



A Hybrid Method Based on Rational Legendre Physics Informed Neural Networks and Cuckoo Optimization Algorithm for Solving Nonlinear Inverse Partial Differential Equations

Farhad Fahimi¹, Reza Pourgholi^{1,*}, Abdolali Basiri¹

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

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

Received: 29/04/2026 Revised: 20/07/2026 Accepted: 23/07/2026 Published: 04/09/2026

10.22128/ansne.2026.3307.1213

Abstract

We develop an improved physics-informed neural network (PINN) framework for solving nonlinear inverse partial differential equations. Two separate network architectures are explored: one utilizing Legendre polynomials as activation functions and another employing rational Legendre polynomials. In the latter case, there is a crucial hyperparameter, L , that significantly impacts both performance and efficiency. To optimize this parameter, we employ the cuckoo optimization algorithm (COA), which fine-tunes L based on the problem's characteristics. The proposed methods are tested on two benchmark problems, the Fisher equation and the Burgers-Fisher equation. The results highlight the effectiveness of both approaches, demonstrating superior accuracy in reconstructing unknown boundary conditions. All experiments were conducted in a Jupyter Notebook environment using PyTorch, confirming the potential of tailored activation functions and optimization techniques in physics-informed deep learning.

Keywords: Physics-informed neural networks, Nonlinear inverse partial differential equations, Rational Legendre polynomials, Cuckoo optimization algorithm, Fisher equation, Burgers-Fisher equation.

Mathematics Subject Classification (2020): 35K05, 65M20

1 Introduction

Partial differential equations (PDEs) model phenomena in heat conduction, fluid dynamics, electromagnetics, biology, and many other fields [1, 2]. In a direct (forward) problem, the governing equation together with its initial and boundary data is fully specified. In contrast, a nonlinear inverse PDE (NIPDE) problem seeks unknown model components (often a boundary function or a parameter) from limited interior measurements. Inverse problems arise in heat conduction monitoring, contaminant transport, medical imaging, and system identification [3, 4]. They are frequently ill-posed: small measurement errors can produce large reconstruction errors. Classical remedies include Tikhonov and iterative regularization, basis-function methods [5], Haar wavelet techniques [6], and function-specification approaches [7]. These tools are well developed for linear inverse problems but are less effective for nonlinear NIPDEs. Nonlinear PDEs are substantially harder than their linear counterparts because superposition no longer holds and closed-form solutions are rare. This has motivated finite-difference [8], finite-element [9], spectral, and machine-learning solvers. Within machine learning, physics-informed neural

networks (PINNs) embed PDE residuals in the training loss and have become a flexible tool for both forward and inverse problems [10, 11]. Metaheuristic optimizers—including genetic algorithms [4, 12], particle swarm optimization [13], imperialist competitive algorithms [14], the cuckoo optimization algorithm (COA) [15, 16], and water-strider algorithms [17]—have also been used to estimate unknown parameters in inverse models. This work proposes two PINN variants for NIPDEs:

1. **Legendre PINN (LPINN)**: Legendre polynomials $P_0, \dots, P_9$ serve as activations in the final hidden layer.
2. **Rational Legendre PINN (RLPINN)**: rational Legendre polynomials with mapping parameter L are used in the same layer; L is tuned by COA.

COA is a population-based metaheuristic inspired by brood parasitism in cuckoo birds [15, 16]. In our setting it searches for the L value that minimizes the PINN loss after a fixed training budget. The methods are evaluated on two benchmark NIPDEs:

- the Fisher equation [18];
- the Burgers–Fisher equation [19, 20].

Results are compared with a standard tanh PINN and with the improved teaching–learning-based optimization (ITLBO) method of [21]. The proposed rational Legendre PINN with COA (RLPINN) yields the smallest reconstruction errors for the unknown boundary functions. All experiments were implemented in PyTorch within a Jupyter Notebook environment. This study shows that polynomial activations matched to the solution structure, together with metaheuristic tuning of L , can improve PINN accuracy on nonlinear inverse parabolic problems.

The rest of this paper is structured as follows. Section 2 formulates nonlinear inverse partial differential equation problems and their role in applications. Section 3 presents the proposed rational Legendre PINN framework for NIPDEs: it reviews physics-informed neural networks and the loss formulation, and explains Legendre and Rational Legendre polynomials and their use in the network. Section 4 describes how Legendre and Rational Legendre activations are incorporated into PINNs. Section 5 explains optimization of the parameter L in Rational Legendre polynomials using the cuckoo optimization algorithm. Section 6 gives a discussion and numerical comparison of the methods on the Fisher and Burgers–Fisher equations. Section 7 concludes with the main findings and the advantages of the proposed approach over existing methods, including ITLBO.

2 Nonlinear Inverse Partial Differential Equation Problems

PDEs are mathematical models involving partial derivatives of an unknown function with respect to multiple independent variables. They can be expressed generally as:

$$F\left(x_1, x_2, \dots, x_n, U, \frac{\partial U}{\partial x_1}, \frac{\partial U}{\partial x_n}, \frac{\partial^2 U}{\partial x_1^2}, \frac{\partial^2 U}{\partial x_2 \partial x_1}, \dots, \frac{\partial^k U}{\partial x_i \partial x_n}\right) = 0. \quad (1)$$

Here $U(x_1, x_2, \dots, x_n)$ denotes the unknown function. A PDE is linear when U and its derivatives enter F linearly; otherwise it is nonlinear. Representative nonlinear models include the Fisher and Burgers–Fisher equations [22, 23]. In a forward problem, all coefficients, initial data $f(x)$, and boundary data are prescribed and one seeks U . In the NIPDE setting addressed here, the governing equation and most initial/boundary data are known, but *one* boundary function remains unknown and must be recovered from interior observations.

In this study, we address nonlinear inverse problems associated with PDEs, represented in the following form:

$$U_t(x, t) = \phi(x, t, U, U_x, U_{xx}), \quad x \in [0, 1], \quad t \in [0, T], \quad (2)$$

where ϕ denotes a nonlinear expression involving U, U_x and U_{xx} its derivatives. The problem is supplemented by the initial condition:

$$U(x, 0) = f(x), \quad x \in [0, 1], \quad (3)$$

as well as boundary conditions:

$$\begin{aligned} U(0, t) &= p(t), \quad t \in [0, T], \\ U(1, t) &= q(t), \quad t \in [0, T]. \end{aligned} \quad (4)$$

Additionally, the problem includes an over-specified condition:

$$U(x_l, t) = s(t), \quad x_l \in (0, 1), \quad t \in [0, T]. \quad (5)$$

Here $f(x)$ is prescribed, T is the final time, and $s(t)$ denotes noise-free sensor readings at x_l . In the inverse problems below, *one* of the boundary functions $p(t)$ or $q(t)$ is unknown and is reconstructed from (5); the other boundary function, together with $f(x)$, is treated as known input data. When (4) is written with both $p(t)$ and $q(t)$ present, this denotes the generic form; the unknown term is removed from the PINN loss for the corresponding test case (Section 3).

In this study, we use the division of the interval into the desired number of points to discretize the problem.

$$\begin{aligned} x_i &= ih, \quad i = 0, 1, \dots, N, \quad h = \frac{1}{N}, \\ t_j &= jk, \quad j = 0, 1, \dots, M, \quad k = \frac{T}{M}. \end{aligned} \quad (6)$$

As shown in Figure 1, a sensor placed at $x_l \in (0, 1)$ records $U(x_l, t)$ for $t \in [0, T]$. These readings constitute the observed (over-specified) data. In the numerical experiments, synthetic observations are generated from the exact solution, i.e. $U(x_l, t_j)$ with $t_j = jk$, $j = 1, \dots, M$.

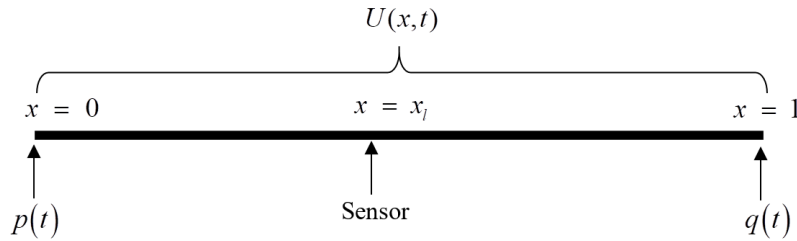


Figure 1. A one-dimensional heat conduction problem.

We study two benchmark NIPDEs from [21]: the Fisher equation (biology, combustion, heat and mass transfer, ecology [24]) and the Burgers–Fisher equation (wave phenomena in gas dynamics, acoustics, and hydrodynamics [25]). In each case the initial condition $f(x)$ and one boundary function are known, while the opposite boundary function is unknown and recovered from interior data at $x_l = 0.4$.

The Fisher inverse problem reads

$$U_t(x, t) = U_{xx}(x, t) + aU(x, t)(1 - U(x, t)), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \quad (7)$$

$$U(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad (8)$$

$$U(0, t) = p(t), \quad 0 \leq t \leq 1, \quad (9)$$

$$U(1, t) = q(t), \quad 0 \leq t \leq 1, \quad (10)$$

with reaction coefficient $a > 0$ (the Fisher reaction parameter). In the numerical experiments below we set $a = 1$, consistent with [21]. Here $p(t)$ and $f(x)$ are prescribed and $q(t)$ is unknown.

The Burgers–Fisher inverse problem is

$$U_t(x, t) = U_{xx}(x, t) + U(x, t)U_x(x, t) + F(x, t), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \quad (11)$$

$$U(x, 0) = f(x), \quad 0 \leq x \leq 1, \quad (12)$$

$$U(0, t) = p(t), \quad 0 \leq t \leq 1, \quad (13)$$

$$U(1, t) = q(t), \quad 0 \leq t \leq 1, \quad (14)$$

where $F(x, t)$ is a known source term. In the numerical test below we take $F(x, t) = U(x, t)(1 - U(x, t))$, as in [21]. Here $q(t)$ and $f(x)$ are known while $p(t)$ is unknown.

3 Rational Legendre Physics-Informed Neural Networks for Solving Nonlinear Inverse Partial Differential Equations

In this section, we present our proposed method for solving NIPDEs. We use Rational Legendre polynomials as activation functions in the last hidden layer of our neural network architecture. Before detailing our proposed architecture, we first introduce the main concepts behind PINNs. In a separate subsection, we provide a comprehensive explanation of our architecture.

3.1 Physics-Informed Neural Networks

PINNs are an innovative computational approach that integrates neural networks with the underlying physics of a system, expressed in the form of PDEs. Unlike traditional numerical methods that rely on discretization of the spatial and temporal domains, PINNs embed the governing physical laws into the neural network training process by incorporating the residuals of PDEs into the loss function. This approach allows the network to simultaneously learn from limited observational data and satisfy the governing equations of the system, offering a flexible and efficient solution framework for both direct and inverse problems [10, 11].

The general form of the loss function for a second-order PDEs in PINNs, defined for $0 \leq x \leq 1$ and $t \geq 0$, is as follows:

$$Loss = \lambda_1 Loss_{PDE} + \lambda_2 Loss_{Initial} + \lambda_3 Loss_{Boundary_{-1}} + \lambda_4 Loss_{Boundary_{-2}} + \lambda_5 Loss_{Data}, \quad (15)$$

$$Loss_{PDE} = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M \left[F(x_i, t_j, U(x_i, t_j), \frac{\partial U(x_i, t_j)}{\partial x_i}, \frac{\partial U(x_i, t_j)}{\partial t_j}, \frac{\partial^2 U(x_i, t_j)}{\partial x_i^2}, \frac{\partial^2 U(x_i, t_j)}{\partial x_i \partial t_j}, \frac{\partial^2 U(x_i, t_j)}{\partial t_j^2}) \right]^2, \quad (16)$$

$$Loss_{Initial} = \frac{1}{N} \sum_{i=1}^N [U(x_i, 0) - f(x_i)]^2, \quad (17)$$

$$Loss_{Boundary_{-1}} = \frac{1}{M} \sum_{j=1}^M [U(0, t_j) - p(t_j)]^2, \quad (18)$$

$$Loss_{Boundary_{-2}} = \frac{1}{M} \sum_{j=1}^M [U(1, t_j) - q(t_j)]^2, \quad (19)$$

$$Loss_{Data} = \frac{1}{M} \sum_{j=1}^M [U(x_l, t_j) - s(t_j)]^2.$$

In the Fisher equation, given in Eq. (7), we assume that the second boundary condition (Eq. (10)) is unknown. This leads to an inverse problem. In the absence of $Boundary_{-2}$, the loss function for the Fisher equation is defined as:

$$Loss = \lambda_1 Loss_{PDE} + \lambda_2 Loss_{Initial} + \lambda_3 Loss_{Boundary_{-1}} + \lambda_4 Loss_{Data}. \quad (20)$$

In the Burgers-Fisher equation (Eq. (11)), we assume that the first boundary condition (Eq. (13)) is unknown, which results in an inverse problem. In the absence of $Boundary_{-1}$, the loss function for the Burgers-Fisher equation is defined as:

$$Loss = \lambda_1 Loss_{PDE} + \lambda_2 Loss_{Initial} + \lambda_3 Loss_{Boundary_{-2}} + \lambda_4 Loss_{Data}. \quad (21)$$

Here, $Loss_{PDE}$ measures the PDE residual at collocation points, $Loss_{Initial}$ enforces the initial condition, $Loss_{Boundary_{-1}}$ and $Loss_{Boundary_{-2}}$ enforce the known boundary data, and $Loss_{Data}$ matches the interior sensor readings. For inverse problems, the term corresponding to the unknown boundary is removed from the loss. Figure 2 shows the baseline PINN architecture: three hidden layers with 10, 20, and 10 neurons, tanh activations in the first two hidden layers, and a scalar output layer. Hyperbolic tangent is smooth and supports stable automatic differentiation of the PDE residual.

For LPINN and RLPINN, the final hidden layer contains 10 neurons. Each neuron applies a distinct polynomial activation to its pre-activation scalar input z :

$$\text{neuron } k \text{ uses } P_k(z) \quad (k = 0, \dots, 9), \quad (22)$$

where P_k is either the standard Legendre polynomial on $[-1, 1]$ or the rational Legendre polynomial $R_k(z) = P_k(\frac{z-L}{z+L})$. Thus degrees 0–9 are used, one per node. The width 10 was chosen to match the preceding layer and to provide a low-order spectral basis without

over-parameterizing the network; preliminary trials with five nodes (degrees 0–4) gave larger boundary errors, whereas adding an eleventh node (degree 10) did not improve the reported MAE. The mapping parameter L in RLPINN is optimized by COA as described in Section 5.

For the inverse problems considered, one boundary term is omitted from the loss (Eqs. (20)–(21)) while the interior sensor data enforce the missing boundary information.

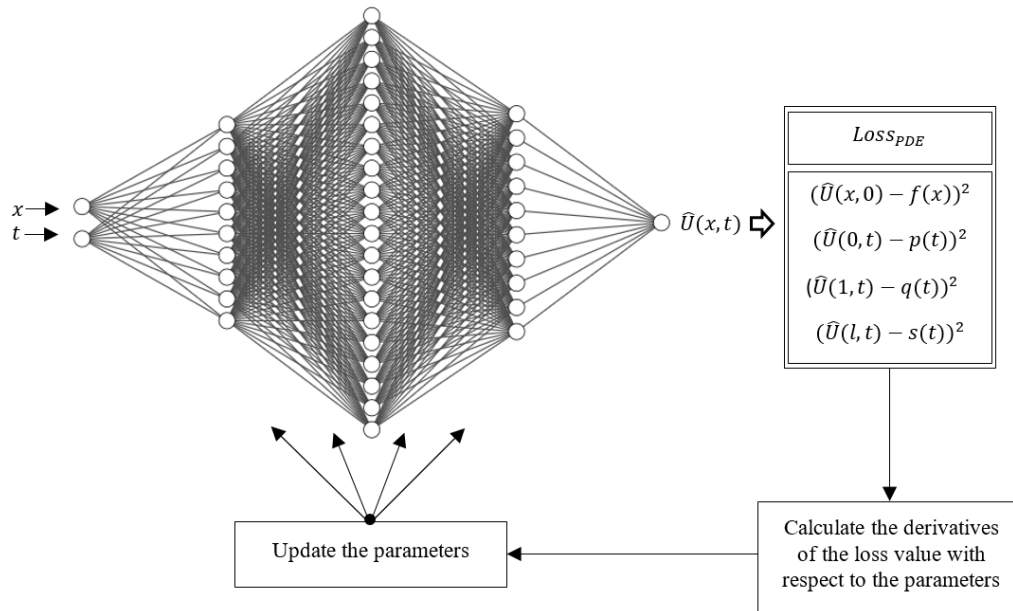


Figure 2. The architecture of the PINNs.

3.2 Detailed Explanation of Legendre and Rational Legendre Neural Networks

Legendre polynomials, denoted $P_n(x)$, form a set of orthogonal polynomials defined over the interval $[-1, 1]$. These polynomials satisfy the Legendre differential equation:

$$\frac{d}{dx}(1-x^2) \frac{dP_n(x)}{dx} + n(n+1)P_n(x) = 0, \quad (23)$$

where n is the polynomial order. Their orthogonality is given with respect to the weight function $w(x) = 1$:

$$\int_{-1}^1 P_n(x) P_m(x) dx = 0 \quad \text{for } n \neq m. \quad (24)$$

They can be derived explicitly using Rodrigues' formula:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n. \quad (25)$$

These polynomials are widely applied in numerical analysis, physics, and engineering due to their orthogonality and ability to approximate functions over finite intervals. They are frequently used in solving PDEs in spherical coordinates and in spectral methods for numerical computation. The first few Legendre polynomials are:

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{1}{2}(3x^2 - 1), \quad P_3(x) = \frac{1}{2}(5x^3 - 3x). \quad (26)$$

Higher-order polynomials are generated using the recurrence relation:

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x). \quad (27)$$

Rational Legendre polynomials are introduced to address problems defined on semi-infinite domains such as $[0, \infty)$, where standard Legendre polynomials may not be ideal. $R_n(x)$ are designed for such cases, mapping $[0, \infty)$ to $[-1, 1]$ using the transformation:

$$R_n(x) = P_n\left(\frac{x-L}{x+L}\right), \quad (28)$$

here, L is a constant real value that should be properly defined based on the problem under consideration. These polynomials retain the orthogonality property but are defined with a weight function $w(x) = (x+L)^{-2}$:

$$\int_0^{\infty} R_n(x)R_m(x)w(x)dx = 0 \quad \text{for } n \neq m. \quad (29)$$

The orthogonality makes them particularly useful in numerical solutions of differential equations over unbounded domains, as they provide efficient spectral approximations.

Legendre and Rational Legendre polynomials are widely used in solving boundary value problems. Due to their fractional structure, Rational Legendre polynomials offer enhanced capabilities in approximating functions with complex behavior compared to standard Legendre polynomials. This enables them to capture rapid variations and intricate features of functions with higher accuracy. For example, the researchers in [26] demonstrated that employing rational basis functions in spectral methods results in more precise and efficient approximations of complex functions, particularly for problems exhibiting nonlinear behavior and sharp transitions. Similarly, in [27], the authors highlighted that Rational Legendre polynomials, owing to their fractional nature, provide improved accuracy in capturing functional complexity and outperform standard Legendre polynomials in numerical solutions. Nonetheless, the choice between these two types of polynomials depends on the specific characteristics of the problem being addressed.

Rational Legendre polynomials are classical tools on semi-infinite domains $[0, \infty)$, where the map $\xi = (x-L)/(x+L)$ sends $[0, \infty)$ to $[-1, 1]$ [26, 27]. Although the physical space in our benchmarks is the bounded interval $[0, 1]$, the activations are *not* evaluated at the spatial coordinate x ; they act on bounded pre-activation values z produced by the preceding linear layer. The rational map therefore reshapes the nonlinear basis on a bounded argument range, and the parameter $L > 0$ controls how aggressively that map compresses large positive or negative z . Optimizing L by COA adapts this compression to the Fisher and Burgers–Fisher profiles, which exhibit steep gradients and traveling-wave structure. In practice, CRLPINN outperformed both the tanh PINN and LPINN on the reported metrics, even though the spatial domain is bounded.

Standard Legendre activations are appropriate when the pre-activation distribution is well represented on $[-1, 1]$:

1. bounded argument range;
2. relatively smooth local behavior [28].

Rational Legendre activations are advantageous when

1. the solution profile contains sharp transitions or asymmetric decay;
2. tuning L improves the fit of the rational map to the learned features.

4 Using Legendre and Rational Legendre Polynomials as Activation Functions in Physics-Informed Neural Networks

In this study, we propose a novel architecture for PINNs by introducing customized activation functions based on Legendre and Rational Legendre polynomials in the final hidden layer. The architecture is designed to enhance the capability of PINNs in solving complex NIPDEs.

4.1 Design and Motivation

The network retains the general structure of PINNs, where the governing partial differential equations are embedded into the loss function to ensure adherence to physical laws. However, in the final hidden layer, we replace the standard activation functions with tailored Legendre and Rational Legendre activations. These polynomials are particularly well-suited for representing solutions over finite and semi-infinite domains, respectively, due to their orthogonality and efficient spectral approximation properties.

1. **Legendre activation:** This activation is used for domains with finite boundaries. Its orthogonality ensures minimal cross-term interference, enabling accurate representation of complex solution patterns.
2. **Rational Legendre activation:** For semi-infinite domains, the Rational Legendre polynomial provides a robust transformation, efficiently mapping the infinite domain to a finite interval.

By leveraging these activations, the network adapts more naturally to the problem's spatial characteristics, reducing the approximation error and improving convergence.

4.2 Benefits of Custom Activations

1. **Enhanced approximation:** The polynomial basis functions expand the network's representational capacity, allowing it to capture intricate patterns in the solution space that might elude traditional activation functions like tanh or ReLU.
2. **Improved convergence:** Incorporating domain-specific knowledge through these activations accelerates the training process, as the network starts closer to the true solution.
3. **Flexibility in problem domains:** The use of Rational Legendre polynomials, with a tunable parameter for mapping domains, offers flexibility in handling problems across a wide range of physical applications.

4.3 Applications and Results

This approach has been tested on benchmark problems such as the Fisher equation and the Burgers-Fisher equation, demonstrating significant improvements in accuracy and computational efficiency. By combining this activation strategy with the inherent physics-aware framework of PINNs, we achieve robust solutions even in scenarios with limited observational data. The previous PINNs changes as shown in Figure 3.

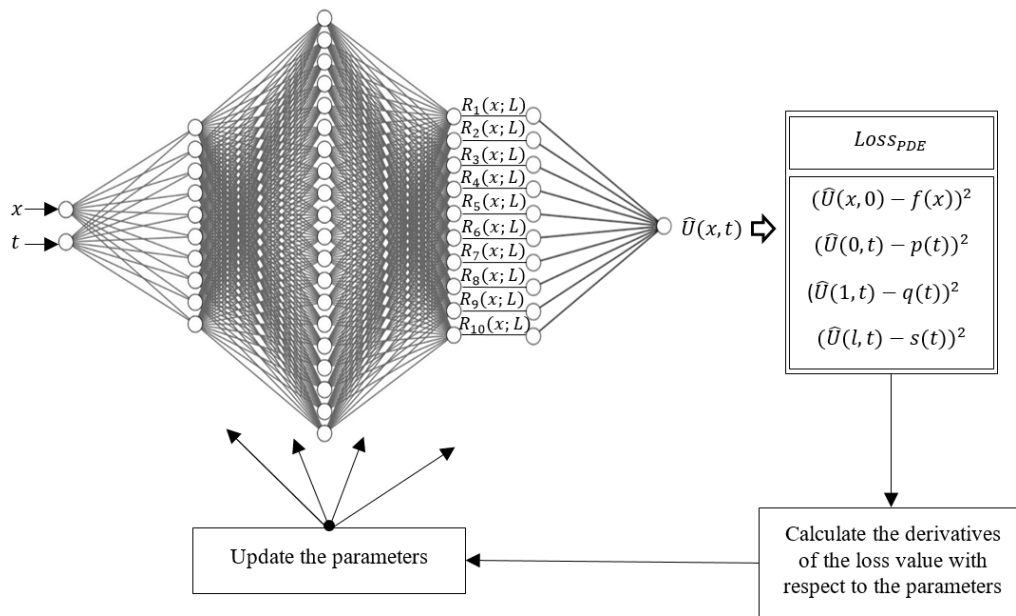


Figure 3. The architecture of the PINNs with a Rational Legendre activation function.

5 Optimization of the Parameter L in Rational Legendre Polynomials using the COA

To improve RLPINN accuracy, the mapping parameter L in the rational Legendre activations is optimized by COA. The rational map $\xi = (z - L)/(z + L)$ sends $(-L, \infty)$ to $(-1, 1)$; the choice of L therefore controls how pre-activation values are compressed before the orthogonal polynomials are evaluated.

5.1 Cuckoo Optimization Algorithm

The cuckoo optimization algorithm is an evolutionary computation method inspired by the unique reproductive behavior of cuckoo birds. In nature, certain cuckoo species lay their eggs in the nests of other bird species, relying on the host birds to incubate and raise their offspring. This parasitic strategy involves precise mimicry of eggs to avoid detection, making it an intriguing model for optimization. COA models this behavior by considering solutions as "habitats" where cuckoos (candidate solutions) attempt to maximize the survival of their eggs (optimization variables). The algorithm begins with an initial population of solutions, where each "cuckoo" lays a certain number of eggs randomly within its "Egg Laying Radius". These eggs represent new potential solutions, and their survival depends on their "fitness," evaluated by the objective function. Key steps in the COA include:

1. **Egg laying and survival:** A proportion of the least-fit eggs are eliminated, simulating natural selection. The remaining solutions represent more promising regions of the search space.
2. **Migration:** Groups of cuckoos migrate toward better habitats (solutions with higher fitness). This migration is guided by probabilistic mechanisms to maintain exploration and prevent premature convergence.
3. **Convergence:** Over successive generations, the population converges toward the optimal habitat, representing the global optimum of the problem.

The COA has emerged as a powerful metaheuristic tool for solving complex optimization problems, particularly those involving nonlinear and high-dimensional spaces. Inspired by the brood parasitism behavior of cuckoo birds and their adaptive search for optimal nesting locations, COA strikes an effective balance between exploration (global search) and exploitation (local refinement) in the solution space. One of the key advantages of COA is its ability to avoid premature convergence to local optima. Through its stochastic nature and Lévy flight-based global search mechanism, it exhibits strong robustness in escaping local minima and ensuring a more comprehensive exploration of the search domain [29]. This makes it especially suitable for nonlinear inverse problems, where the objective landscape is often rugged and multi-modal. Moreover, COA features a simple and intuitive structure, with fewer tuning parameters compared to other metaheuristic algorithms such as particle swarm optimization and genetic algorithms. This ease of implementation not only reduces computational overhead but also enhances its adaptability across various problem domains [15, 16, 30]. Recent studies have demonstrated the superior performance of COA in benchmark tests and real-world applications. For instance, COA has shown better convergence rates and solution quality in engineering design, scheduling, structural optimization, and control systems, outperforming algorithms like artificial bee colony and firefly algorithm in several metrics [31, 32]. In our study, COA was chosen due to its effectiveness in handling the nonlinear nature of the inverse partial differential equations under investigation. Specifically, its ability to efficiently explore complex search spaces and identify accurate parameter estimations contributed significantly to the precision and robustness of the proposed numerical method. The algorithm's biological mimicry ensures both global diversity and local intensification, which are essential for reliable convergence in high-dimensional and nonlinear problem settings. The flowchart of the COA is illustrated in Figure 4. This diagram provides an overview of the algorithmic procedure, including population initialization, objective function evaluation, generation of new solutions, and updating of nests.

5.2 COA for Hyperparameter L Optimization in Physics-Informed Neural Networks

The optimization process integrates the COA into the training of the PINNs as follows:

1. **Training the neural networks:** The PINNs is trained for 1000 epochs for each candidate L value to ensure a stable and accurate assessment of the loss function, which combines residuals of the governing PDEs, boundary conditions, and observed data.
2. **Cuckoo optimization algorithm:** The COA runs for 100 iterations to iteratively search for the optimal L . The COA is initialized with a population of candidate solutions (values of L), which are treated as the habitats of cuckoo agents. The fitness of each candidate is evaluated based on the loss function of the CRLPINNs.
3. **Egg laying and survival:** During each iteration, cuckoos generate new solutions (new L values) within their egg laying radius. The solutions with the least fitness are eliminated, mimicking natural selection.

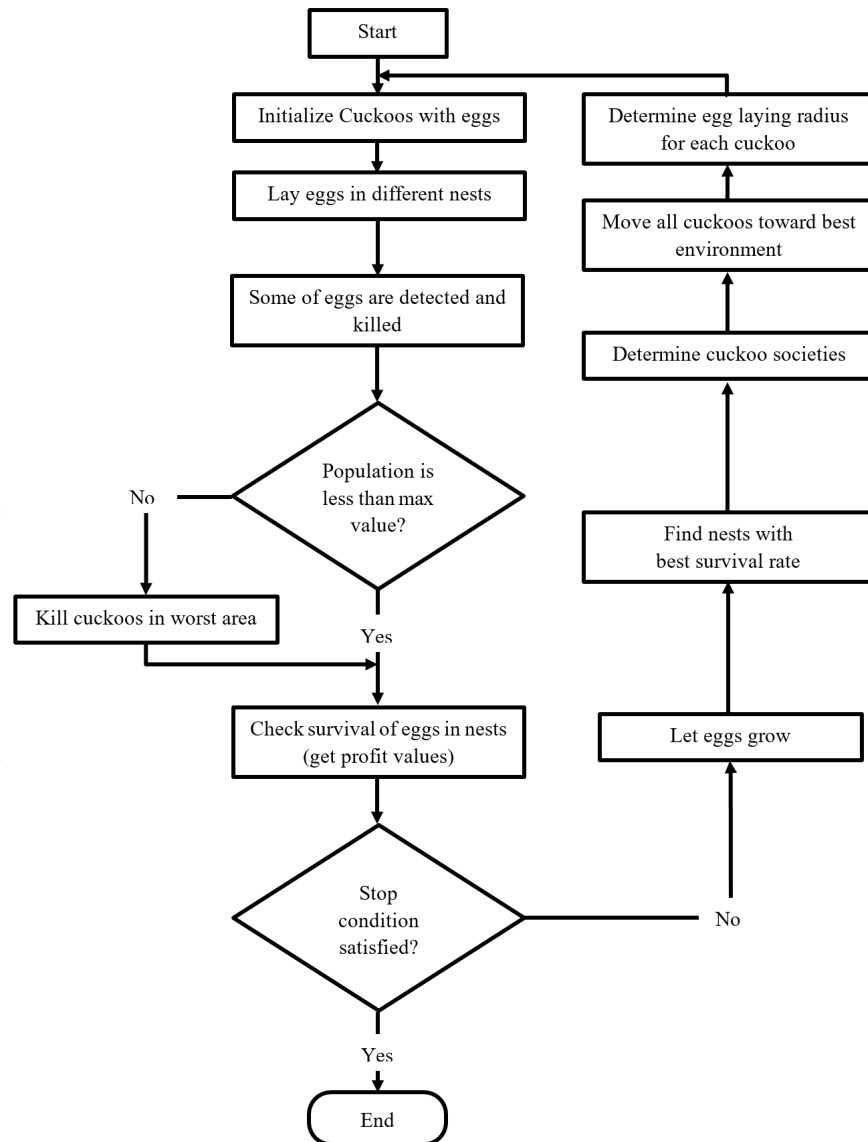


Figure 4. Flowchart of the cuckoo optimization algorithm.

4. **Migration:** Cuckoos migrate towards better habitats, guided by probabilistic mechanisms, ensuring a balance between local exploitation and global exploration.
5. **Optimal L selection:** After 100 iterations, the L value that minimizes the loss function is selected as the optimal parameter for the Rational Legendre polynomials.

By combining the robustness of COA in exploring high-dimensional solution spaces with the flexibility of Rational Legendre polynomials in approximating semi-infinite domains, this approach achieves efficient and accurate parameter tuning. The optimized L significantly improves the convergence and precision of the PINNs in solving complex NIPDEs. The flowchart illustrating the use of the COA for finding the hyperparameter L in PINNs for solving PDEs is shown in Figure 4. This flowchart outlines the steps involved in optimizing the hyperparameter and the weight update process in the neural network.

6 Discussion and Numerical Comparison

In this section, we compare the performance of three configurations of PINNs with the ITLBO method from [21]. The focus is on solving NIPDEs, specifically the Fisher and Burgers–Fisher equations, by reconstructing the unknown boundary conditions $q(t)$ and $p(t)$, respectively.

6.1 Methods for Comparison

The three configurations of PINNs considered are as follows:

1. **Standard PINNs:** A baseline PINNs using standard activation functions (e.g., $\tanh$).
2. **Legendre PINNs:** A modified PINNs incorporating Legendre polynomial-based activation functions.
3. **Rational Legendre PINNs (optimized):** A PINNs utilizing Rational Legendre polynomial-based activation functions, where the optimal parameter L is determined via the COA. The COA is used exclusively to optimize L , ensuring the Rational Legendre polynomials effectively adapt to the problem's semi-infinite domain.

These PINNs-based methods are evaluated alongside the ITLBO algorithm, which combines a numerical solver with an optimization method to estimate the unknown boundary conditions $q(t)$ or $p(t)$.

6.2 Problem Setup

- **Fisher equation:** The goal is to estimate the unknown boundary condition $q(t)$ using the governing equation, initial conditions, and over-specified interior condition.
- **Burgers–Fisher equation:** The goal is to estimate the unknown boundary condition $p(t)$ using similar conditions.

To ensure a fair and consistent comparison, the following setup is applied:

- The PINNs-based methods are trained for **1000 epochs** to ensure convergence and optimal performance.
- For the Rational Legendre PINNs, the COA performs **100 iterations** to identify the best L parameter, which is subsequently used during training.
- All methods solve the Fisher and Burgers–Fisher equations using identical boundary and initial conditions.

6.3 Comparative Metrics

The performance of the methods is evaluated based on:

1. Accuracy of $q(t)$ in the Fisher equation and $p(t)$ in the Burgers–Fisher equation: A numerical and visual comparison of the reconstructed boundary conditions with their exact solutions at predefined time points $t \in \{0.2, 0.4, 0.6, 0.8, 1\}$.
2. Mean Absolute Error (MAE) and metric S : The MAE for $q(t)$ and $p(t)$ is computed for each method to quantify deviations from the exact solutions. The metric S is defined as:

$$S = \left[\frac{1}{N-1} \sum_{i=1}^N (\hat{p}_i - p_i)^2 \right]^{\frac{1}{2}}. \quad (30)$$

Here, N represents the total number of estimated values, $\hat{p}_i$ denotes the values computed from the interpolation equation, and p_i refers to the exact values of $p(t)$.

3. Numerical solution $U(x, t)$: The solutions $U(x, t)$, computed using the reconstructed boundary conditions, are compared to validate each method's accuracy in solving the NIPDEs.

6.4 Objectives of the Comparison

The comparison aims to:

1. Highlight the impact of optimization: Examine whether optimizing L in the Rational Legendre PINNs improves its performance compared to the Standard PINNs and Legendre PINNs.
2. Benchmark against ITLBO: Evaluate the accuracy and computational efficiency of PINNs-based approaches, particularly the Rational Legendre PINNs, relative to the ITLBO algorithm.
3. Assess accuracy with noise-free data: Validate the ability of all methods to accurately reconstruct the unknown boundary conditions without noise interference.

For this case, consider the Fisher inverse problem (7) with reaction coefficient $a = 1$ (as in [21]):

$$U_t(x, t) = U_{xx}(x, t) + aU(x, t)(1 - U(x, t)), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \quad a = 1, \\ U(x, 0) = \frac{1}{(1 + e^{\sqrt{\frac{1}{6}}x})^2}, \quad 0 \leq x \leq 1, \quad (31)$$

$$U(0, t) = \frac{1}{(1 + e^{-\frac{5}{6}t})^2}, \quad 0 \leq t \leq 1, \\ U(1, t) = q(t), \quad 0 \leq t \leq 1, \quad (32)$$

so that $p(t)$ and $f(x)$ are prescribed and $q(t)$ is unknown. The over-specified condition is

$$U(0.4, t_i) = s(t_i), \quad t_i = i \times 0.2, \quad i = 1, 2, \dots, 5. \quad (33)$$

Where the unknown function is the continuous function $q(t)$ as:

$$q(t) = \frac{1}{(1 + e^{\sqrt{\frac{1}{6}} - \frac{5}{6}t})^2}, \quad 0 \leq t \leq 1. \quad (34)$$

And the exact solution of this problem is:

$$U(x, t) = \frac{1}{(1 + e^{\sqrt{\frac{1}{6}}x - \frac{5}{6}t})^2}, \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1. \quad (35)$$

Table 1. Comparison of methods for the Fisher equation: estimated $q(t)$ at selected times and MAE with noise-free data. CRLPINNs uses the optimal L obtained from COA.

Method	t=0.2	t=0.4	t=0.6	t=0.8	t=1.0	MAE($q(t)$)	S
ITLBO	0.2064	0.2488	0.2926	0.3365	0.3871	0.0177553	0.0091729
PINNs	0.1934848	0.2316660	0.2736249	0.3187183	0.3660759	0.0001990	8.857e-05
LPINNs	0.1936399	0.2318679	0.2736702	0.3183694	0.3650662	0.0002384	8.689e-05
CRLPINNs	0.1935695	0.2316893	0.2735250	0.3184275	0.3655615	6.987e-05	6.579e-05
EXACT	0.1935090	0.2316304	0.2734472	0.3183751	0.3656613		

In this example, the Burgers–Fisher inverse problem (11) is considered with $F(x, t) = U(x, t)(1 - U(x, t))$:

$$U_t(x, t) = U_{xx}(x, t) + U(x, t)U_x(x, t) + U(x, t)(1 - U(x, t)), \quad 0 \leq x \leq 1, \quad 0 \leq t \leq 1, \\ U(x, 0) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4}x\right), \quad 0 \leq x \leq 1, \quad (36)$$

$$U(0, t) = p(t), \quad 0 \leq t \leq 1, \quad (37)$$

$$U(1, t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4}\left(1 + \frac{5}{2}t\right)\right), \quad 0 \leq t \leq 1.$$

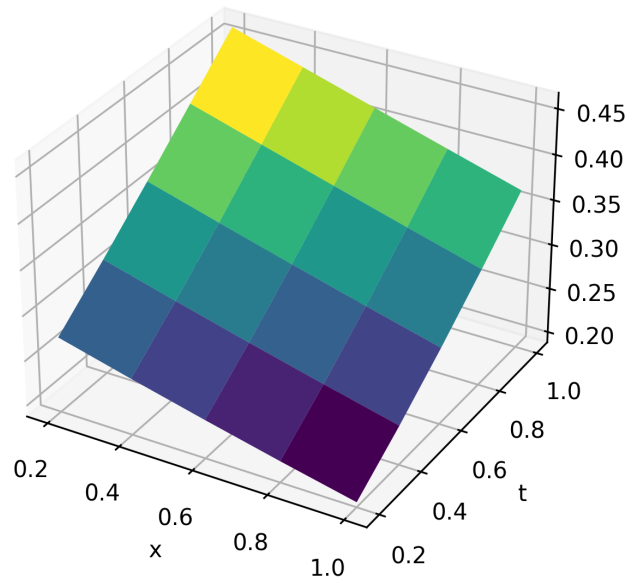
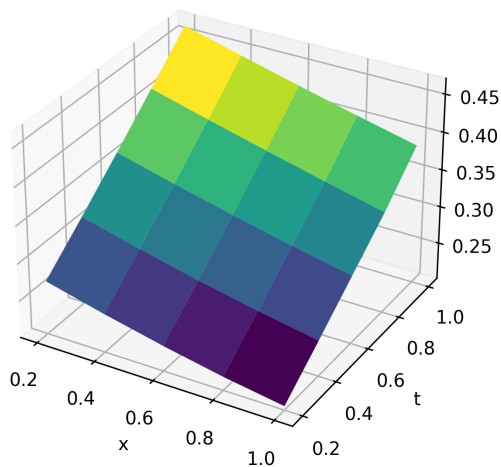
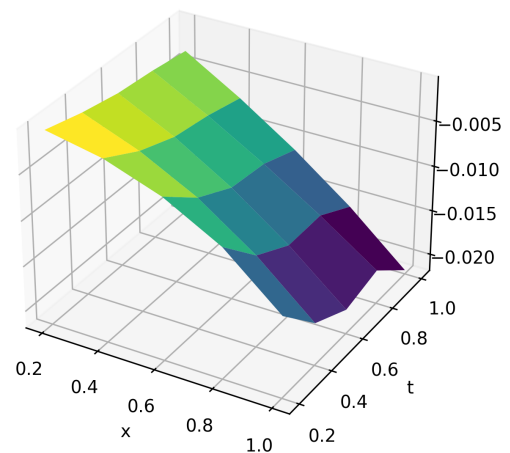


Figure 5. The exact answer of the NIPDEs problem $U(x, t)$



(a)



(b)

Figure 6. (a) The solution of the NIPDEs problem using the ITLBO method. (b) The difference between the exact solution and the ITLBO method.

Here $q(t)$ and $f(x)$ are known and $p(t)$ is unknown. The over-specified condition is

$$U(0.4, t_i) = s(t_i), \quad t_i = i \times 0.2, \quad i = 1, 2, \dots, 5. \quad (38)$$

where the unknown function is the continuous function $p(t)$ as:

$$p(t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4} \cdot \frac{5}{2} t\right), \quad 0 \leq t \leq 1. \quad (39)$$

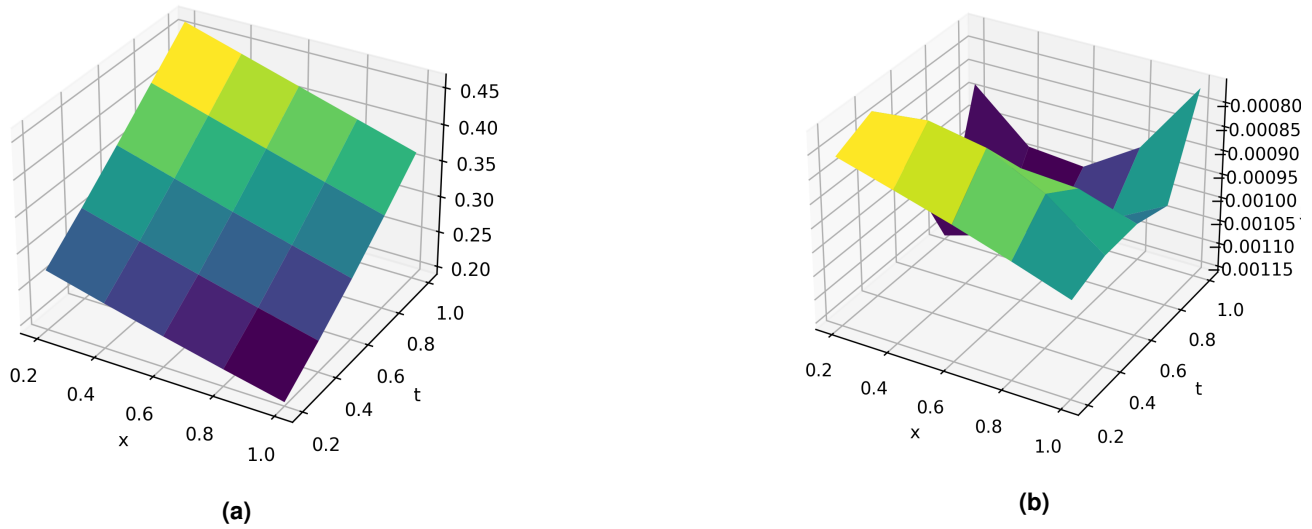


Figure 7. (a) The solution of the NIPDEs problem using the PINNs method. (b) The difference between the exact solution and the PINNs method.

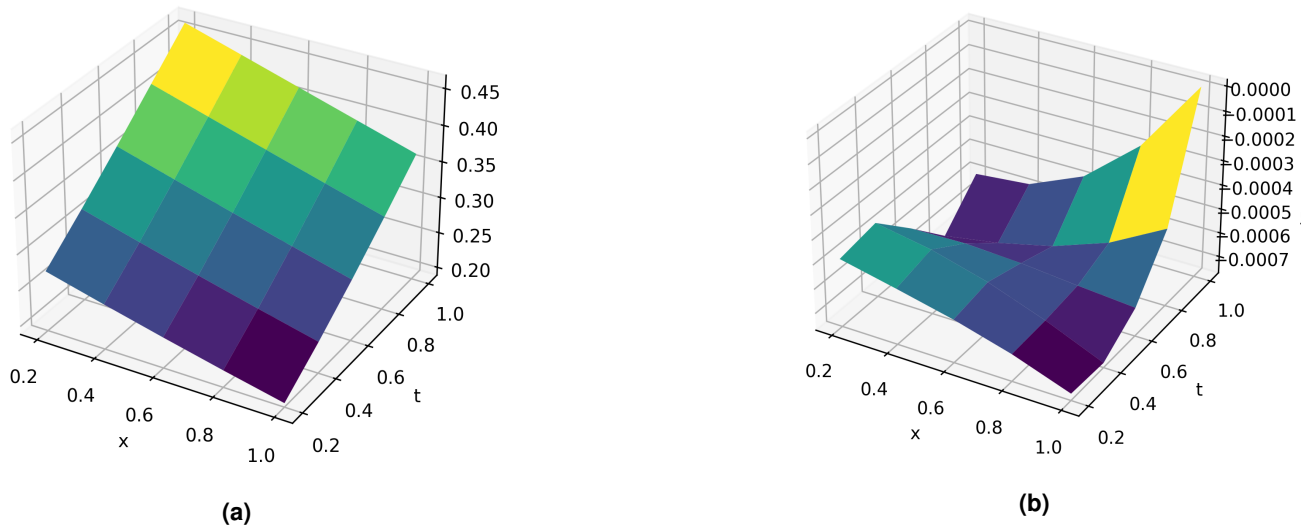


Figure 8. (a) The solution of the NIPDEs problem using the LPINNs method. (b) The difference between the exact solution and the LPINNs method.

And the exact solution of this problem is:

$$U(x,t) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4}\left(x + \frac{5}{2}t\right)\right), \quad 0 \leq x \leq 1, 0 \leq t \leq 1. \quad (40)$$

All numerical results in this section were obtained using the implementations described above (PyTorch in Jupyter Notebook). The optimal L values reported correspond to the COA runs used to generate the tables and figures; small variations in L may occur across different runs while preserving the reported accuracy.

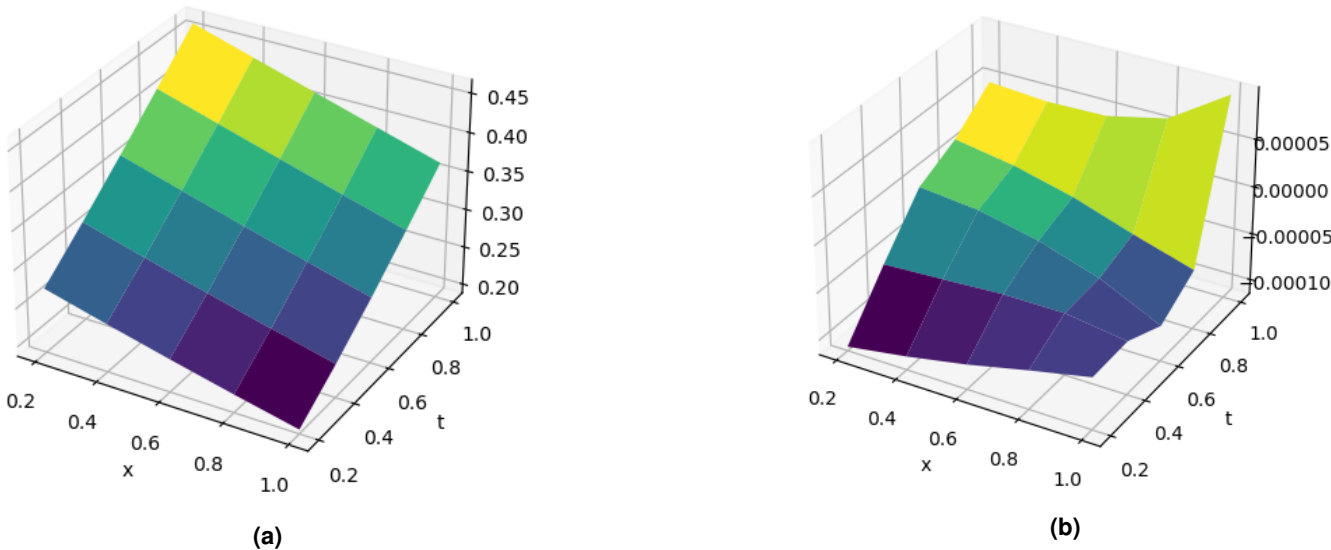


Figure 9. (a) The solution of the NIPDEs problem using the CRLPINNs method. (b) The difference between the exact solution and the CRLPINNs method.

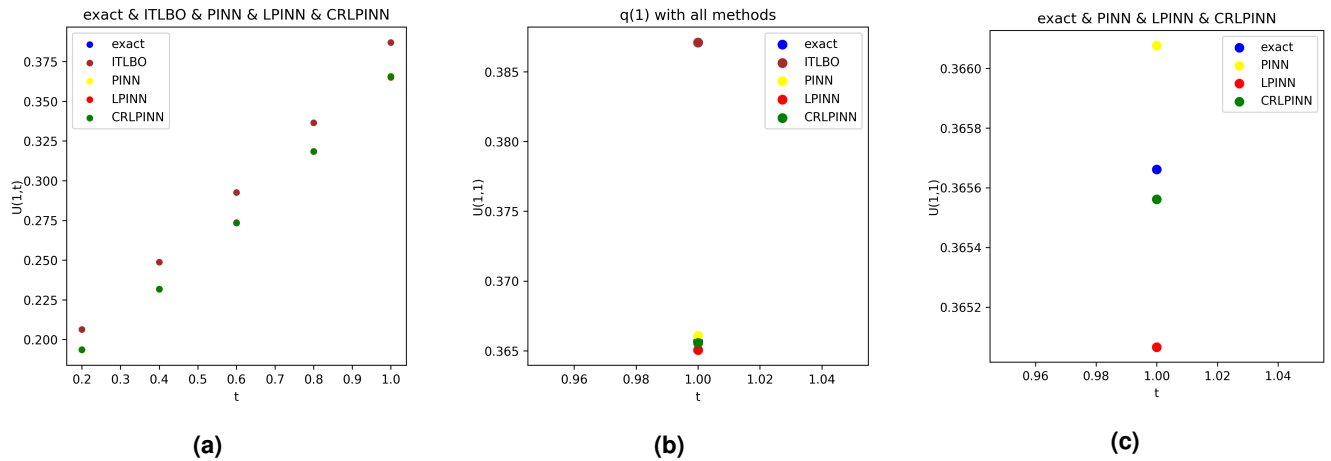


Figure 10. (a) q at all unknown points with all methods. (b) q at point $q(1)$ with all methods. (c) q at point $q(1)$ with all methods except ITLBO.

Table 2. Comparison of methods for the Burgers–Fisher equation: estimated $p(t)$ at selected times and MAE with noise-free data. CRLPINNs uses the optimal L obtained from COA.

Method	t=0.2	t=0.4	t=0.6	t=0.8	t=1.0	MAE($p(t)$)	S
ITLBO	0.5638	0.6210	0.6768	0.7299	0.7746	0.0018639	0.0070153
PINN _s	0.5634611	0.6239280	0.6806042	0.7322253	0.7779769	0.0012045	0.0011208
LPINN _s	0.5623750	0.6227077	0.6793375	0.7309771	0.7768133	0.0002347	9.300e-05
CRLPINN _s	0.5622668	0.6225638	0.6792283	0.7309972	0.7771002	0.0001010	3.351e-05
EXACT	0.5621765	0.6224593	0.6791787	0.7310585	0.7772998		

7 Conclusion

- An important observation in our study is the superior performance of Rational Legendre polynomials when applied to nonlinear problems with nontrivial spatial profiles and steep gradients, as illustrated by the Fisher equation example. In our implementation,

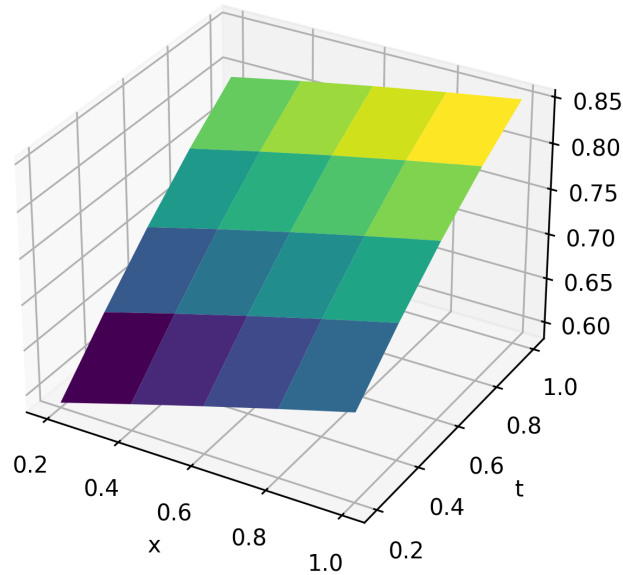


Figure 11. The exact answer of the NIPDEs problem $U(x,t)$

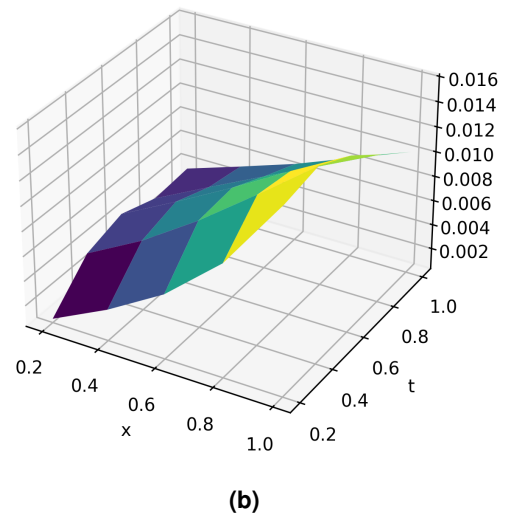
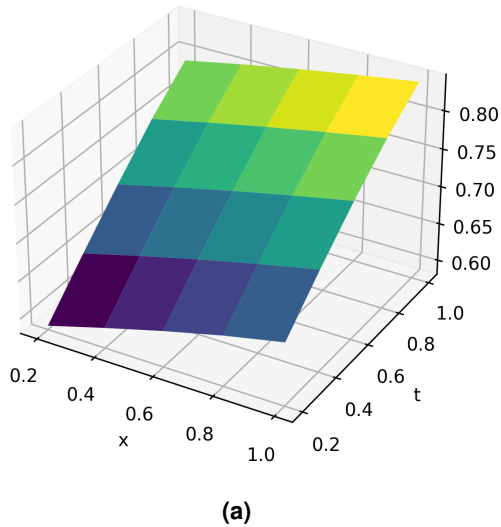


Figure 12. (a) The solution of the NIPDEs problem using the ITLBO method. (b) The difference between the exact solution and the ITLBO method.

the Rational Legendre polynomials effectively captured the key features of the solution, leading to improved numerical accuracy and reduced error. Compared to standard Legendre-based methods, our results showed that the Rational Legendre approach provided a better approximation with a lower mean absolute error. This improvement can be attributed to the fractional form of the Rational Legendre basis, which is well suited for representing complex solution behavior on the considered domains.

This finding highlights the importance of choosing basis functions that are well matched to the expected behavior of the solution. In problems where the solution exhibits sharp transitions or nontrivial asymptotic behavior—as is the case in many nonlinear and reaction–diffusion systems—Rational Legendre polynomials can offer a robust and accurate spectral approximation framework.

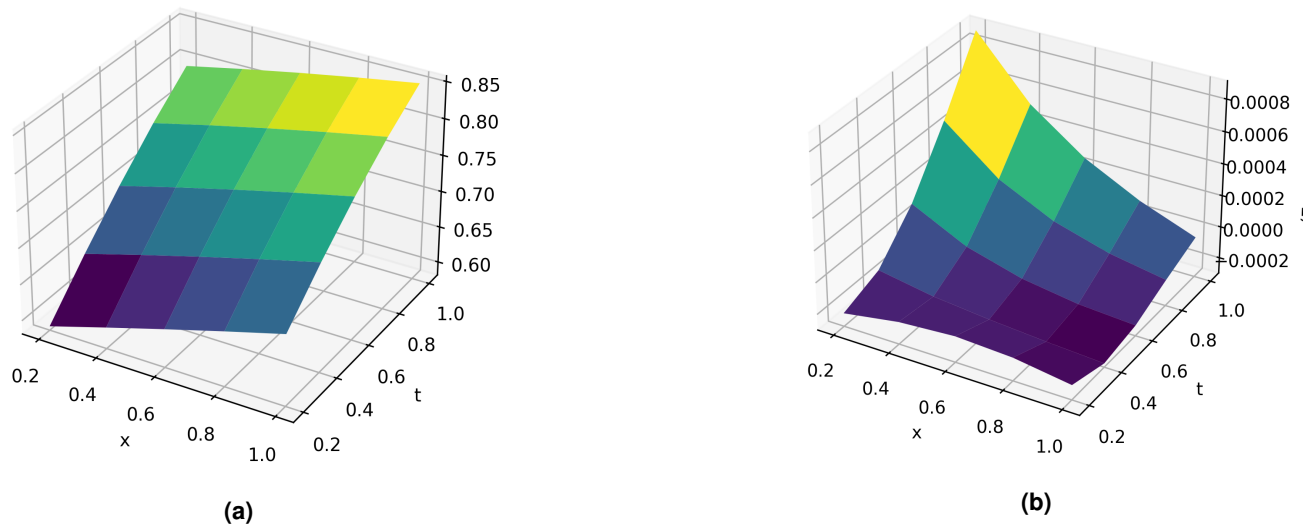


Figure 13. (a) The solution of the NIPDEs problem using the PINNs method. (b) The difference between the exact solution and the PINNs method.

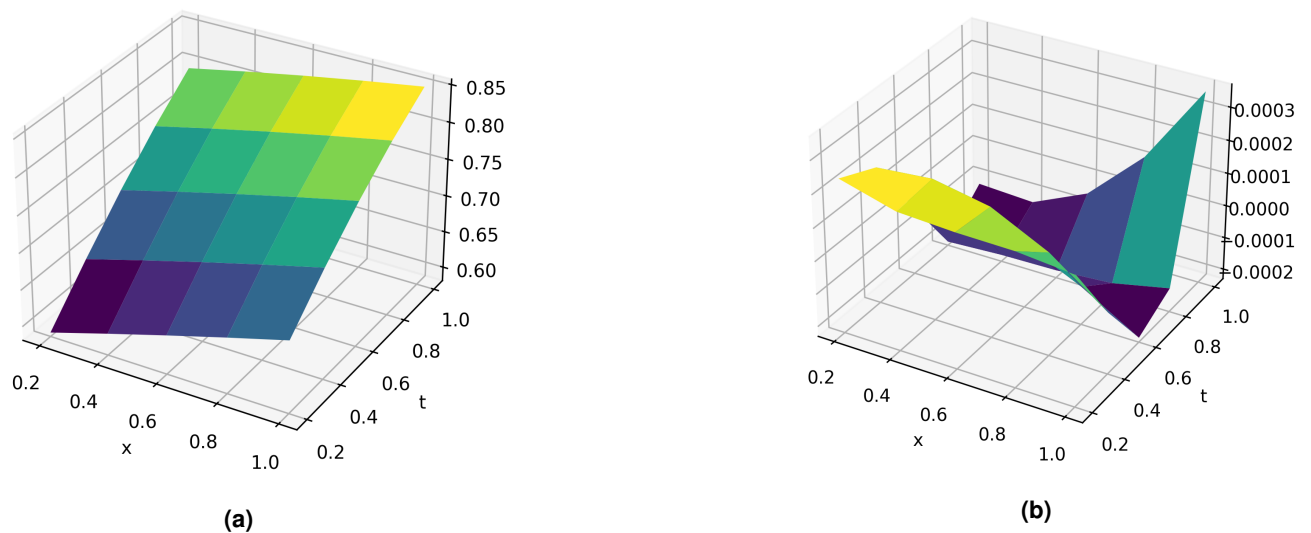


Figure 14. (a) The solution of the NIPDEs problem using the LPINNs method. (b) The difference between the exact solution and the LPINNs method.

- The optimized Rational Legendre PINNs achieved the strongest overall accuracy among the PINN variants considered, owing to the rational mapping and tuning of L via COA.
- Comparisons of $q(t)$, $p(t)$, and MAE across all methods provide quantitative insight into their relative performance.
- The absence of noise ensures that the results focus solely on the inherent strengths and limitations of each method.

This comprehensive comparison validates the effectiveness of the proposed Rational Legendre PINNs approach and its advantages over existing methods, including the ITLBO algorithm, in solving complex NIPDEs.

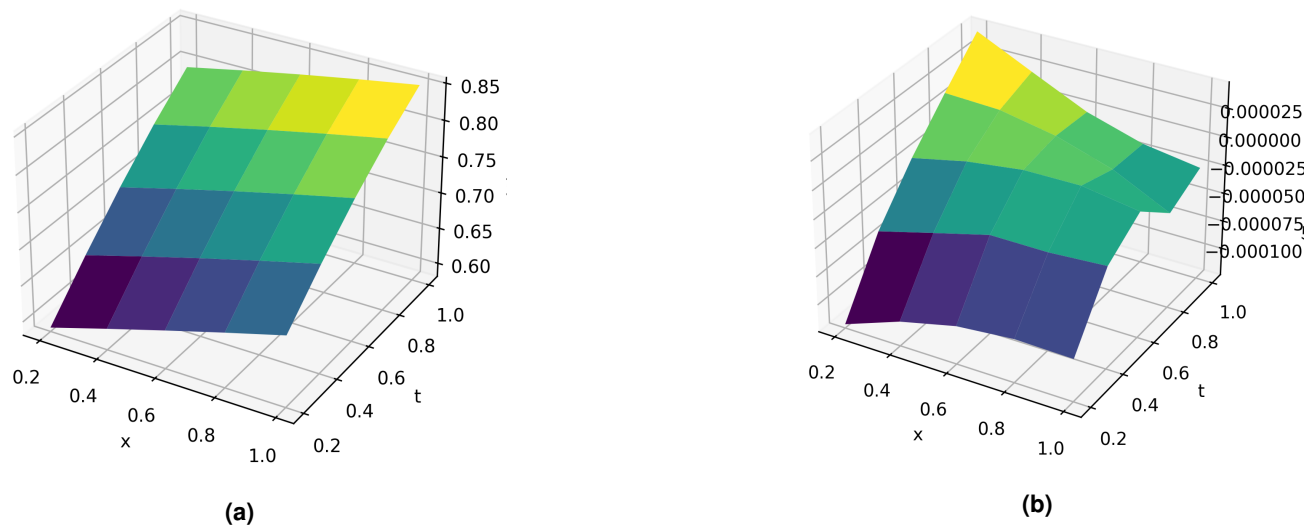


Figure 15. (a) The solution of the NIPDEs problem using the CRLPINNs method (optimal $L = 7.41$ from COA). (b) The difference between the exact solution and the CRLPINNs method.

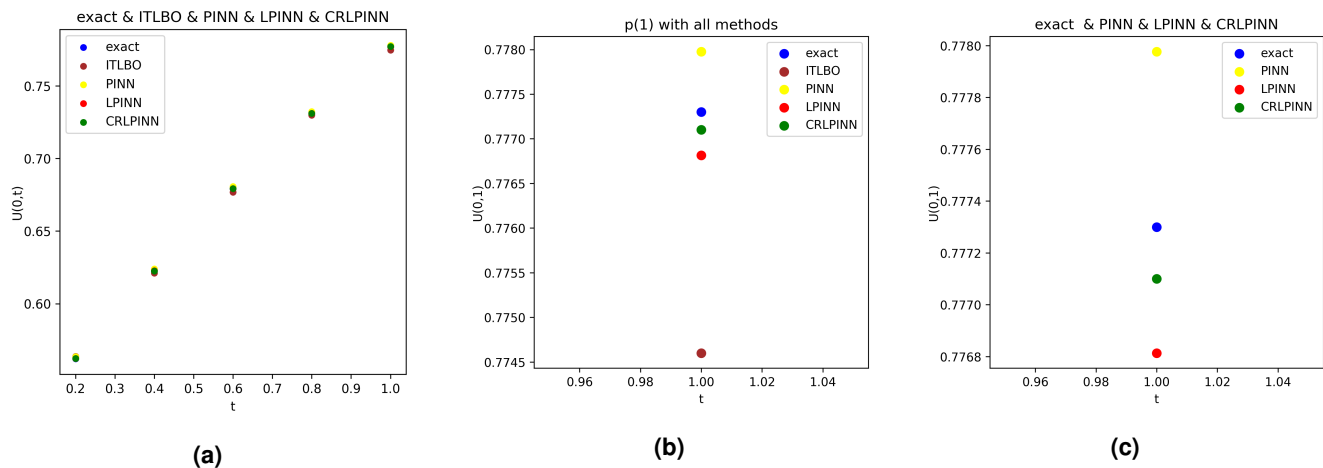


Figure 16. (a) p at all unknown points with all methods. (b) p at point $p(1)$ with all methods. (c) p at point $p(1)$ with all methods except ITLBO.

Authors' Contributions

All authors contributed to the conceptualization, implementation, analysis, and writing of this study.

Data Availability

The data and results generated in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Ethical Considerations

This study does not involve human or animal subjects, and therefore no ethics approval is required.

Funding

This research received no external funding.

Acknowledgments

The authors would like to thank their colleagues for helpful discussions.

References

- [1] H. D. Mazraeh and K. Parand, An innovative combination of deep q-networks and context-free grammars for symbolic solutions to differential equations, *Engineering Applications of Artificial Intelligence*, 142, 109733, (2025).
- [2] H. D. Mazraeh, K. Parand, M. Hosseinzadeh, J. Lansky, and V. Nulícek, An improved water strider algorithm for solving the inverse burgers huxley equation, *Scientific Reports*, (2024).
- [3] A. H. Hasanoglu and V. G. Romanov, *Introduction to Inverse Problems for Differential Equations*. Springer Cham, 2017.
- [4] R. Pourgholi, H. Dana, and S. H. Tabasi, Solving an inverse heat conduction problem using genetic algorithm: Sequential and multi-core parallelization approach, *Applied Mathematical Modelling*, 38(7), 1948–1958, (2014).
- [5] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, vol. 375. Springer Science & Business Media, 1996.
- [6] Ü. Lepik, Solving pdes with the aid of two-dimensional haar wavelets, *Computers & Mathematics with Applications*, 61(7), 1873–1879, (2011).
- [7] S. Dong and Y. Wang, A method for computing inverse parametric pde problems with random-weight neural networks, *Journal of Computational Physics*, 489, 112263, (2023).
- [8] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, SIAM, 2007.
- [9] S. C. Brenner, *The mathematical theory of finite element methods*. Springer, 2008.
- [10] 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, 686–707, (2019).
- [11] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, *Nature Reviews Physics*, 3(6), 422–440, (2021).
- [12] Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang, Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde), *Physical Review Research*, 4(2), 023174, (2022).
- [13] T. T. N. Nguyen, Bi-level iterative regularization for inverse problems in nonlinear pdes, *Inverse Problems*, 40(4), 045020, (2024).
- [14] H. Dana Mazraeh, K. Parand, H. Farahani, and S. R. Kheradpisheh, An improved imperialist competitive algorithm for solving an inverse form of the huxley equation, *Iranian Journal of Numerical Analysis and Optimization*, 14(3), 681–707, (2024).
- [15] R. Rajabioun, Cuckoo optimization algorithm, *Applied Soft Computing*, 11(8), 5508–5518, (2011).

- [16] X.-S. Yang, Cuckoo search for inverse problems and simulated-driven shape optimization, *Journal of Computational Methods in Science and Engineering*, 12(1-2), 129–137, (2012).
- [17] S. Pakravan, P. A. Mistani, M. A. Aragon-Calvo, and F. Gibou, Solving inverse-pde problems with physics-aware neural networks, 440, 110414, (2021).
- [18] J. Ross, A. F. Villaverde, J. R. Banga, S. Vázquez, and F. Morán, A generalized fisher equation and its utility in chemical kinetics, *Proceedings of the National Academy of Sciences*, 107(29), 12777–12781, (2010).
- [19] J. J. Bramburger and C. Henderson, The speed of traveling waves in a fkpp-burgers system, *Archive for Rational Mechanics and Analysis*, 241(2), 643–681, (2021).
- [20] A.-M. Wazwaz, The tanh method for generalized forms of nonlinear heat conduction and burgers–fisher equations, *Applied Mathematics and Computation*, 169(1), 321–338, (2005).
- [21] A. Aliyari Boroujeni, R. Pourgholi, and S. H. Tabasi, A new improved teaching–learning-based optimization (itlbo) algorithm for solving nonlinear inverse partial differential equation problems, *Computational and Applied Mathematics*, 42(2), 99, (2023).
- [22] N. A. Kudryashov, Exact solutions of a family of fisher equations, *Theoretical and Mathematical Physics*, 94(2), 211–218, (1993).
- [23] X. Wang and Y. Lu, Exact solutions of the extended burgers-fisher equation, *Chin. Phys. Lett.*, 7(4), 145–147, (1990).
- [24] M. El-Hachem, S. W. McCue, W. Jin, Y. Du, and M. J. Simpson, Revisiting the fisher–kolmogorov–petrovsky–piskunov equation to interpret the spreading–extinction dichotomy, *Proceedings of the Royal Society A*, 475(2229), 20190378, (2019).
- [25] A. P. Márquez, R. de la Rosa, T. M. Garrido, and M. L. Gandarias, Conservation laws and exact solutions for time-delayed burgers–fisher equations, *Mathematics*, 11(17), 3640, (2023).
- [26] T. Tang, L.-L. Wang, H. Yuan, and T. Zhou, Rational spectral methods for pdes involving fractional laplacian in unbounded domains, *SIAM Journal on Scientific Computing*, 42(2), A585–A611, (2020).
- [27] C. Sheng, D. Cao, and J. Shen, Efficient spectral methods for pdes with spectral fractional laplacian, *Journal of Scientific Computing*, 88(1), 4, (2021).
- [28] J. P. Boyd, *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.
- [29] G. G. Tejani, N. Mashru, P. Patel, S. K. Sharma, and E. Celik, Application of the 2-archive multi-objective cuckoo search algorithm for structure optimization, *Scientific Reports*, 14(1), 31553, (2024).
- [30] X.-S. Yang and S. Deb, Engineering optimisation by cuckoo search, *International Journal of Mathematical Modelling and Numerical Optimisation*, 1(4), 330–343, (2010).
- [31] E. Shadkam and M. Bijari, Evaluation the efficiency of cuckoo optimization algorithm, *arXiv preprint arXiv:1405.2168*, (2014).
- [32] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, 29, 17–35, (2013).