-equation residual constitutes the most direct measure of physical consistency, and this result indicates that the proposed Chebyshev activation yields a solution that more accurately satisfies the underlying differential equation over most of the domain.

This improvement in physical consistency is obtained at the cost of a longer training time, 679.62 seconds for the ChebyPINN compared to 285.58 seconds for the conventional PINN.

Table 11 investigates the influence of the Chebyshev polynomial degree on the residual error and computational cost for Example 3. The residual error decreases substantially as the degree increases from 3 to 5, where the minimum residual is achieved. Increasing the polynomial degree beyond this point generally leads to a deterioration in accuracy together with greater computational expense, indicating that excessive polynomial complexity does not improve the approximation and may adversely affect optimization.

Table 11. Effect of the Chebyshev polynomial degree on prediction accuracy and computational cost for Example 3.

Chebyshev Degree	MAE (residual)	Training time (s)
3	$1.57587528 \times 10^{-2}$	144.50
4	$6.04619831 \times 10^{-2}$	649.37
5	$\mathbf{8.15521576 \times 10^{-4}}$	688.76
6	$1.41532719 \times 10^{-2}$	731.39
7	$1.42259756 \times 10^{-3}$	795.77
8	$1.30394911 \times 10^{-2}$	2091.86
9	$1.59133300 \times 10^{-2}$	1337.37

Table 12 examines the effect of the maximum number of LBFGS refinement iterations on the residual error. Increasing the maximum iterations from 100 to 500 produces more than an order-of-magnitude reduction in the residual error. However, further increasing the iteration limit to 1000 and 3000 yields virtually identical residual errors while providing no meaningful reduction in training time. This behavior indicates that the optimization process has effectively converged after approximately 500 LBFGS iterations. Consequently, a maximum of 500 LBFGS iterations was selected as the default configuration for the subsequent experiments.

Table 12. Effect of the LBFGS maximum iterations on prediction accuracy and computational cost for Example 3 for fixed Adam epochs $= 6000$, and Chebyshev degree $= 5$.

LBFGS max iter	MAE (residual)	Training time (s)
100	$1.08736185 \times 10^{-2}$	839.84
500	$\mathbf{7.87713332 \times 10^{-4}}$	774.38
1000	$7.87713332 \times 10^{-4}$	808.69
3000	$7.87713332 \times 10^{-4}$	1102.79

Table 13 presents the quantitative comparison between the conventional PINN and the proposed ChebyPINN for Example 3. The proposed ChebyPINN achieves a residual error approximately three times smaller than that of the conventional PINN under the same training conditions, demonstrating its superior ability to produce a physics-consistent solution. Although the proposed architecture requires a longer training time because of the additional computations associated with the Chebyshev polynomial layer, the substantial reduction in residual error highlights the effectiveness of incorporating Chebyshev polynomial representations into the hidden-layer architecture for solving nonlinear Duffing equations without closed-form analytical solutions.

Example 4. The final test case considers a strongly nonlinear, undamped, and unforced Duffing oscillator with parameters leading to complex dynamical behavior. Since no exact analytical solution is available, performance is evaluated primarily through residual error. To further validate the predictive capability of the proposed model, the obtained solutions are also compared with the numerical results reported

Article in press

Table 13. Quantitative comparison of the conventional PINN and the proposed ChebyPINN for Example 3.

Model	MAE (residual)	Training time (s)
ChebyPINN (proposed)	$7.87713332 \times 10^{-4}$	679.62
Conventional PINN	$2.42905458 \times 10^{-3}$	285.58

in Table 1 of Syam et al. [29].

$$u''(t) + u(t) + u^3(t) = 0, \tag{24}$$

with the initial conditions

$$u(0) = 1, \qquad u'(0) = 0. \tag{25}$$

Unlike Example 3, the initial conditions are imposed through the conventional soft-constrained PINN formulation rather than a hard-constraint parameterization. Consequently, both the governing equation and the prescribed initial conditions are enforced simultaneously during training. This example therefore evaluates the proposed ChebyPINN under the same optimization framework as the conventional PINN, allowing the influence of the Chebyshev activation function on the solution quality to be assessed directly.

Figure 10 compares the results predicted by the conventional PINN and the proposed ChebyPINN over the computational interval. The predicted solutions almost completely overlap throughout the domain, indicating that both models accurately capture the nonlinear oscillatory behavior of the Duffing equation. Although the overall solution trajectories are visually indistinguishable, the subsequent analysis of the physics-informed residual provides further insight into the accuracy and consistency of the two models.

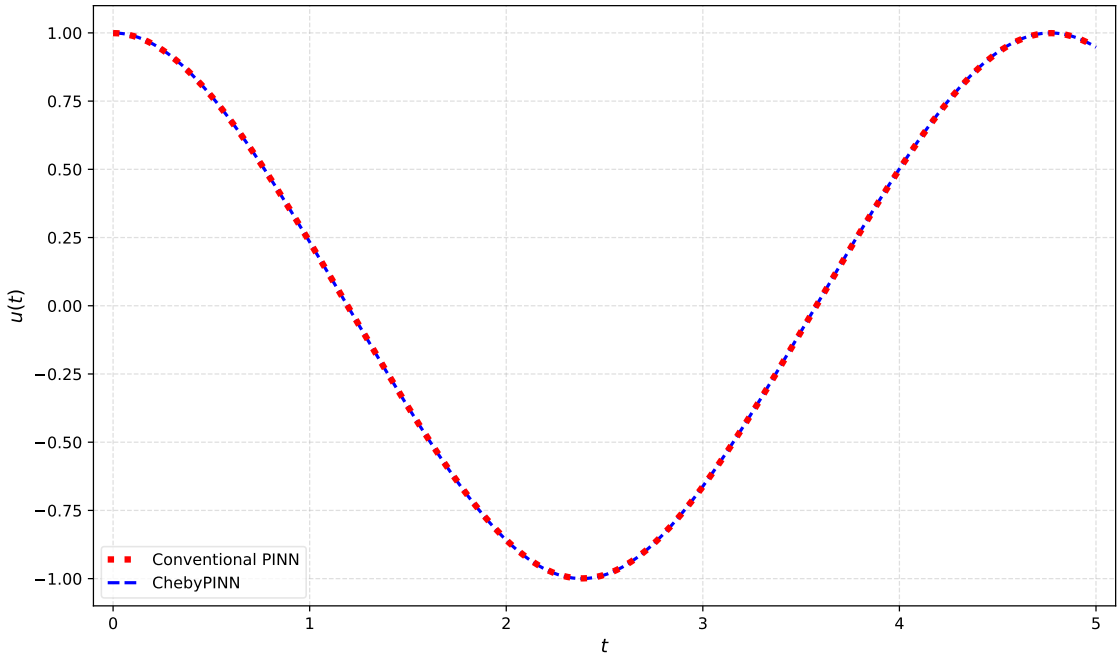


Figure 10. Comparison of the predicted solution profiles obtained by the ChebyPINN and the conventional PINN for Example 4.

Figure 11 displays the physics-informed residual of both models on a logarithmic scale. The ChebyPINN exhibits a regular oscillatory bounded within approximately $[1 \times 10^{-6}, 4 \times 10^{-3}]$. The conventional PINN follows a similar oscillatory pattern but spans a systematically higher range of approximately $[1 \times 10^{-5}, 8 \times 10^{-3}]$, remaining generally above the ChebyPINN curve over most of the computational domain. The mean absolute residuals are 1.160×10^{-4} and 5.761×10^{-4} for the ChebyPINN and the conventional PINN, respectively, corresponding to an improvement of approximately a factor of five. The governing-equation residual provides the most direct measure of physical consistency, indicating that the proposed Chebyshev activation enables the network to satisfy the underlying differential equation

more accurately over most of the computational interval. This improvement in physical consistency is obtained at the cost of a longer training time, with 389.00 seconds required for the ChebyPINN compared to 294.83 seconds for the conventional PINN.

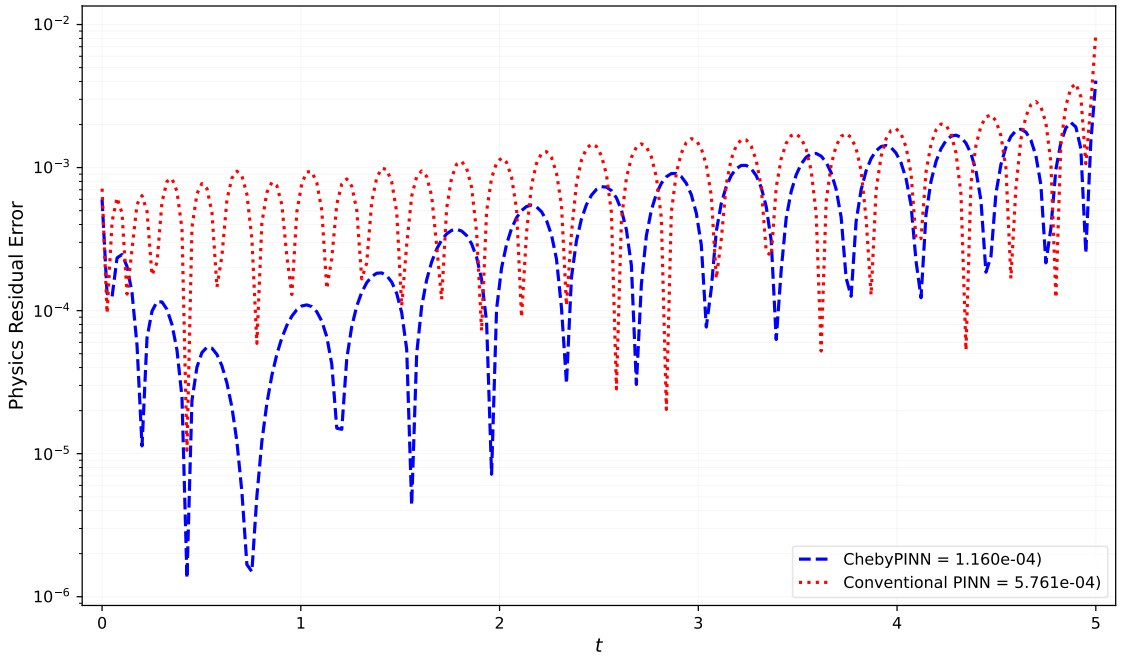


Figure 11. Physics-informed residual of the Duffing equation for both models on a logarithmic scale for Example 4.

Table 14 investigates the influence of the number of Adam optimization epochs on the prediction accuracy of the proposed ChebyPINN. Increasing the number of training epochs from 20,000 to 40,000 leads to a significant reduction in the prediction error, indicating that the network continues to improve its approximation capability during this stage of the optimization process. However, beyond 40,000 epochs, the improvement becomes considerably less pronounced despite the substantial increase in computational cost. Consequently, 40,000 Adam epochs were selected as the default training configuration, since this setting provides an appropriate balance between prediction accuracy and computational efficiency for the subsequent experiments.

Table 14. Effect of the number of Adam training epochs on the governing-equation residual and computational cost for Example 4.

Adam epochs	Residual MAE	Training time (s)
20k	0.0010693548	316.07
30k	0.0026879602	663.15
40k	0.0002369090	1480.51
50k	0.0003274823	1645.00
60k	0.0003435644	2068.91
80k	0.0004144612	2032.93

Table 15 examines the influence of the Chebyshev polynomial degree on the governing-equation residual. The residual decreases substantially as the polynomial degree increases from three to four, where the minimum residual is achieved. Increasing the polynomial degree beyond this point generally leads to a deterioration in accuracy together with greater computational expense. Therefore, a fourth-degree Chebyshev polynomial was selected for the remaining experiments.

Table 16 compares the predictions of the proposed ChebyPINN and the conventional PINN with the numerical reference solutions reported by Syam et al. [29]. Both architectures show excellent agreement with the published numerical results at all selected time instances. The proposed ChebyPINN generally produces slightly smaller deviations from the reference values, demonstrating its ability to accurately reproduce the nonlinear dynamics of the Duffing equation.

Article in

Table 15. Effect of the Chebyshev polynomial degree on the governing-equation residual and computational cost for Example 4.

Chebyshev degree	Residual MAE	Training time (s)
3	0.0013595445	983.55
4	0.0002087551	1623.95
5	0.0002369090	1480.51
6	0.0003207647	1822.43
7	0.0005877474	1944.07

Table 16. Comparison of the proposed ChebyPINN and the conventional PINN with the numerical results reported in Table 1 of Syam et al. [29] for Example 4.

t	HPM	MHPM	SHPM	Numerical Results	RKHSM	Conventional PINN	ChebyPINN
0.5	0.762476	0.768902	0.768766	0.768802	0.768802	0.768394	0.768702
1.0	0.176929	0.233741	0.233680	0.233692	0.233692	0.234031	0.233761
2.0	-1.055110	-0.891260	-0.859323	-0.859349	-0.859349	-0.858033	-0.859047
3.5	-0.461650	-0.079433	-0.093034	-0.093013	-0.093011	-0.094607	-0.093359
5.0	2.049041	0.996472	0.947107	0.947130	0.947130	0.947140	0.947121

Table 17 provides a quantitative comparison between the conventional PINN and the proposed ChebyPINN for Example 4. Under identical training conditions, the proposed ChebyPINN achieves a lower governing-equation residual than the conventional PINN, indicating an improved capability in satisfying the underlying physical constraints of the Duffing system. Although the incorporation of the Chebyshev polynomial layer increases the computational cost and results in a longer training time, the considerable reduction in the residual error highlights the effectiveness of enriching the hidden-layer representation with Chebyshev polynomial features. These results demonstrate the potential of the proposed architecture for obtaining more accurate and physically consistent solutions for strongly nonlinear Duffing equations.

Table 17. Comparison of the proposed ChebyPINN and the conventional PINN in terms of residual MAE and training time.

Model	MAE(Residual)	Training time (s)
ChebyPINN	$1.16005674 \times 10^{-4}$	389.00
Conventional PINN	$5.76105516 \times 10^{-4}$	294.83

5 Discussion

The numerical experiments presented in Section 4 demonstrate that the proposed ChebyPINN consistently outperforms a conventional PINN baseline across a diverse set of Duffing oscillator configurations. This section interprets these results, examines the mechanisms underlying the observed improvements, discusses the sensitivity of the architecture to hyperparameter choices, and situates the findings within the broader landscape of physics-informed scientific machine learning.

The most striking result is the magnitude of accuracy improvement achieved by replacing a conventional pointwise activation with a Chebyshev polynomial basis in the second hidden layer. In Example 1, which admits the exponentially decaying exact solution $u(t) = \frac{1}{2}e^{-t}$, the ChebyPINN achieves a mean absolute error of 1.243×10^{-7} compared with 1.944×10^{-6} for the conventional PINN—an improvement of approximately one order of magnitude (Table 6). More dramatically, in Example 2, where the exact solution $u(t) = \sin(t)$ is explicitly oscillatory, the improvement exceeds *two* orders of magnitude (1.999×10^{-4} versus 6.288×10^{-7}; Table 10).

This disparity in improvement factors between Examples 1 and 2 is not coincidental. It reflects the fundamental theoretical property that motivated the Chebyshev modification: the capacity of Chebyshev polynomials to mitigate spectral bias. As noted by Naveen Raj [17], standard smooth activations such as tanh exhibit a frequency preference that causes high-frequency solution components to be learned far more slowly than low-frequency modes. The Duffing oscillator, particularly in its undamped and forced configurations

Article in press

(Example 2), possesses rich oscillatory content that conventional PINNs struggle to capture efficiently. By contrast, Chebyshev polynomials of the first kind constitute a complete orthogonal basis on $[-1,1]$ with near-minimax approximation properties. Their explicit polynomial structure provides direct representational access to oscillatory modes without relying on the slow accumulation of frequency content through compositional layers. The fact that the largest improvement occurs in Example 2, the most oscillatory of the test cases, provides empirical confirmation that the Chebyshev activation specifically addresses the spectral bias limitation identified in prior PINN literature.

For Examples 3 and 4, where no closed-form analytical solution is available, the governing-equation residual serves as the primary indicator of physical consistency. The ChebyPINN reduces the mean absolute residual by factors of approximately 3 and 5, respectively (Tables 13 and 17). These improvements, while smaller in relative terms than those observed in Examples 1 and 2, are nonetheless significant because they demonstrate that the Chebyshev representation enhances physics consistency even in the absence of exact solutions. The smaller improvement factors in Examples 3 and 4 may be attributed to the additional complexity introduced by strong nonlinearity and, in Example 3, the large initial slope $u'(0) = 5$, which amplifies any residual inaccuracy in the learned solution.

A critical practical consideration in deploying the ChebyPINN is the selection of the Chebyshev polynomial degree d. The experimental results reveal that this choice is problem-dependent and that performance does not improve monotonically with increasing degree.

The investigation of Adam training epochs reveals a consistent pattern across examples: accuracy improves with additional epochs up to an optimal point, beyond which further training induces degradation.

The LBFGS refinement stage plays a crucial role in final accuracy. In Example 1, increasing LBFGS iterations from 1 to 500 reduces the MAE from 1.779×10^{-3} to 1.323×10^{-7} (Table 3), demonstrating that second-order optimization is essential for extracting the full benefit of the Chebyshev representation. Similarly, in Example 3, LBFGS refinement from 100 to 500 iterations produces more than an order-of-magnitude improvement in residual error (Table 12). Beyond approximately 500 iterations, however, returns diminish, suggesting that the ChebyPINN loss landscape is sufficiently well-conditioned to converge within a moderate number of second-order steps once Adam pretraining has reached a favorable basin.

The accuracy improvements of the ChebyPINN come at the cost of increased computational time. Across the four examples, the training time overhead ranges from approximately $1.3\times$ (Example 4) to $3.6\times$ (Example 2) relative to the conventional PINN. This overhead is attributable to the additional operations required for Chebyshev polynomial evaluation via the recurrence relation and the resulting increase in the effective feature dimension passed to the output layer. However, the absolute training times remain modest (under 100 seconds for Examples 1, 2, and 4; under 700 seconds for Example 3).

From a practical perspective, the improvements in approximation accuracy and physical consistency achieved by ChebyPINN suggest that its additional training time can be justified for Duffing problems where accurately capturing nonlinear oscillatory responses is essential for characterizing system dynamics. Conversely, conventional PINNs may remain preferable when computational speed is the primary concern and their approximation accuracy is sufficient.

6 Conclusion

This study introduced a Chebyshev-based physics-informed neural network for solving nonlinear Duffing equations. The proposed framework incorporates Chebyshev polynomial activation function into the second hidden layer of a conventional PINN architecture to enhance its approximation capability for nonlinear solution spaces. The effectiveness of the proposed approach was investigated through several Duffing equation cases, including different nonlinear behaviors, and compared with a conventional PINN model using comparable network depths and training conditions.

The obtained results demonstrate that the proposed ChebyPINN consistently provides improved accuracy compared with the standard PINN architecture. The incorporation of Chebyshev polynomial activation reduced the governing-equation residuals and improved the prediction accuracy in the considered examples, indicating that the Chebyshev polynomial activation has the potential to enhance the representation capability of physics-informed neural networks for nonlinear dynamical systems. Furthermore, the analysis of the Chebyshev polynomial degree showed that an appropriate selection of the polynomial order plays an important role in achieving accurate and stable solutions.

Authors' Contributions

The contributions of each author to this study are as follows:

Article in press

- Fatemeh Nikzaddinan conceptualized the research design, developed the proposed Chebyshev-enhanced physics-informed neural network architecture, implemented the algorithms and numerical experiments, performed the validation and sensitivity analyses, generated the figures and tables, and drafted the original manuscript.

- Hassan Dana Mazraeh supervised the research, contributed to the mathematical formulation and theoretical analysis of the Chebyshev polynomial activation mechanism, verified the numerical results, critically reviewed and edited the manuscript, managed the project administration, drafted the original manuscript, and submission process.

Both authors have read and approved the final version of the manuscript and agree to be accountable for all aspects of the work.

Data Availability

The manuscript has no associated data or the data will not be deposited.

Code Availability

The steps for implementing the method described in this paper are self-explanatory. Sample code will be made available after publication upon reasonable request to the corresponding author by email.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Considerations

This study is a theoretical and computational work based on mathematical modeling, numerical simulation, and machine learning algorithms. It does not involve human participants, animal subjects, biological materials, or any personally identifiable or sensitive data. Consequently, ethical approval from an institutional review board or ethics committee was not required for this research. The authors have nevertheless diligently addressed ethical concerns, including plagiarism, data fabrication, misconduct, falsification, duplicate publication, redundancy, and submission to multiple journals, and confirm that all applicable ethical standards have been fully observed throughout the conduct of this study and the preparation of this manuscript.

Funding

The authors have not received any funding for the completion of this study.

Acknowledgments

The authors would like to express their sincere gratitude to the anonymous reviewers for their careful reading of the manuscript and for their valuable comments. The authors also wish to thank the editor for their thoughtful guidance and handling of the manuscript throughout the review process.

References

[1] V. Murugesh, M. Priyadharshini, T. R. Mahesh, and E. A. Eslema, A thermodynamic inspired AI based search algorithm for solving ordinary differential equations, Scientific Reports, 15(1), 18141, (2025).

[2] M. V. B. Santana, Exact solutions of nonlinear second-order autonomous ordinary differential equations: Application to mechanical systems, Dynamics, 3(3), (2023).

Article in press

[3] G. M. Ismail, G. M. Moatimid, and S. V. Kontomaris, Approximate analytical solutions of nonlinear jerk equations using the parameter expansion method, Computation, 14(1), (2026).

[4] A. H. Salas Salas, J. E. Castillo Hernández, and L. J. Martínez Hernández, The duffing oscillator equation and its applications in physics, Mathematical Problems in Engineering, 2021(1), 9994967, (2021).

[5] M. Merino-Olagüe, X. Iriarte, C. Castellano-Aldave, and A. Plaza, Hybrid modelling and identification of mechanical systems using physics-enhanced machine learning, Engineering Applications of Artificial Intelligence, 159(1), 111762, (2025).

[6] M. Raissi, P. Perdikaris, and G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, 378(1), 686–707, (2019).

[7] M. Movahedian Moghaddam, H. Dana Mazraeh, and K. Parand, Symbolic solution of fractional volterra population models via physics-informed neural networks and distributed particle swarm optimization, Applied Soft Computing, 189(1), 114512, (2026).

[8] M. M. Moghaddam, H. Dana Mazraeh, and K. Parand, A double deep q network-powered approach to discovering symbolic solutions for nonlinear integral equations, Engineering Applications of Artificial Intelligence, 169(1), 114113, (2026).

[9] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, Nature Machine Intelligence, 3(1), 218229, (2021).

[10] M. Movahedian Moghaddam, H. Dana Mazraeh, and K. Parand, Symbolic solution of fractional volterra population models via physics-informed neural networks and distributed particle swarm optimization, Applied Soft Computing, 189(1), 114512, (2026).

[11] M. Naser, Fundamental flaws of physics-informed neural networks and explainability methods in engineering systems, Computers & Industrial Engineering, 212(1), 111704, (2026).

[12] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics, 404(1), 109136, (2020).

[13] A. Farea and M. S. Celebi, Learnable activation functions in physics-informed neural networks for solving partial differential equations, Computer Physics Communications, 315(1), 109753, (2025).

[14] N. A. Khan, T. Hameed, O. A. Razzaq, and M. Ayaz, Intelligent computing for duffing-harmonic oscillator equation via the bio-evolutionary optimization algorithm, Journal of Low Frequency Noise, Vibration and Active Control, 38(1), (2018).

[15] T. V. Lazovskaya, G. F. Malykhina, D. M. Pashkovskiy, and D. A. Tarkhov, Polynomial neural network approximation of duffing equation solution, in Cyber-Physical Systems and Control II (D. G. Arseniev and N. Aouf, eds.), (Cham), 554–562, Springer International Publishing, 2023.

[16] A. Verma, W. Sumelka, and P. K. Yadav, The numerical solution of nonlinear fractional lienard and duffing equations using orthogonal perceptron, Symmetry, 15(9), (2023).

[17] R. Naveen Raj, Physics informed neural network forăsolution ofăduffing oscillators, in Proceedings of the 2nd International Conference on Nonlinear Dynamics and Applications (ICNDA 2024), Volume 3 (A. Saha and S. Banerjee, eds.), (Cham), 164–172, Springer Nature Switzerland, 2024.

[18] A. A. Aghaei, A. G. Nejad, S. Yousefi, and K. Parand, An efficient spectral lssvm framework for solving systems of nonlinear fractional integro-differential equations, Mathematics and Computers in Simulation, 250(1), 1085–1110, (2026).

[19] K. Parand and M. Razzaghi, Rational chebyshev tau method for solving volterras population model, Applied Mathematics and Computation, 149(3), 893–900, (2004).

[20] M. Kyselov and S. G. Solodky, Numerical differentiation of functions of two variables using chebyshev polynomials, arXiv, 1(1), (2026).

Article in press

[21] A. Afzal Aghaei, A. Ghasem Nejad, S. Yousefi, and K. Parand, A new chebyshev operational matrix formulation of least-squares support vector regression for solving fractional integro-differential equations, Journal of Computational and Applied Mathematics, 474(1), 116816, (2026).

[22] F. Mostajeran and S. A. Faroughi, Elastoplasticity informed kolmogorovarnold networks using chebyshev polynomials, International Journal for Numerical and Analytical Methods in Geomechanics, 50(8), 3331–3354, (2026).

[23] H. Dana Mazraeh and K. Parand, Gepinn: An innovative hybrid method for a symbolic solution to the laneemden type equation based on grammatical evolution and physics-informed neural networks, Astronomy and Computing, 48(1), 100846, (2024).

[24] J. C. Mason and D. C. Handscomb, Chebyshev polynomials. New York: Chapman and Hall/CRC, 2002.

[25] K. Parand and M. Shahini, Rational chebyshev pseudospectral approach for solving thomasfermi equation, Physics Letters A, 373(1), 210–213, (2009).

[26] H. Ammari, S. Barandun, and P. Liu, Applications of chebyshev polynomials and toeplitz theory to topological metamaterials, Reviews in Physics, 13(1), 100103, (2025).

[27] G. Adomian, Solving Frontier Problems of Physics: The Decomposition Method. Boston: Kluwer Academic Publishers, 1994.

[28] M. AL-Jawary and S. Abd-AL-Razaq, Analytic and numerical solution for duffing equations, International Journal of Basic and Applied Sciences, 5(2), 115119, (2016).

[29] M. I. Syam, M. A. Raja, M. M. Syam, and H. M. Jaradat, An accurate method for solving the undamped duffing equation with cubic nonlinearity, International Journal of Applied and Computational Mathematics, 4(2), 16, (2018).

Article in press