Global Integrated Mathematics

https://gim.gospub.com/gim

Global Open Share Publishing

Article

Causal Physics-Informed Neural Networks for Singular and Singularly Perturbed Boundary Value Problems in Chemical Systems

Muhammad Israr, Ilyas Khan*

Abdus Salam School of Mathematical Sciences, Government College University, Lahore, Pakistan

*Corresponding author: Ilyas Khan, ilyas.khan_22@sms.edu.pk

Abstract

Singular boundary value problems (SBVPs) and singularly perturbed boundary value problems (SPBVPs) are commonly arise in the modeling of critical chemical processes, including isothermal gas spheres, electroactive polymer films, thermal explosions, and chemical reactor theory. Due to their extensive applications, these problems require the creation of effective numerical methods for precise solutions. Traditional methods often struggle with stiffness and multiscale behavior, while analytical solutions are limited to specific cases. In this work, we use a data-driven approach using Physics-Informed Neural Networks (PINNs) to solve SBVP and SPBVPs. By initializing the perturbation parameter as a trainable variable, it is optimized alongside net-work weights and biases during training. Numerical simulations demonstrate that causality-enhanced PINNs significantly improve the solution accuracy for singular chemical boundary value problems.

Keywords

Scientific machine learning, Causality, PINNs, Chemical differential equations, SBVPs and SPBVPs

AMS Classification

94A08, 65N06, 68U10, 65N12.

Article history

Received: 14 May 2025 Accepted: 18 June 2025

Revised: 09 June 2025 Available Online: 24 June 2025

Copyright:

© 2025 by the authors. This article is published by the Global Open Share Publishing Pty Ltd under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/

1. Introduction

Singular boundary value problems (SBVPs) are encountered across numerous scientific fields, including chemistry, physics, and biology, with applications in areas such as astrophysics [1], nuclear physics [2], modeling tumor growth [3], electrodynamics [4], chemical reactor theory [5], and physiological studies [6]. The analytical solutions for SBVPs are confined to certain types of equations, which has led to the creation of various numerical techniques to overcome these obstacles. Among these techniques, the finite difference method is notably a popular direct approach that has been effectively utilized in numerous scientific applications [7,8]. Another direct method, known as the collocation approach, involves placing local or global approximate solutions into the differential equation and reducing the residual at specific collocation points [9,10]. Other techniques include the series expansion method (SEM) [11], the Sinc-Galerkin method (SGM) [12], and methods based on splines [13,14]. Nonetheless, many of these methods depend on high-order formulations, which may lead to extensive computational requirements for linearization or discretization processes.

The variational iteration method (VIM) offers an alternative strategy when addressing SBVPs [15,16]. Although it has significant potential, VIM faces challenges in determining the Lagrange multiplier and may not be ideal for certain nonlinear issues. In a comparable manner, the Adomian decomposition method (ADM) [17] requires employing the Volterra integral representation of the differential equation to adequately tackle singularities [18,19]. To overcome this limitation, the modified Adomian method [5] incorporates Green functions but at the cost of computational inefficiency. The advanced Adomian method [20] has been developed to address these drawbacks and enhance computational performance.

The study of singular perturbation boundary value problems (SPBVPs) is frequently encountered in various fields of applied sciences and engineering, including chemical reactor theory, combustion, fluid dynamics, elasticity, bio-science, nuclear engineering, and control theory [21-25]. SPBVPs arise within the context of adiabatic tubular chemical flow reactors that incorporate axial diffusion in chemistry. The difficulties associated with these problems stem from their multiscale characteristics [26,27], which are defined by a small perturbation parameter ϵ that acts as the coefficient for the highest derivative. This multiscale nature frequently leads to thin transition layers where solutions undergo rapid changes, while the remainder of the interval shows gradual and consistent variation. This structure makes SPBVPs difficult to solve with traditional numerical methods, particularly for very small ϵ values. Robust numerical techniques are essential for addressing these challenges.

Current approaches for SPBVPs involve converting the problems into initial value problems (IVPs), referred to as initial value methods [26,28-33]. Direct techniques, including the finite difference method [34,35] and spline methods [36-38], are also commonly employed. Spline techniques provide advantages over finite difference approaches by delivering continuous approximate solutions. In addition, advanced methods such as the wavelet-optimized difference technique [39] and the uniform Haar wavelet approach [40] have demonstrated significant efficiency. The Haar wavelet method transforms differential equations into systems of algebraic equations using sparse matrices, achieving faster computation. Analytic methods, such as those based on matched asymptotic expansions [41], complement numerical approaches by analyzing function behavior.

Recently, deep learning techniques have gained traction for solving partial differential equations (PDEs) due to their mesh-free nature, efficiency in handling inverse problems, and capability in high-dimensional spaces [42,43]. Prominent methods in scientific machine learning include the deep Ritz method [44], the deep Galerkin method [45], and physics-informed neural networks (PINNs) [46]. These approaches leverage deep neural networks (DNNs) to minimize loss functions that incorporate domain-specific information and physical laws. In PINNs, the loss function integrates the PDE residual, boundary and initial conditions, and observational data, making them versatile for approximating PDE solutions.

The optimization of DNN parameters during PINN training typically employs algorithms like Adam [47], L-BFGS [48], and stochastic gradient descent (SGD) [49]. However, these methods often underperform in stiff or ill-conditioned problems [50]. To address these challenges, researchers have developed adaptive weighting strategies [51], multiscale architectures [52,53], and ensemble learning techniques [54], including gradient boosting [55]. Adaptive activation functions [56,57] and novel optimizers, such as Gauss-Newton methods [58,59], block coordinate methods [60], and variable projection techniques [61-63], further enhance PINN performance, creating a bridge between traditional numerical methods and modern machine learning techniques.

A loss function for PINNs that respects causality effectively tackles these limitations and imposes the necessary physical principles. However, applying this adjusted causal loss function necessitates a more complex neural network with a greater number of parameters. Given the significance of different elements in engineering design and the challenges of simulating every moment, transfer learning is introduced within the causal PINN framework. By integrating transfer learning, the parameters from an already trained model are utilized to initialize and train new models. As a result, this approach lessens the computational load and speeds up convergence for future tasks, thereby enhancing the efficiency of complex system simulations.

Although there are several numerical techniques for addressing singular and singularly perturbed boundary value problems (SBVPs and SPBVPs), numerous methods encounter considerable difficulties, especially when handling stiff

or multi scale systems. Traditional methods, such as finite difference, collocation, and spline techniques, often require high-order formulations or computationally expensive discretization, limiting their efficiency and scalability. Moreover, analytical solutions are only feasible for specific types of equations, leaving a wide range of complex problems unresolved. Recent advancements in deep learning, particularly PINNs, offer a promising alternative by leveraging the mesh-free nature of neural networks and their ability to handle high-dimensional and inverse problems. However, existing PINN frameworks often struggle with stiff or ill-conditioned problems, leading to suboptimal performance. This work aims to address these limitations by introducing a causality-driven PINN framework that optimizes the perturbation parameter alongside network weights and biases, thereby enhancing solution accuracy and computational efficiency for challenging chemical boundary value problems. While causal Physics-Informed Neural Networks (cPINNs) can be applied to linear differential equations, this study focuses on SBVPs and SPBVPs due to their complexity and the challenges they pose for traditional numerical methods. These problems highlight the strength of PINNs in handling singularities, stiffness, and multiscale behavior where conventional approaches often struggle.

PINNs are particularly suitable for SBVPs and SPBVPs because of their ability to approximate complex solution behaviors without requiring mesh discretization. Traditional numerical methods often struggle with singularities or steep boundary layers induced by small perturbation parameters, requiring fine meshes or asymptotic techniques. In contrast, PINNs naturally handle such features by incorporating the governing differential equations and boundary conditions directly into the loss function, allowing the neural network to learn accurate solutions even in the presence of singularities or sharp gradients.

Causal training offers a promising enhancement to PINNs by directing the learning process from simple to more complex regions of the solution domain. This mitigates error accumulation and improves convergence, especially in problems where standard PINNs might fail due to imbalance in learning dynamics caused by singularities. Our work differs from existing literature in multiple ways. While recent works, such as [64], use causal PINNs for transfer learning in beam simulations to improve generalizability and convergence, our study focuses on using causal PINNs specifically for SBVPs and SPBVPs. Notably, we treat the perturbation parameterϵ as a trainable variable within the network, enabling its optimization alongside the model weights. Additionally, we integrate causal training into the PINN framework and validate its effectiveness on four distinct SBVPs and SPBVPs. For three of these problems, we compare our results with known exact solutions, while for the fourth problem, we compare with the numerical solution obtained using the Finite Element Method (FEM). The comparisons demonstrate significant improvements in solution accuracy and training stability over standard PINNs, highlighting the robustness and flexibility of our proposed approach.

The remainder of the paper is organized as follows: Section 2 discusses the architectures of PINNs. Section 3 outlines the framework for applying transfer learning with causal PINNs to model training. Section 4 details the results of numerical experiments, demonstrating the effectiveness of our approach in tackling challenging chemical boundary value problems (BVPs), where traditional numerical methods often fall short. Finally, Section 5 provides a summary of the key findings and conclusions derived from this research.

2. Physics-Informed Neural Networks

The problem is as follows: We consider differential equations of a generic type. Make sure that $u: \Omega \to \mathbb{R}$

$$\begin{aligned} \mathcal{A}(\mathrm{u}(\mathrm{x})) &= \mathrm{g}(\mathrm{x}) && \text{for } \mathrm{x} \in \Omega \\ \mathcal{P}^{\mathrm{k}}(\mathrm{u}(\mathrm{x})) &= \mathrm{h}^{\mathrm{k}}(\mathrm{x}) && \text{for } \mathrm{x} \in \Gamma^{\mathrm{k}} \subseteq \partial\Omega, \quad \mathrm{k} = 1,2,\ldots,\mathrm{n}_{\Omega} \end{aligned} \tag{1}$$

A nonlinear differential operator is represented by the symbol $\mathcal{A}$, while a set of boundary condition operators is indicated by the symbol $\mathcal{P}_{k=1}^{n_\Gamma}$. We treat time as an additional component of the vector $x \in \mathbb{R}^d$, where d is the dimension of the problem, while working with time-independent problems. Given that the computational domain Ω encompasses both spatial and temporal dimensions, the initial condition can be interpreted as a particular kind of boundary condition within the spatiotemporal domain.

2.1 DNN Approximation of the Solution

In accordance with [46], we want to use PINNs to approximate a solution of (1), i.e., $u(x) \approx u_{\mathrm{NN}}(\theta, x)$, where the trainable parameters $\theta \in \mathbb{R}$ parametrize the nonlinear mapping $u_{\mathrm{NN}}: \mathbb{R}^d \to \mathbb{R}$. The network architecture usually induces the exact shape of u_{NN}. In this scenario, we analyze a network that consists of an input layer, $l-1$ hidden layers, and an output layer, structured as follows:

$$\begin{aligned} \text{Input layer: } & y_i = W_i x, \\ \text{Hidden layers: } & y_h = \mathbb{N}(y_{h-1}), \text{for } h = 1,2,3,\ldots,l-1, \\ \text{Output layer: } & y_o = W_o y_{l-1} + b_l \end{aligned} \tag{2}$$

The input layer maps input features $\mathrm{x} \in \Omega \subset \mathbb{R}$ into the dimension of the hidden layers using weights $\mathrm{W_i} \in \mathbb{R}^{\mathrm{d}\times\mathrm{nh}}$. The h^{th} hidden layer uses a nonlinear operator $\mathbb{N}: \mathbb{R}^{\mathrm{nh}} \to \mathbb{R}^{\mathrm{nh}}$ to non-linearly transform the $(\mathrm{h}-1)^{\mathrm{th}}$ layer's output $\mathrm{y_{h-1}}$ into

y_o. There are various ways to represent the operator $\mathbb{N}$. We make use of a skip-connected single-layer perceptron, which is as follows:

$$\mathbb{N}(y_{h-1}) = y_{h-1} + \sigma_h(W_h y_{h-1} + b_h) \quad (3)$$

The parameters related to the layer h can collectively be expressed as: $[\theta_h = (\text{flat}\ (W_h), \text{flat}\ (b_h))]$ where the weights and biases are indicated by $[W_h \in \mathbb{R}^{n_h \times n_h}, b_h \in \mathbb{R}^{n_h}]$ respectively. The function (flat(.)) represents a flattening operator, and $[\sigma_h: \mathbb{R}^{n_h} \to \mathbb{R}^{n_h}]$ denotes an activation function. Finally, the output of the last hidden layer is linearly transformed using W_l and b_l to construct the output of the network y_l. Figure 1 illustrates the architecture of a cPINN. The process begins with the input x, passed through a feed-forward neural network NN(w, b) parameterized by weights w and biases b. The network outputs an approximation û of the target solution u.

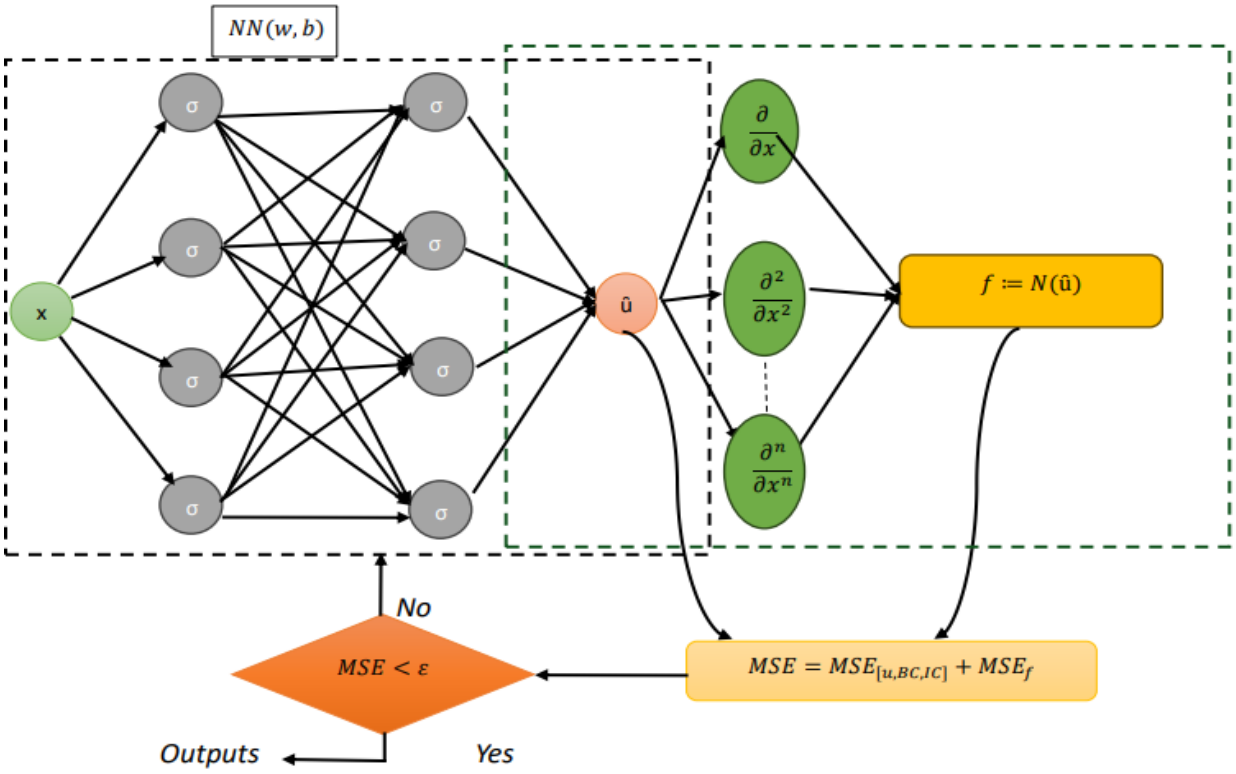

Figure 1. Methodology of the proposed cPINN, illustrating the network architecture, PDE residual computation at collocation points, and integration of causal training.

To enforce the governing physical laws, û is further processed to compute its derivatives: $\frac{\partial u}{\partial x}, \frac{\partial^2 b}{\partial x^2}, \ldots, \frac{\partial^n \hat{u}}{\partial x^n}$, as required by the PDE. These derivatives are substituted into the differential operator f: $= \mathcal{N}(\hat{u})$, representing the PDE residual at selected collocation points.

$MSE_{0,BC,IC}$: Mean squared error on the predicted solution û at boundary and initial condition points.

MSE_f : Mean squared error of the PDE residual evaluated at interior collocation points. In the cPINNs framework, this term incorporates the adaptive causal weights that guide the training by emphasizing regions with higher residual errors.

Training proceeds by minimizing the total MSE until the condition MSE $< \epsilon$ is satisfied, indicating convergence. If not, the weights are updated, and the process continues iteratively.

This architecture differs from standard PINNs by employing causal training, in which the model learns progressively from simpler to more complex regions of the domain. This strategy helps mitigate error accumulation and accelerates convergence. Moreover, the structure supports transfer learning, enabling improved generalization to new but related problems. These innovations are particularly emphasized in the work of Kapoor et al [64]. for beam simulation problems.

2.2 Training Data and Loss Functional

To train the network to approach a solution of (1), we create a dataset $\mathcal{D}_{\text{int}} = \{x_j\}_{j=1}^{n_{\text{int}}}$ comprising n_{int} collocation points, which are in the interior of the computational domain Ω. The physics that the PDE captures is enforced by these points. To enforce the boundary conditions on Γ, we also take into consideration the dataset $\mathcal{D}_{bc} = \{(x_j^k, g_j^k)\}_{j=1}^{n_{bc}^k}$, where each

$\mathcal{D}_{bc}^{k}$ contains n_{bc}^{k} points. Using the values of $\mathcal{D}_{\text{int}}$ and $\mathcal{D}_{bc}^{k}$ for $k = 1, ..., n_{\Gamma}$, the following minimization problem is solved to obtain the network's ideal parameters:

$$\theta^{*} = \arg\min_{\theta \in \mathbb{R}^{n}} (\mathcal{L}_{\text{int}}(\theta) + \mathcal{L}_{bc}(\theta)) \quad \text{where } \mathcal{L}_{\text{int}}(\theta) \underset{\text{interior loss}}{\overset{\text{boundary}}{\rightarrow}} \text{ and } \mathcal{L}_{bc}(\theta) \text{ loss} \tag{4}$$

Here, the interior and the boundary losses are given as:

$$\begin{aligned} \mathcal{L}_{\text{int}}(\theta) &= \sum_{x_j \in \mathrm{D}_{\text{int}}} \left\| \mathcal{P}\left(u_{NN}(\theta, x_j)\right) - f(x_j) \right\|^{2}, \\ \mathcal{L}_{bc}(\theta) &= \sum_{(x_j^k, g_j^k) \in \mathcal{D}_{bc}^{k}} \| \left(u_{NN}(\theta, x_j^k) - g^k(x_j^k) \right\|^{2} \end{aligned} \tag{5}$$

Therefore, the expression for the interior loss $\mathcal{L}_{\text{int}}$ is the mean PDE residual obtained by Monte Carlo integration, that is, by averaging across the residuals evaluated at collocation points sampled in the interior of the domain. Remember that evaluating the PDE residual typically requires a comprehension of the partial derivatives of the network output $u_{NN}(\theta, x_j)$ with respect to the input x_j. With the use of sophisticated automatic differentiation techniques, these derivatives can be efficiently generated. Boundary loss $\mathcal{L}_{bc}$, the second term, guarantees that the boundary requirements are met for a collection of points sampled along boundaries $\Gamma^{k^{n_{\Gamma}}}{}_{k=1}^{n^2}$.

We modify the PINNs loss functions [64]. The PDE term $\mathcal{L}_{\text{int}}(\theta)$ is affected by the loss function alteration, but the boundary loss $\mathcal{L}_{bc}(\theta)$ terms are unaffected. The definition of the causal PDE loss phrase is:

$$\begin{aligned} \mathcal{L}_{\text{int}}(\theta) &= \sum_{x_j \in \mathbb{D}_{\text{int}}} w_i \mathcal{L}_{\text{int}}(t_i, \theta) \\ w_1 = 1, w_i &= e^{-\epsilon \sum_{x_j \in \mathbb{D}_{\text{int}}} \mathcal{L}_{\text{int}}(t_k, \theta)}, i \in \mathbb{D}_{\text{int}}. \end{aligned} \tag{6}$$

The computing domain has been divided into $\mathbb{D}_{\text{int}}$ timesteps. The sharpness of the weights is influenced by the causality hyperparameter ϵ. For the loss at each time step t_i, an adjustment incorporates a weighting factor, w_i. The weight w_i is determined by the total PDE loss up to time t_i. By taking the exponential of the negative cumulative loss, the weights are adjusted to prioritize the well-resolved solutions at earlier time levels. In conclusion, the causal PINN's adjusted loss function $\mathcal{L}_{\text{int}}(\theta)$ can be represented as:

$$\mathcal{L}_{\text{int}}(\theta) = \frac{1}{\mathbb{D}_{\text{int}}} \left[\sum_{x_j \in \mathrm{D}_{\text{int}}} w_i e^{-\epsilon \sum_{x_j \in \mathrm{D}_{\text{int}}} \mathcal{L}_{\text{int}}(\theta)} \mathcal{L}_{\text{int}}(t_i, \theta) \right] \tag{7}$$

The weights $w_1, ..., w_N$ must be big in order for $\mathcal{L}_{\text{int}}(\theta)$ to be minimized, according to the modified loss function. Starting at lower time levels and working its way up to higher ones, this architecture compels the neural network to minimize the loss in a sequential manner. To put it simply, the causal PINN gives lower time step fully resolved solutions priority over higher time step approximations.

After training, the optimal parameters θ can be used to infer an approximate solution u_{NN} of (1). The quality of this solution can be assessed by means of the error $\epsilon(u_{NN}, u^{*}) = \|u_{NN} - u^{*}\|$, where u^{*} denotes an exact solution. In general, the error ϵ consists of three components: the discretization error, network approximation error and optimization error [65]. More precisely, the quantity and position of the collocation points dictate the discretization error. The DNN's representation capacity is linked to the networks' approximation error, meaning that it is directly influenced by the network architecture (number of layers, network width). The optimization error is a measure of the optimizer's solution quality. Our goal in this effort is to develop better training techniques that reduce optimization mistake.

2.3 The L-BFGS Optimizer

The Limited memory Broyden Fletcher Goldfarb Shanno (L-BFGS) algorithm is a quasi-Newton optimization method that approximates the inverse Hessian matrix efficiently by storing only a limited number of vectors from recent iterations. This makes L-BFGS highly suitable for large-scale optimization problems where maintaining the full Hessian is computationally infeasible. The update rule for the parameter vector θ_k at iteration k is given by:

$$\theta_{k+1} = \theta_k - \alpha_k H_k \nabla f(\theta_k) \tag{8}$$

where H_k is the approximate inverse Hessian matrix, $\nabla f(\theta_k)$ is the gradient of the objective function at iteration k, and α_k is the step size typically determined by a line search method.

The approximation matrix H_k is updated using the history of the last m parameter differences $s_k = \theta_{k+1} - \theta_k$ and gradient differences $y_k = \nabla f(\theta_{k+1}) - \nabla f(\theta_k)$, which reduces memory requirements and computational cost significantly compared to full BFGS [66,67].

2.4 Advantages over Adam

While adaptive gradient methods like Adam are widely used due to their simplicity and good performance on noisy or sparse gradients, L-BFGS often exhibits faster and more stable convergence on smooth, deterministic loss landscapes typical in scientific computing and PINNs. Unlike Adam, which uses a fixed exponential decay to estimate first and second moments of gradients, L-BFGS leverages curvature information to make more informed update steps that approximate Newton's method. This often results in fewer iterations and better final accuracy, especially for problems where the Hessian matrix structure is informative and the gradient noise is low. However, L-BFGS requires more computation per iteration and a line search, which might not be ideal for large-scale or highly stochastic problems where Adam excels.

3. Numerical Experiments

This section outlines the numerical examples that were used to evaluate the PINNs technique. We also examine the L-BFGS method's numerical performance. We track the loss functional $\mathcal{L}$ and the quality of PINNs solution (u_{NN}) during our inquiry. The latter is evaluated using the L^2 relative error, which is provided as:

$$\mathcal{E}_{rel}(u_{NN}, u^*) = \frac{\|u_{NN} - u^*\|_2}{\|u_{NN}\|_2} \quad (9)$$

Here u^* denotes an exact solution obtained using an analytical expression or solving the underlying problem using a SciPy's library of python. In particular, we consider the following benchmark problems. Now we discuss each problem with its solution one by one.

3.1 Isothermal Gas Sphere

Examine the following nonlinear SBVP that arises in modeling an isothermal gas sphere [68]:

$$u''(x) + \frac{2}{x}u'(x) = -u^5(x), 0 < x < 1 \quad (10)$$

accompanied by the boundary conditions $u'(0) = 0$ and $u(1) = \sqrt{3}/2$. The literature provides the exact solution for this SBVP as $u(x) = \sqrt{\frac{3}{3+x^2}}$ [68].

3.1.1 Solution Comparison and Accuracy

The analysis of the exact and predicted solutions in Figure 2a reveals that the Physics-Informed Neural Network (PINN) effectively approximates the solution throughout the domain $x \in (0,1)$, resulting in a final loss of $\mathcal{L} = 4.96 \times 10^{-8}$; for further details, refer to Table 1. This suggests that the PINN effectively captures the physics of the isothermal gas sphere problem. From a chemical perspective, the isothermal assumption (constant temperature) is key in simulating gas behavior in steady-state processes like adiabatic compression, where density and pressure adjust according to equilibrium conditions. Table 2 presents a detailed comparison between the PINNs predictions and the exact analytical solutions at selected spatial points. The corresponding pointwise absolute errors are also reported, providing a clear view of the prediction accuracy and highlighting the maximum deviations at each stage.

Table 1. Relative error *Erel* and final loss *L*.

Problem	**Isothermal Gas Sphere**	**Thermal Explosion**	**Chemical Reactor Theory**
Erel	2.99e-12	2.50e-08	5.96e-06
L	4.96e-08	7.06e-07	8.81e-07
Epochs	5000	5000	5000

Table 2. Comparison of exact and PINNs predictions at selected points.

x	0.0001	0.1001	0.2001	0.3001	0.4001	0.5001	0.6001	0.7001	0.8001	0.9001	1.0000
Exact	1.0000	0.9983	0.9934	0.9853	0.9743	0.9608	0.9449	0.9271	0.9078	0.8873	0.8660
PINNs	1.0000	0.9983	0.9934	0.9853	0.9743	0.9608	0.9449	0.9271	0.9078	0.8873	0.8660
Error	4.9e-06	6.6e-07	2.06e-06	0	1.01e-06	2.65e-06	2.5e-07	1.45e-06	1.04e-06	1.74e-06	4e-07

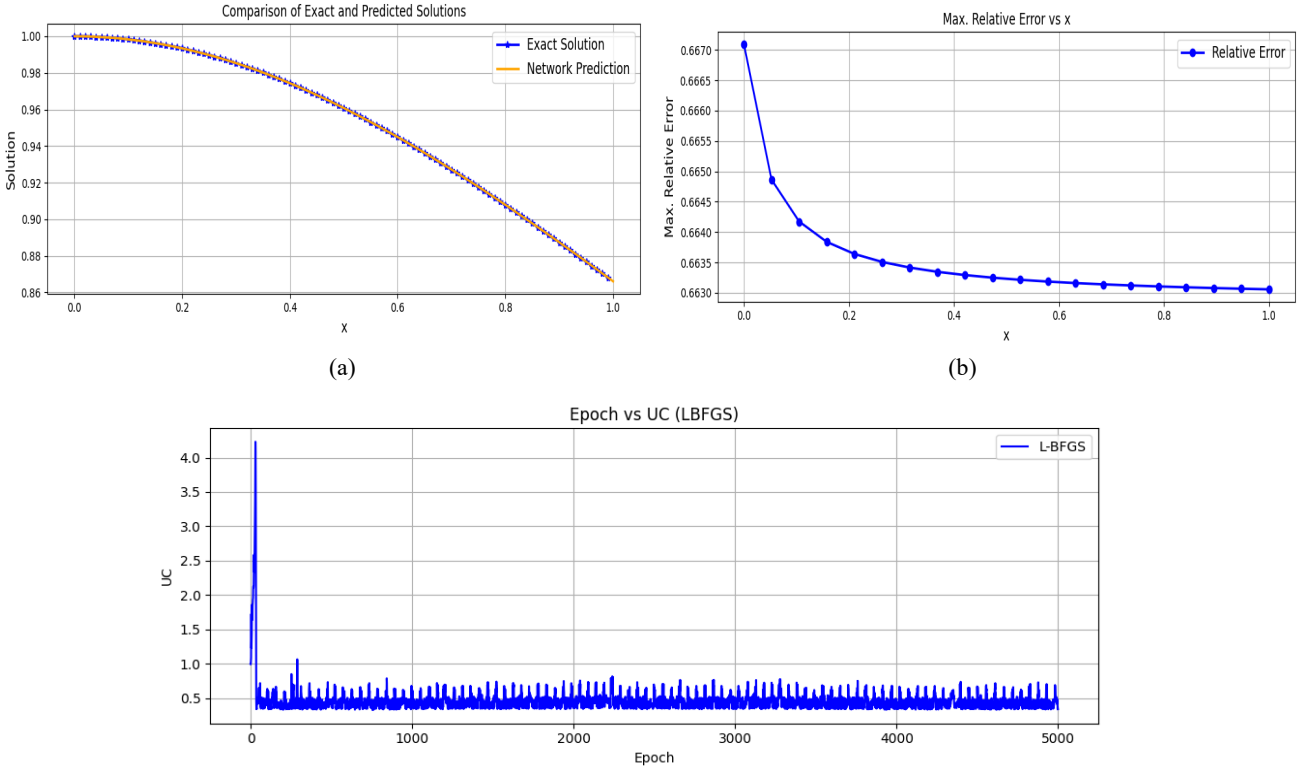

Figure 2. Results for the isothermal gas sphere problem: (a) Comparison between the exact solution and cPINN prediction, (b) Maximum relative error across spatial domain, (c) Optimizer Update cost versus training epochs.

3.1.2 Error Distribution and Chemical Dynamics

In Figure 2b, the relative error is highest at $x = 0$ (around 0.667) and decreases as x increases, stabilizing beyond $x \approx 0.5$. This trend aligns with the behavior of gas spheres, where higher pressure and density near the center cause larger gradients, leading to increased error. As we move away from the center, the pressure and density stabilize, causing the error to decrease. This pattern is consistent with chemical systems, where local deviations from equilibrium are more pronounced near reaction fronts or high-gradient regions.

3.1.3 Optimizer Performance and Chemical Analogy

Figure 2c shows the update cost (UC) over the training epochs when using the L-BFGS optimizer. Initially, the update cost exhibits large fluctuations, reflecting the optimizer's adjustments to minimize the loss function. After the initial phase, the cost stabilizes, indicating convergence. From a chemical dynamics perspective, these fluctuations are akin to the transient phases in a chemical reaction, where rapid adjustments occur before the system stabilizes. This analogy highlights the importance of optimizing parameters during dynamic phases of chemical processes, such as the early stages of combustion or phase transitions.

3.2 Electroactive Polymer Film

The following nonlinear SBVP arises from the analysis of electroactive polymer films [69]:

$$u''(x) + \frac{2}{x}u'(x) = \frac{\alpha u(x)}{1+\mu u(x)}, 0 < x < 1 \quad (11)$$

with the boundary conditions $u'(0) = 0$ and $u(1) = 1$. The equation 11 provides a new analytical closed-form expression for the dimensionless substrate concentration $u(x)$, applicable for all ranges of the saturation parameter μ and the diffusion parameter α. As a result, this equation can be utilized to evaluate how substrate diffusion influences the kinetics of the reaction. The electrode surfaces created with electroactive polymer films have applications in various fields, such as chemical sensor technology, electrocatalysis, and energy storage.

3.2.1 Solution Comparison and Accuracy

The electroactive polymer film problem is modeled by a nonlinear SBVP that characterizes the diffusion and reaction kinetics of a substrate within a polymer film. The Physics-Informed Neural Network (PINN) solutions for different parameter values (α, μ) closely match the exact solutions across the domain $x \in (0,1)$, as shown in Figure 3a, 3c, 3e. The concluding loss values, shown in Table 3, verify the effectiveness of the PINN method:

Case (i): $\alpha = 0.1, \mu = 0.1$, with $\mathcal{L} = 1.3 \times 10^{-8}$.

Case (ii): $\alpha = 5, \mu = 1$, with $\mathcal{L} = 1.98 \times 10^{-6}$.

Case (iii): $\alpha = 0.1, \mu = 1$, with $\mathcal{L} = 4.69 \times 10^{-8}$.

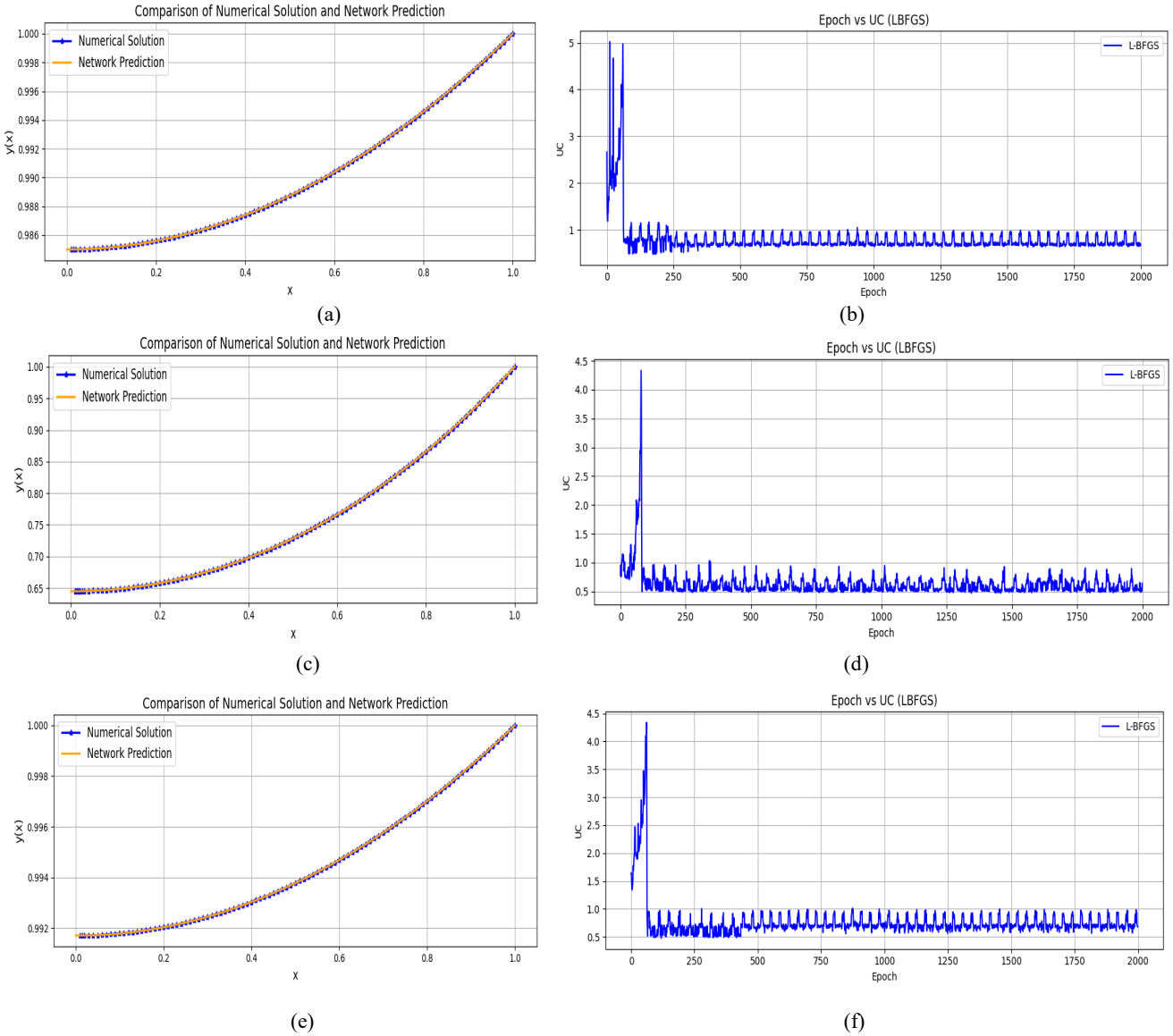

Figure 3. Comparison of PINN-predicted and exact solutions for the electroactive polymer film problem at different parameter settings: (a, c, e) show the solution profiles for cases (i) $\alpha = 0.1, \mu = 0.1$, (ii) $\alpha = 5, \mu = 1$, and (iii) $\alpha = 0.1, \mu = 1$, respectively. Corresponding plots (b, d, f) display the optimizer update cost versus training epochs, illustrating convergence behavior across these cases.

Table 3. Electroactive polymer film final loss $\mathcal{L}$ for different cases of α and μ over 2000 epochs.

-	$\alpha = 0.1, \mu = 0.1$	$\alpha = 5, \mu = 1$	$\alpha = 0.1, \mu = 1$
$\mathcal{L}$	$1.3\text{e}-08$	$1.98\text{e}-06$	$4.69\text{e}-08$

These results indicate that the PINN effectively captures the governing physics, regardless of the diffusion (α) and saturation (μ) parameters. The increase in substrate concentration near $x = 1$ aligns with diffusion-reaction models where substrate accumulates at the outer boundary. Tables 4-6 illustrate the comparison between the FEM and PINNs solutions for three different cases. It is evident from the error rows that PINNs closely approximate the FEM results, demonstrating their accuracy and reliability across all selected points.

Table 4. Comparison of FEM and PINNs predictions at selected points (Case I).

x	0.0000	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000
FEM	0.9849979	0.9851431	0.9855912	0.9863387	0.9873859	0.9887331	0.9903812	0.9923309	0.9945831	0.9971391	1.0000000
PINNs	0.9849991	0.9851422	0.9855928	0.9863383	0.9873871	0.9887351	0.9903820	0.9923306	0.9945841	0.9971395	1.0000005
Error	0.0000012	0.0000008	0.0000016	0.0000004	0.0000012	0.0000020	0.0000008	0.0000003	0.0000010	0.0000004	0.0000005

Table 5. Comparison of FEM and PINNs predictions at selected points (Case II).

x	0.0000	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000
FEM	0.6447837	0.6479590	0.6577982	0.6743232	0.6977103	0.7282045	0.7661144	0.8118070	0.8657023	0.9282655	1.0000000
PINNs	0.6446959	0.6479491	0.6577975	0.6743097	0.6977153	0.7281991	0.7661018	0.8118080	0.8657057	0.9282588	0.9999989
Error	0.0000878	0.0000099	0.0000007	0.0000134	0.0000050	0.0000055	0.0000125	0.0000010	0.0000034	0.0000067	0.0000011

Table 6. Comparison of FEM and PINNs predictions at selected points (Case III).

x	0.0000	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000	0.8000	0.9000	1.0000
FEM	0.9916935	0.9917741	0.9920229	0.9924378	0.9930189	0.9937661	0.9946796	0.9957596	0.9970062	0.9984196	1.0000000
PINNs	0.9917005	0.9917715	0.9920241	0.9924350	0.9930186	0.9937676	0.9946791	0.9957582	0.9970060	0.9984213	0.9999999
Error	0.0000071	0.0000026	0.0000012	0.0000028	0.0000003	0.0000015	0.0000005	0.0000014	0.0000002	0.0000017	0.0000001

3.2.2 Update Cost and Convergence Behavior

The update cost (UC) evolution over training epochs Figure 3b, 3d, 3f provides insights into the optimization dynamics of the PINN. Initially, the UC exhibits large fluctuations, particularly in Case (ii) where $\alpha = 5$, indicating significant adjustments by the optimizer to minimize loss. As the training continues, the cost of updates levels off, indicating that the network is converging to a solution that meets the differential equation and the boundary conditions.

From a numerical optimization perspective, these fluctuations resemble transient phases in stiff differential equations, where rapid initial adjustments occur before reaching a stable solution. In a chemical context, this behavior is analogous to reaction kinetics in catalytic processes, where sharp variations in reaction rates occur initially due to high reactant concentrations before equilibrium is reached. The eventual stabilization of update cost reflects the PINN's ability to efficiently learn the underlying physical dynamics of the problem.

3.2.3 Optimizer Performance and Chemical Analogy

The L-BFGS optimizer used in training efficiently minimizes the loss function, allowing the network to learn the solution even for different parameter regimes. The rapid drop in update cost during early epochs suggests effective parameter updates, similar to the initial fast reaction phase observed in chemical kinetics. The stable convergence in later epochs mirrors equilibrium states in reaction-diffusion systems, reinforcing the PINN's applicability in modeling complex chemical processes such as diffusion-controlled reactions in electroactive polymer films.

3.3 Thermal Explosion Problem

Consider the following nonlinear singular boundary value problem, referred to as the Thermal Explosion Problem [69]:

$$u''(x) + \frac{1}{x}u'(x) = \lambda e^{u(x)}, 0 < x < 1 \quad (12)$$

with the constraints $u'(0) = 0$ and $u(1) = 0$. The exact solution to this nonlinear singular boundary value problem is documented in the literature as:

$$u(x) = 2\log\left(\frac{\psi+1}{\psi x^2+1}\right) \quad (13)$$

where $\psi = -5 + 2\sqrt{6}$ when $\lambda = 1$[69].

3.3.1 Solution Comparison and Accuracy

The analysis presented in Figure 4a illustrates that the Physics-Informed Neural Network (PINN) accurately approximates the solution for the thermal explosion problem within the range $x \in (0,1)$. The significant correlation between the predicted and exact solutions shows that the PINN successfully captures the nonlinear behaviors characteristic of the thermal explosion problem. This accuracy is particularly significant in chemical engineering

applications, where precise modeling of thermal behavior is crucial for predicting reaction stability and preventing runaway conditions. Moreover, the ability of the PINN to approximate solutions to singular boundary value problems highlights its potential for tackling complex thermal and chemical systems. The Table 7 presents a detailed comparison between the exact and PINNs-predicted values at various spatial points, demonstrating the high accuracy of the neural network approximation. The corresponding error values show that the maximum absolute error remains within acceptable limits across the domain.

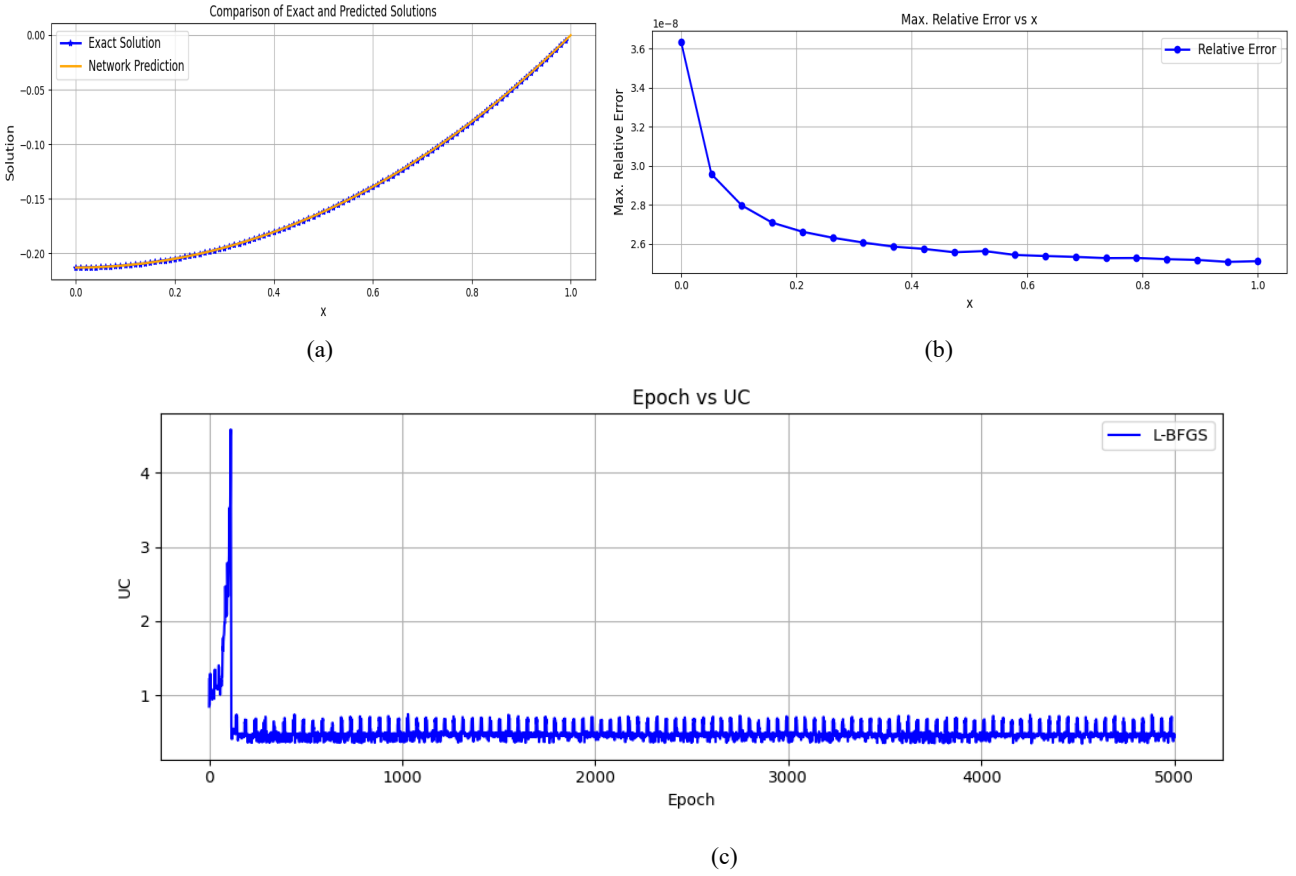

Figure 4. Comparison of predicted and exact solutions for the thermal explosion problem: (a) solution profiles, (b) maximum relative error at different spatial points x, and (c) optimizer update cost versus training epochs demonstrating convergence behavior.

Table 7. Comparison of exact and PINNs predictions at selected points.

x	0.0001	0.1001	0.2001	0.3001	0.4001	0.5001	0.6001	0.7001	0.8001	0.9001	1.0000
Exact	-0.2129897	-0.2109661	-0.2048869	-0.194715	-0.1803877	-0.1618153	-0.1388794	-0.1114303	-0.0792831	-0.04221441	0.0000
PINNs	-0.2130675	-0.2109984	-0.2049181	-0.1947346	-0.1804015	-0.1618348	-0.1388922	-0.1114324	-0.07928911	-0.0422234	-0.000003
Error	7.773e-05	3.226e-05	3.121e-05	1.967e-05	1.378e-05	1.953e-05	1.284e-05	2.110e-06	6.014e-06	8.993e-06	3.014e-06

3.3.2 Error Distribution and Thermal Dynamics

As shown in Figure 4b, the maximum relative error is observed near $x = 0$ and gradually decreases as x approaches 1. This trend aligns with the physical characteristics of the thermal explosion problem, where the temperature gradient is steepest near the center $(x = 0)$ due to the singular nature of the governing differential equation. As x increases, the temperature distribution becomes more uniform, leading to a reduction in error. This behavior is consistent with thermal dynamics in chemical systems, where high gradients near heat sources or reaction centers can result in localized instabilities, while distant regions tend to stabilize. The highest errors occurring near the center of the domain further confirm the physical behavior of thermal systems, where steep gradients often lead to greater numerical discrepancies.

3.3.3 Optimizer Performance and Thermal Analogy

Figure 4c illustrates the update cost (UC) evolution over training epochs when using the L-BFGS optimizer. Initially, the update cost exhibits significant fluctuations, reflecting the optimizer's adaptation to the nonlinear and singular nature of the problem. As training progresses, the update cost stabilizes, indicating convergence to an optimal solution. From a thermal dynamics perspective, these initial fluctuations resemble the transient phase of a thermal process, where rapid

changes in temperature and heat flux occur before the system reaches a steady state. This analogy highlights the importance of optimizing parameters during transient phases in thermal processes, such as ignition or cooling, to ensure accurate and stable predictions. The stabilization of the update cost further reinforces the effectiveness of the PINN in handling nonlinear singular boundary value problems.

3.4 Chemical Reactor Theory

Consider the singularly perturbed boundary value problem below, which is arises from the theory of chemical reactors [70]:

$$\epsilon u''(x) + u'(x) - u^2 = 0, 0 < x < 1 \quad (14)$$

with the boundary conditions $u(0) = 0$ and $u(1) + \epsilon u'(1) = 1$. The exact solution for this nonlinear singularly perturbed boundary value problem is presented in the existing literature as [71]:

$$u(x) = \frac{1}{2-x} + \epsilon \frac{e^{-x/\epsilon}}{4} \quad (15)$$

3.4.1 Solution Comparison and Accuracy

Figure 5a compares the exact solution with the predicted solution obtained using the Physics-Informed Neural Network (PINN) for the chemical reactor theory problem. The PINN closely approximates the exact solution across the domain $x \in (0,1)$, demonstrating its ability to capture the complex dynamics of the singularly perturbed boundary value problem (SPBVP). The accuracy of the PINN is particularly important in chemical reactor theory, where precise modeling of reaction kinetics and diffusion processes is crucial for optimizing reactor performance. The close agreement between the predicted and exact solutions highlights the effectiveness of the PINN in handling SPBVPs, even in the presence of a small perturbation parameter ϵ. The Table 8 compares the exact and predicted solutions at selected spatial points using PINNs. The prediction closely follows the exact solution, with a nearly uniform small error across the domain. The results demonstrate the consistency and accuracy of the PINNs approximation for this problem.

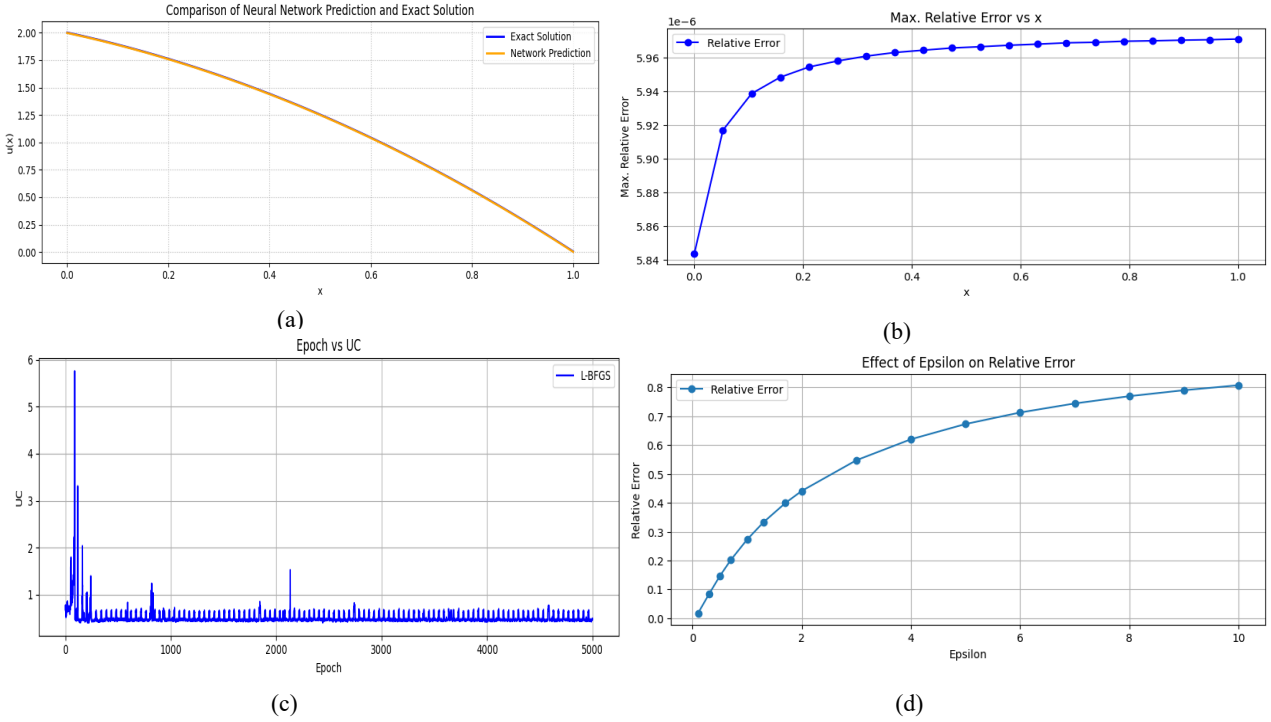

Figure 5. PINN-predicted and exact solutions for the chemical reactor problem: (a) solution profiles, (b) maximum relative error across spatial domain, (c) optimizer update cost versus training epochs, and (d) influence of parameter ϵ on the relative error, illustrating model sensitivity.

Table 8. Comparison of Exact and PINNs predictions at selected points.

x	0.0001	0.1001	0.2001	0.3001	0.4001	0.5001	0.6001	0.7001	0.8001	0.9001	1.0000
Exact	2.001506	1.891857	1.762160	1.612461	1.442761	1.253058	1.043353	0.8136467	0.5639378	0.2942272	0.0048141
PINNs	1.998317	1.888699	1.758943	1.609242	1.439625	1.249882	1.040091	0.8104342	0.5608223	0.2910324	0.0016134
Error	0.003189	0.003157	0.003217	0.003219	0.003135	0.003176	0.003262	0.003213	0.003115	0.003195	0.003201

3.4.2 Error Distribution and Reactor Dynamics

Figure 5b shows the maximum relative error distribution across the domain. The error is highest near $x = 0$ and decreases as x approaches 1 . This trend is consistent with the physical behavior of chemical reactors, where steep gradients near the reactor inlet (at $x = 0$) lead to higher numerical errors. As the reaction progresses and the system stabilizes, the error decreases, reflecting the smoother variation in concentration and reaction rates. This behavior aligns with the nature of SPBVPs, where the perturbation parameter ϵ introduces a thin boundary layer near $x = 0$, making the solution more challenging to approximate in this region.

3.4.3 Optimizer Performance and Reactor Analogy

Figure 5c illustrates the update cost (UC) over training epochs when using the L-BFGS optimizer. Initially, the update cost exhibits significant fluctuations, indicating the optimizer's struggle to minimize the loss function in the presence of the singular perturbation. As training progresses, the update cost stabilizes, reflecting the convergence of the network to an optimal solution. This behavior is analogous to the transient phase in chemical reactors, where rapid changes in reactant concentrations and reaction rates occur before the system reaches a steady state. The stabilization of the update cost underscores the PINN's ability to handle the stiff nature of SPBVPs, making it a valuable tool for simulating complex chemical processes.

3.4.4 Effect of Perturbation Parameter ϵ

Figure 5d illustrates how the perturbation parameter ϵ influences the relative error. When ϵ decreases, the relative error rises, especially close to $x = 0$. This is expected, as smaller values of ϵ intensify the singular nature of the problem, leading to sharper gradients and more pronounced boundary layers. The increase in error with decreasing ϵ highlights the challenges associated with solving SPBVPs using traditional numerical methods. However, the PINN's ability to maintain relatively low errors across different values of ϵ showcases its robustness and adaptability in handling singularly perturbed problems in chemical reactor theory.

Remark 1 In this study, we have validated the PINN-based solutions by comparing them with corresponding analytical solutions, which serve as reliable benchmarks for assessing accuracy. Since the selected problems are classical test cases with known exact solutions, this direct comparison offers more rigorous validation than refer-encing numerical results from the literature. However, we acknowledge that future work may include comparisons with existing numerical methods to provide broader benchmarking and contextual relevance.

Remark 2 To validate the performance of our proposed cPINNs, we compare our solutions with exact analytical solutions in three benchmark problems (see Sections 3.1, 3.3, and 3.4) and with the FEM in one problem (see Section 3.2). This provides both an accuracy benchmark and a quantitative comparison with a traditional numerical method, demonstrating the effectiveness and generaliz-ability of cPINNs.

4. Conclusion

In this research, we demonstrated the effectiveness of causality-enhanced Physics-Informed Neural Networks (cPINNs) in solving singular and singularly perturbed boundary value problems (SBVPs and SPBVPs) arising in chemical systems, in-cluding isothermal gas spheres, electroactive polymer films, thermal explosions, and chemical reactor theory. By integrating causal training and optimizing the pertur-bation parameter ϵ alongside network weights, our approach achieved high accuracy and robustness across diverse benchmark problems. The L-BFGS optimizer and adaptive loss weighting further improved convergence, particularly for stiff and mul-tiscale systems. Comparisons with analytical solutions and FEM results validated the method’s reliability, while transfer learning reduced computational costs for related tasks.

5. Limitations and Future Directions

Despite these advancements, several challenges and opportunities remain:

High-Dimensional Systems: The current framework focuses on 1D problems. Extending cPINNs to higher-dimensional chemical systems (e.g., 3D reactor models) would require architectural adjustments to manage computational complexity.

Generalizability: While cPINNs excel for specific SBVP/SPBVP classes, their performance on problems with discontinuous or stochastic coefficients warrants further investigation.

Training Efficiency: The reliance on L-BFGS, though effective for smooth loss landscapes, may not scale efficiently to larger networks. Hybrid optimizers (e.g., Adam-LBFGS) could be explored.

Experimental Validation: Future work should incorporate experimental data to bridge the gap between theoretical predictions and real-world chemical systems.

Uncertainty Quantification: Integrating Bayesian frameworks or ensemble methods could enhance cPINNs' ability to quantify prediction uncertainties, critical for safety-sensitive applications like combustion or reactor design.

This study bridges traditional numerical methods and modern machine learning, offering a scalable tool for complex chemical BVPs. Future efforts will focus on addressing these limitations while expanding applications to coupled multiphysics problems and industrial-scale simulations.

Data availability

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Acknowledgments

The author expresses heartfelt gratitude to their parents for their unwavering support, encouragement, and sacrifices that have been instrumental in the completion of this work.

Conflict of Interest

Not applicable.

References

[1] Mehrpouya MA. An efficient pseudospectral method for numerical solution of nonlinear singular initial and boundary value problems arising in astrophysics. Mathematical Methods in the Applied Sciences, 2016, 39(12), 3204-3214. DOI: 10.1002/mma.3763

[2] Ali MR, Hadhoud AR, Ma WX. Evolutionary numeri-cal approach for solving nonlinear singular periodic boundary value problems. Journal of Intelligent & Fuzzy Systems, 2020, 39(5), 7723-7731. DOI: 10.3233/JIFS-201045

[3] Adam JA, Maggelakis SA. Mathematical models of tumor growth. IV. Effects of a necrotic core. Mathematical Biosciences, 1989, 97(1), 121-136. DOI: 10.1016/0025-5564(89)90045-X

[4] Roul P. A fourth-order non-uniform mesh optimal B-spline collocation method for solving a strongly nonlinear singular boundary value problem de-scribing electrohydrodynamic flow of a fluid. Applied Numerical Mathematics, 2020, 153, 558-574. DOI: 10.1016/j.apnum.2020.03.018

[5] Kumar M, Singh N. Modified adomian decomposition method and computer implementation for solving singular boundary value problems arising in various physical problems. Computers & Chemical Engineering, 2010, 34(11), 1750-1760. DOI: 10.1016/j.compchemeng.2010.02.035

[6] Khuri SA, Sayfy A. A novel approach for the solution of a class of singular boundary value problems arising in physiology. Mathematical and Computer Modelling, 2010, 52(3-4), 626-636. DOI: 10.1016/j.mcm.2010.04.009

[7] Chawla MM, Katti CP. A uniform mesh finite difference method for a class of singular two-point boundary value problems. SIAM Journal on Numerical Analysis, 1985, 22(3), 561-565. DOI: 10.1137/0722033

[8] Roul P, Goura VP, Agarwal R. A compact finite differ-ence method for a general class of nonlinear singular boundary value problems with neumann and robin boundary conditions. Applied Mathematics and Computation, 2019, 350, 283-304. DOI: 10.1016/j.amc.2019.01.001

[9] De Hoog FR, Weiss R. Collocation methods for singular boundary value problems. SIAM Journal on Numerical Analysis, 1978, 15(1), 198-217. DOI: 10.1137/0715013

[10] Roul P, Thula K. A new high-order numerical method for solving singular two-point boundary value problems. Journal of Computational and Applied Mathematics, 2018, 343, 556-574. DOI: 10.1016/j.cam.2018.04.056

[11] Turkyilmazoglu M. Effective computation of exact and analytic approximate solutions to singular nonlinear equations of lane-emden-fowler type. Applied Mathematical Modelling, 2013, 37(14-15), 7539-7548. DOI: 10.1016/j.apm.2013.02.014

[12] Babolian E, Eftekhari A, Saadatmandi A. A sinc-galerkin technique for the numerical solution of a class of singular boundary value problems. Compu-tational and Applied Mathematics, 2015, 34(1), 45-63. DOI: 10.1007/s40314-013-0103-x

[13] Kanth AR, Bhattacharya V. Cubic spline for a class of non-linear singular boundary value problems arising in physiology. Applied Mathematics and Computation, 2006, 174(1), 768-774. DOI: 10.1016/j.amc.2005.05.022

[14] Çağlar H, Çağlar N, Özer M. B-spline solution of non-linear singular boundary value problems arising in physiology. Chaos, Solitons & Fractals, 2009, 39(3), 1232-1237. DOI: 10.1016/j.chaos.2007.06.007

[15] Kanth AR, Aruna K. He's variational iteration method for treating non-linear singular boundary value problems. Computers & Mathematics with Applications, 2010, 60(3), 821-829. DOI: 10.1016/j.camwa.2010.05.029

[16] Wazwaz AM. The variational iteration method for solving nonlinear singular boundary value problems arising in various physical models. Communications in Nonlinear Science and Numerical Simulation, 2011, 16(10), 3881-3886. DOI: 10.1016/j.cnsns.2011.02.026

[17] Wang F, Salama SA, Khater MM. Optical wave solutions of perturbed time-fractional nonlinear schr¨odinger equation. Journal of Ocean Engineering and Science, 2022. DOI: 10.1016/j.joes.2022.03.014

[18] El-Kalla IL, El Mhlawy AM, Botros M. A continuous solution of solving a class of nonlinear two-point boundary value problem using adomian decomposition method. Ain Shams Engineering Journal, 2019, 10(1), 211-216. DOI: 10.1016/j.asej.2018.11.002

[19] Ebaid A. A new analytical and numerical treatment for singular two-point boundary value problems via the adomian decomposition method. Journal of Computational and Applied Mathematics, 2011, 235(8), 1914-1924. DOI: 10.1016/j.cam.2010.09.007
[20] Umesh, Kumar M. Numerical solution of singular boundary value problems using advanced adomian decomposition method. Engineering with Computers, 2021, 37(4), 2853-2863. DOI: 10.1007/s00366-020-00972-6
[21] Roos HG, Stynes M, Tobiska L. Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems, volume 24. Springer, Berlin Heidelberg, 2008.
[22] O'malley RE. Singular Perturbation Methods for Ordinary Differential Equations, volume 89. Springer, New York, 1991.
[23] Lin B, Li K, Cheng Z. B-spline solution of a singularly perturbed boundary value problem arising in biology. Chaos, Solitons & Fractals, 2009, 42(5), 2934-2948. DOI: 10.1016/j.chaos.2009.04.036
[24] Natesan S, Ramanujam N. Initial-value technique for singularly perturbed boundary-value problems for second-order ordinary differential equations arising in chemical reactor theory. Journal of Optimization Theory and Applications, 1998, 97(2), 455-470. DOI: 10.1023/A:1022639003366
[25] Glizer VY. Asymptotic solution of a boundary-value problem for linear singularly-perturbed functional differential equations arising in optimal control theory. Journal of Optimization Theory and Applications, 2000, 106(2), 309-335. DOI: 10.1023/A:1004651430364
[26] El-Zahar ER, El-Kabeir SM. A new method for solving singularly perturbed boundary value problems. Applied Mathematics & Information Sciences, 2013, 7(3), 927.
[27] Jiwrai R, Mittal RC. A higher order numerical scheme for singularly perturbed burger-huxley equation. Journal of Applied Mathematics & Informatics, 2011, 29(34), 813-829. DOI: 10.14317/jami.2011.29.3_4.813
[28] Kadalbajoo MK, Reddy YN. Initial-value technique for a class of nonlinear singular perturbation problems. Journal of Optimization Theory and Applications, 1987, 53(3), 395-406. DOI: 10.1007/BF00938946
[29] Gasparo MG, Macconi M. New initial-value method for singularly perturbed boundary-value problems. Journal of Optimization Theory and Applications, 1989, 63(2), 213-224. DOI: 10.1007/BF00939575
[30] El-Zahar ER. Approximate analytical solution of singularly perturbed boundary value problems in maple. AIMS Mathematics, 2020, 5(3), 2272-2284.
[31] Gasparo MG, Macconi M. Initial-value methods for second-order singularly perturbed boundary-value problems. Journal of Optimization Theory and Applications, 1990, 66(2), 197-210. DOI: 10.1007/BF00939534
[32] Reddy YN, Chakravarthy PP. An initial-value approach for solving singularly perturbed two-point boundary value problems. Applied Mathematics and Computation, 2004, 155(1), 95-110. DOI: 10.1016/S0096-3003(03)00763-X
[33] Mustafa G, Ejaz ST, Baleanu D, Ghaffar A, Nisar KS. A subdivision-based approach for singularly perturbed boundary value problem. Advances in Difference Equations, 2020, 2020(1), 1-20. DOI: 10.1186/s13662-020-02732-8
[34] Lubuma JM, Patidar KC. Uniformly convergent non-standard finite difference methods for self-adjoint singular perturbation problems. Journal of Computational and Applied Mathematics, 2006, 191(2), 228-238. DOI: 10.1016/j.cam.2005.06.039
[35] Niijima K. On a three-point difference scheme for a singular perturbation problem without a first derivative term i. Memoirs of Numerical Mathematics, 1980, 7, 1-10.
[36] Aziz T, Khan A. A spline method for second-order singularly perturbed boundary-value problems. Journal of Computational and Applied Mathematics, 2002, 147(2), 445-452. DOI: 10.1016/S0377-0427(02)00479-X
[37] Khan I, Aziz T. Tension spline method for second-order singularly perturbed boundary-value problems. International Journal of Computer Mathematics, 2005, 82(12), 1547-1553. DOI: 10.1080/00207160410001684280
[38] Bawa RK, Natesan S. A computational method for self-adjoint singular perturbation problems using quintic spline. Computers & Mathematics with Applications, 2005, 50(8-9), 1371-1382. DOI: 10.1016/j.camwa.2005.04.017
[39] Kumar V, Mehra M. Wavelet optimized finite difference method using interpolating wavelets for self-adjoint singularly perturbed problems. Journal of Computational and Applied Mathematics, 2009, 230(2), 803-812. DOI: 10.1016/j.cam.2009.01.017
[40] Pandit S, Kumar M. Haar wavelet approach for numerical solution of two parameters singularly perturbed boundary value problems. Applied Mathematics & Information Sciences, 2014, 8(6), 2965. DOI: 10.12785/amis/080634
[41] Gadyl'shin RR. Concordance method of asymptotic expansions in a singularly-perturbed boundary-value problem for the laplace operator. Journal of Mathematical Sciences, 2005, 125(5), 579-609. DOI: 10.1007/s10958-005-0062-x
[42] Weinan E, Han J, Jentzen A. Algorithms for solving high dimensional PDEs: from nonlinear Monte Carlo to machine learning. Nonlinearity, 2021, 35(1), 278. DOI: 10.1088/1361-6544/ac337f
[43] Madenci E, Guven I. The finite element method and applications in engineering using ANSYS. Springer, 2015.
[44] E W, Yu B. The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 2018, 6(1), 1-12. DOI: 10.1007/s40304-018-0127-z
[45] JSirignano J, Spiliopoulos K. Dgm: A deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 2018, 375, 1339-1364. DOI: 10.1016/j.jcp.2018.08.029
[46] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 2019, 378, 686-707. DOI: 10.1016/j.jcp.2018.10.045
[47] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint, 2014. DOI: 10.48550/arXiv.1412.6980
[48] Lagaris IE, Likas A, Fotiadis DI. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 1998, 9(5), 987-1000. DOI: 10.1109/72.712178
[49] Bottou L. Stochastic gradient descent tricks. Neural Networks: Tricks of the Trade, 2nd ed., Springer: Berlin, 2012; pp. 421-436. DOI: 10.1007/978-3-642-35289-8_25
[50] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing, 2021, 43(5), A3055-A3081. DOI: 10.1137/20M1318043
[51] Gao B, Yao R, Li Y. Physics-informed neural networks with adaptive loss weighting algorithm for solving partial differential equations. Computers & Mathematics with Applications, 2025, 181, 216-227. DOI: 10.1016/j.camwa.2025.01.007

[52] Dolean V, Heinlein A, Mishra S, Moseley B. Multilevel domain decomposition-based architectures for physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering, 2024, 429, 117116. DOI: 10.1016/j.cma.2024.117116
[53] Li R, Lee E, Luo T. Physics-informed neural networks for solving multiscale mode-resolved phonon Boltzmann transport equation. Materials Today Physics, 2021, 19, 100429. DOI: 10.1016/j.mtphys.2021.100429
[54] Fang Z, Wang S, Perdikaris P. Ensemble learning for physics informed neural networks: A gradient boosting approach. arXiv preprint arXiv:2302.13143, 2023. DOI: 10.48550/arXiv.2302.13143
[55] Martinek P, Krammer O. Optimising pin-in-paste technology using gradient boosted decision trees. Soldering & Surface Mount Technology, 2018, 30(3), 164-170. DOI: 10.1108/SSMT-09-2017-0024
[56] Jagtap AD, Kawaguchi K, Em Karniadakis G. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. Proceedings of the National Academy of Sciences, 2020, 476(2239), 21948-21956. DOI: 10.1098/rspa.2020.0334
[57] Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics, 2020, 404, 109136. DOI: 10.1016/j.jcp.2019.109136
[58] Botev A, Ritter H, Barber D. Practical gauss-newton optimization for deep learning. InInternational Conference on Machine Learning, 2017, 70, 557-565.
[59] Hao W, Hong Q, Jin X. Gauss Newton method for solving variational problems of PDEs with neural network discretizaitons. Journal of Scientific Computing, 2024, 100(1), 17. DOI: 10.1007/s10915-024-02535-z
[60] Gratton S, Mercier V, Riccietti E, Toint PL. A block-coordinate approach of multi-level optimization with an application to physics-informed neural networks. Computational Optimization and Applications, 2024, 89(2), 385-417. DOI: 10.1007/s10589-024-00597-1
[61] Golub G, Pereyra V. Separable nonlinear least squares: the variable projection method and its applications. Inverse Problems, 2003, 19(2). DOI: 10.1088/0266-5611/19/2/201
[62] Kim CT, Lee JJ. Training two-layered feedforward networks with variable projection method. IEEE Transactions on Neural Networks, 2008, 19(2),371-375. DOI: 10.1109/TNN.2007.911739
[63] Dong S, Yang J. Numerical approximation of partial differential equations by a variable projection method with artificial neural networks. Computer Methods in Applied Mechanics and Engineering, 2022, 398, 115284. DOI: 10.1016/j.cma.2022.115284
[64] Huang X, Liu H, Shi B, Wang Z, Yang K, et al. A Universal PINNs Method for Solving Partial Differential Equations with a Point Source, International Joint Conferences on Artificial Intelligence, 2022, 3839-3846. DOI: 10.24963/ijcai.2022/533
[65] Zubov K, McCarthy Z, Ma Y, Calisto F, Pagliarino V, et al. Neuralpde: Automating physics-informed neural networks (pinns) with error approximations. arXiv preprint arXiv:2107.09443, 2021. DOI: 10.48550/arXiv.2107.09443
[66] Nash SG, Nocedal J. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. SIAM Journal on Optimization, 1991, 1(3), 358-372. DOI: 10.1137/0801023
[67] Nocedal J. Updating quasi-newton matrices with limited storage. Mathematics of Computation, 1980, 35(151), 773-782.
[68] Roul P, Goura VP. A fast numerical scheme for solving singular boundary value problems arising in various physical models. Journal of Mathematical Chemistry, 2022, 60(3), 514-541. DOI: 10.1007/s10910-021-01316-5
[69] Roul P, Kumari T. A quartic trigonometric B-spline collocation method for a general class of nonlinear singular boundary value problems. Journal of Mathematical Chemistry, 2022, 1-17. DOI: 10.1007/s10910-021-01293-9
[70] Natesan S, Ramanujam N. A “booster method” for singular perturbation problems arising in chemical reactor theory. Applied mathematics and computation, 1999, 100(1), 27-48. DOI: 10.1016/S0096-3003(98)00014-9
[71] Celik E, Tunc H, Sari M. An efficient multi-derivative numerical method for chemical boundary value problems. Journal of Mathematical Chemistry, 2024, 62(3), 634-653. DOI: 10.1007/s10910-023-01556-7