

Modified PINN approach integrating conservation laws for efficient multi-stage training of coupled nonlinear systems

Jie Deng(邓婕) and Lijia Han(韩励佳)

Citation: Chin. Phys. B, 2026, 35 (**4**): 040202. DOI: 10.1088/1674-1056/ae030b
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

LT-PINNs: Physics-informed neural networks based on Laplace transform for solving Caputo-type fractional partial differential equations

Ruibo Zhang(张瑞波), Fengjun Li(李风军), and Jianqiang Liu(刘建强)
Chin. Phys. B, 2026, 35 (**3**): 030201. DOI: 10.1088/1674-1056/adf827

MetaPINNs: Predicting soliton and rogue wave of nonlinear PDEs via the improved physics-informed neural networks based on meta-learned optimization

Yanan Guo(郭亚楠), Xiaoqun Cao(曹小群), Junqiang Song(宋君强), and Hongze Leng(冷洪泽)
Chin. Phys. B, 2024, 33 (**2**): 020203. DOI: 10.1088/1674-1056/ad0bf4

From breather solutions to lump solutions: A construction method for the Zakharov equation

Feng Yuan(袁丰), Behzad Ghanbari, Yongshuai Zhang(张永帅), and Abdul Majid Wazwaz
Chin. Phys. B, 2023, 32 (**12**): 120201. DOI: 10.1088/1674-1056/acf706

Meshfree-based physics-informed neural networks for the unsteady Oseen equations

Keyi Peng(彭珂依), Jing Yue(岳靖), Wen Zhang(张文), and Jian Li(李剑)
Chin. Phys. B, 2023, 32 (**4**): 040207. DOI: 10.1088/1674-1056/ac9cb9

Data-driven parity-time-symmetric vector rogue wave solutions of multi-component nonlinear Schrödinger equation

Li-Jun Chang(常莉君), Yi-Fan Mo(莫一凡), Li-Ming Ling(凌黎明), and De-Lu Zeng(曾德炉)
Chin. Phys. B, 2022, 31 (**6**): 060201. DOI: 10.1088/1674-1056/ac4cc5

Modified PINN approach integrating conservation laws for efficient multi-stage training of coupled nonlinear systems

Jie Deng(邓婕) and Lijia Han(韩励佳)†

Department of Mathematics and Physics, North China Electric Power University, Beijing 102206, *China*

(Received 21 July 2025; revised manuscript received 25 August 2025; accepted manuscript online 4 September 2025)

We propose an innovative conservation multi-stage algorithm (CM) based on physics-informed neural networks (PINNs) to investigate the dynamics of vector solitons governed by Zakharov equation. By ingeniously integrating conservation laws into the multi-stage training algorithm, we enhanced the method's constraint and physical consistency in solving nonlinear systems. Numerical simulations of the Zakharov and nonlinear Schrödinger (NLS) equations demonstrated that our method showed obvious improvements in approximation accuracy, convergence speed, and training efficiency, compared to traditional PINN methods. Moreover, when the parameter representing the speed of sound is sufficiently large, our method efficiently simulates the approximation from the Zakharov equation to the NLS equation. The approximative simulation not only confirms the conservation multi-stage algorithm's applicability in various physical environments but also highlights its potential in controlling sound wave propagation characteristics to simulate NLS equation behavior.

Keywords: physics-informed neural network, Zakharov equation, nonlinear Schrödinger equation, limit behavior

PACS: 02.60.–x, 52.35.–g, 02.30.Jr, 02.60.Cb

DOI: 10.1088/1674-1056/ae030b **CSTR:** 32038.14.CPB.ae030b

1. Introduction

The soliton phenomenon has important applications in various scientific fields such as physics, biology, and communication science.[1] The Zakharov equation is a key model in plasma physics, describing the coupled dynamics of Langmuir waves (high-frequency electron oscillations) and sound waves (low-frequency ion compressional waves) in a medium.[2] Moreover, the solution of the Zakharov equation successfully solves the deep cavity phenomenon observed in laser-target experiments. The classical Zakharov equation can be represented as

$$\begin{cases} \dfrac{1}{c_0^2}\dfrac{\partial^2 n}{\partial t^2} - \Delta n = \nabla^2 |\epsilon|^2, \\ \mathrm{i}\dfrac{\partial \epsilon}{\partial t} + \Delta \epsilon - n\epsilon = 0, \end{cases} \tag{1}$$

where ϵ represents the complex amplitude of the Langmuir wave, n is the perturbed density of the ion medium, and c_0 is the speed of sound in the medium. In this paper, we consider the one-dimensional case, where $\Delta = \nabla^2 = \partial^2/\partial x^2$. There are important connections between Zakharov equation and nonlinear Schrödinger (NLS) equation, when $c_0 \to \infty$, the solution of Zakharov equation will approximate to the NLS equation

$$\mathrm{i}\epsilon_t + \epsilon_{xx} + |\epsilon|^2\epsilon = 0. \tag{2}$$

It is well known that equation (2) plays an important role in mathematical physics. This approximation establishes a theoretical framework for exploring the propagation and stability of plasma waves under various physical conditions and across different scales. By applying classical methods such as the inverse scattering method, the Hirota method, Bäcklund and Darboux transformations, the exact soliton solutions of the nonlinear Schrödinger equation are obtained, see Refs. [3–5] and the references therein. Additionaly, in the field of numerical methods for solving PDEs, there are many widely used methods, such as the finite element method, the finite volume method, the spectral method, and the finite difference method, often referred to as traditional numerical methods, see Refs. [6–9] and the references therein. However, to achieve higher accuracy, traditional numerical methods often require sufficiently fine grids of solution domains, which can result in significant computational costs.[10]

In 2019, Raissi, Perdikaris, and Karniadakis introduced physics-informed neural networks (PINNs) to overcome the limitations of conventional numerical approaches. This innovative framework has since emerged as a powerful solution for addressing physics and engineering problems, especially in solving complex partial differential equations (PDEs).[11–16] PINNs incorporates physical laws directly into the neural network's loss function, optimizing the model learning process while weakly enforcing physical constraints. Compared to traditional numerical methods, PINNs avoid high computational costs and complex grid discretization. They show clear advantages for both high-dimensional and low-dimensional problems. For high-dimensional cases, PINNs effectively bypass the "curse of dimensionality". In low-dimensional prob-

†Corresponding author. E-mail: hljmath@ncepu.edu.cn

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

lems, they improve data efficiency, maintain solution consistency, and provide a unified framework for solving both forward and inverse problems.[17] Based on these advantages, the PINNs methods have also been applied to solve various types of PDEs, and many variants have also subsequently emerged.[17–23] Nevertheless, PINNs still have difficulties in convergence and accuracy, especially for high nonlinearity and strong coupling dynamics.[24,25]

Recent developments have been made to improve both the accuracy and convergence rate of the original PINN method. For instance, the residual-based adaptive refinement (RAR) algorithm[26] enhances the selection of collocation points by concentrating computational resources on regions with higher error. Similarly, a time-adaptive sampling algorithm[27] boosts temporal resolution and model efficiency by adjusting the sampling points dynamically during training. The GP algorithm[28] addresses the issue of imbalanced backpropagated gradients by using adaptive weights. Building upon these advancements,[26–28] the pre-fixed multi-stage training algorithm[29] was introduced, combining error measurement, adaptive sampling, and adaptive weighting strategies. This method notably enhanced both training efficiency and model performance. In parallel, researchers have explored the incorporation of conservation law constraints using different approaches. For example, Jagtap *et al.*[30] utilized conservation laws through discrete domain formulations, while Gurieva *et al.*[31] incorporated conserved quantity residuals into loss functions for specific equations. A more recent study[32] proposed novel methods for conservation-aware training. While these methods improve physical consistency, they mainly focus on static approaches, such as particular discretization techniques,[30] modifications to loss terms,[31] or fixed constraints,[33,34] without accounting for dynamic, multi-stage frameworks.

Based on this, we propose a conservation multi-stage algorithm (CM). This method integrates two established strategies: the multi-stage adaptive framework[29] for dynamic resource optimization, and conservation law constraints[33,34] to enforce physical consistency. The CM algorithm embeds conservation laws directly into each training stage, achieving a dynamic balance between computational efficiency and physical constraints. Compared with the work of Mo *et al.*,[29] which utilized deep learning to solve vector solitons in coupled NLS equations, our CM algorithm extends the research scope to the more complex Zakharov equation. Building on the "two-stage PINN method based on conserved quantities" proposed by Lin and Chen,[35] we adopt a multi-stage design to further optimize training efficiency and scalability. Numerical experiments demonstrate that CM algorithm exhibits improvements in both accuracy and convergence rate compared to isolated implementations of multi-stage training or conservation-loss methods. As far as we know, few studies have explored the application of PINNs to simulate the classical Zakharov equation and its approximation to the NLS equation, see Refs. [36,37].

We conduct numerical simulations for the Zakharov equation and the NLS equation using the CM algorithm and compare the results obtained by original PINNs, conservation law constraint algorithm, and pre-fixed multi-stage training algorithm (see A1, A2, A3 in Table 1). The numerical simulations show that our algorithm provides obvious improvements in terms of accuracy and convergence speed. Moreover, we explore how the solutions of the Zakharov equation approximate those of the NLS equation as the speed of sound c_0 gradually increases. The numerical results reveal the dynamic relationship between the solutions under specific physical conditions and confirm the theoretical predictions in Refs. [38–42]. These numerical simulations demonstrat that the CM algorithm effectively reduces errors and accelerates the rate of decline of the loss function. Additionally, its applicability in exploring complex physical systems is also highlighted. As far as we know, there are no results about using the PINN method to simulate the solution of the Zakharov equation and investigate how the solution of the Zakharov equation approximates the solution of the NLS equation as the speed of sound tends to infinity.

The structure of this paper is as follows: Section 2 introduces the original PINN method and the CM algorithm, and discusses the key details such as training data and loss functions. Section 3 presents the numerical simulation results of the Zakharov equation and compares them with other methods. Section 4 investigates how the solutions of Zakharov equation approximate those of NLS equation as c_0 increases. Section 5 summarizes the main conclusions of the study and proposes directions for future research.

2. Conservation multi-stage algorithm

In this paper, we consider the focusing Zakharov equation with initial and Dirichlet boundary conditions as follows:

$$\begin{cases} \dfrac{1}{c_0^2}\dfrac{\partial^2 n}{\partial t^2} - \Delta n = \nabla^2 |\epsilon|^2, & x \in [-x_L, x_L], t \in [t_0, t_T], \\ \mathrm{i}\dfrac{\partial \epsilon}{\partial t} + \Delta \epsilon - n\epsilon = 0, & x \in [-x_L, x_L], t \in [t_0, t_T], \\ \epsilon(x, t_0) = \epsilon_0(x), n(x, t_0) = n_0(x), n_t(x, t_0) = n_1(x), & x \in [-x_L, x_L], \\ \epsilon(x_L, t) = n(x_L, t) = n_t(x_L, t) = 0, & t \in [t_0, t_T]. \end{cases} \tag{3}$$

Building on these theoretical results, we now shift our focus to solving the Zakharov equation numerically. We begin by briefly reviewing the original PINN method for solving the Zakharov equation. Subsequently, we introduce our proposed conservation multi-stage algorithm, designed to enhance both the efficiency and accuracy of the original PINN method.

2.1. Physics-informed neural networks for solving Zakharov equation

By combining deep learning knowledge with physical prior knowledge, a PINN model can be constructed for numerical simulation of soliton solutions of the Zakharov equation.

Define a fully connected neural network with n layers $f_\theta(x,t)$, where the inputs are spatial and temporal coordinates, and the outputs are predicted physical quantities, such as the function values at specific coordinates. The parameters of the neural network $\Theta = \{w,b\}$ are updated based on the gradient information from the loss function using backpropagation. Let x_i represent the output of the i-th layer of the neural network, with n_i neurons in the i-th layer. The transformation from layer $i-1$ to layer i can be represented as

$$x_i = \sigma(w_i x_{i-1} + b_i), \quad i = 1,\ldots,n, \tag{4}$$

where $w_i \in \boldsymbol{R}^{n_{i+1}\times n_i}$ represents the weight matrix of the i-th layer, b_i is a vector of length n_i representing the bias term, and σ is the nonlinear activation function.

Here, $\epsilon(x,t) = (\epsilon_{\text{real}}, \epsilon_{\text{imag}})$ is a complex-valued function, where ϵ_{real} represents the real part, and ϵ_{imag} represents the imaginary part. For Eq. (1), a corresponding separation can be performed, transforming it into the following form:

$$\begin{cases} \dfrac{\partial \epsilon_{\text{real}}}{\partial t} - \Delta \epsilon_{\text{imag}} + n\epsilon_{\text{imag}} = 0, \\ \dfrac{\partial \epsilon_{\text{imag}}}{\partial t} + \Delta \epsilon_{\text{real}} - n\epsilon_{\text{real}} = 0, \\ \dfrac{1}{c_0}\dfrac{\partial^2 n}{\partial t^2} - \Delta n = \nabla^2 |\epsilon|^2, \end{cases} \tag{5}$$

where $|\epsilon|^2 = \epsilon_{\text{real}}^2 + \epsilon_{\text{imag}}^2$. Thus, the physical constraint information of the Zakharov equation can be defined as follows:

$$\begin{cases} f_{\text{real}} = \dfrac{\partial \epsilon_{\text{real}}}{\partial t} - \Delta \epsilon_{\text{imag}} + n\epsilon_{\text{imag}}, \\ f_{\text{imag}} = \dfrac{\partial \epsilon_{\text{imag}}}{\partial t} + \Delta \epsilon_{\text{real}} - n\epsilon_{\text{real}}, \\ g = \dfrac{1}{c_0}\dfrac{\partial^2 n}{\partial t^2} - \Delta n - \nabla^2 |\epsilon|^2. \end{cases} \tag{6}$$

To simulate the dynamical behavior of soliton solutions in the Zakharov equation, the neural network needs to be trained to minimize the loss function to better satisfy the initial and boundary conditions, as well as the physical constraints. Specifically, the loss function can be defined as

$$\text{loss} = \text{loss}_{\text{ic}} + \text{loss}_{\text{bc}} + \text{loss}_{\text{f}}, \tag{7}$$

with

$$\begin{aligned} \text{loss}_{\text{ic}} = \frac{1}{N_0}\sum_{i=1}^{N_0} \Big[& \left(\epsilon_{\text{real}}^{(i)}(x_0^i,t_0) - \epsilon_{\text{real},0}^i\right)^2 \\ & + \left(\epsilon_{\text{imag}}^{(i)}(x_0^i,t_0) - \epsilon_{\text{imag},0}^i\right)^2 \\ & + \left(n^{(i)}(x_0^i,t_0) - n_0^i\right)^2 \Big], \end{aligned} \tag{8}$$

$$\begin{aligned} \text{loss}_{\text{bc}} = \frac{1}{N_{\text{b}}}\sum_{i=1}^{N_{\text{b}}} \Big[& \left(\epsilon_{\text{real}}^{(i)}(x_L^i,t_i)\right)^2 \\ & + \left(\epsilon_{\text{imag}}^{(i)}(x_L^i,t_i)\right)^2 + \left(n^{(i)}(x_L^i,t_i)\right)^2 \Big], \end{aligned} \tag{9}$$

$$\begin{aligned} \text{loss}_{\text{f}} = \frac{1}{N_{\text{f}}}\sum_{i=1}^{N_{\text{f}}} \Big[& \left(f_{\text{real}}^{(i)}(x_{\text{f}}^i,t_{\text{f}})\right)^2 \\ & + \left(f_{\text{imag}}^{(i)}(x_{\text{f}}^i,t_{\text{f}})\right)^2 + \left(g^{(i)}(x_{\text{f}}^i,t_{\text{f}})\right)^2 \Big], \end{aligned} \tag{10}$$

where N_0, N_{b}, and N_{f} represent the numbers of initial points, boundary points, and collocation points, respectively. The terms $\left\{\left(x_0^i, t_0^i, \epsilon_{\text{real},0}^i, \epsilon_{\text{imag},0}^i, n_0^i\right)\right\}_{i=1}^{N_0}$ represents the initial data. The term $\left\{(\pm x_L, t_{\text{b}}^i)\right\}_{i=1}^{N_{\text{b}}}$ represents the boundary data, which can be randomly sampled. The term $\left\{(x_{\text{f}}^i, t_{\text{f}}^i)\right\}_{i=1}^{N_{\text{f}}}$ represents the collocation points. These points are sampled from the computational domain, often using Latin Hypercube Sampling (LHS). Equations (8) and (9) define the losses for the initial and boundary data. Equation (10) penalizes the collocation points to satisfy the Zakharov equation. The network utilizes the residual loss to penalize deviations from the governing physical equations. This process ensures the solution remains both valid and accurate within the computational domain.

2.2. Improved PINN by conservation multi-stage algorithm

When simulating the solutions of the Zakharov equation, the original PINN method often faces issues with poor approximation accuracy and slow training speed. To address these challenges, we propose the CM algorithm to ensure that the model not only approximates the data well during training, but also satisfies the conservation laws of the physical system.

In the preset multi-stage training strategy, we first divide the entire domain into n subdomains Ω_k $(k = 1,2,\ldots,n)$, and perform staged training to ensure that computational resources are focused on the most critical regions in each stage, thereby improving training efficiency and approximation accuracy.

During the training process, at each stage k, a set of collocation points $\{x_j^i, t_j^i\}_{j=1}^{N_{\text{f}}}$ is selected from the entire domain based on the distance to the initial points $\{x_0^i, t_0^i\}_{i=1}^{N_0}$. Using the collocation points $\{x_j^i, t_j^i\}_{j=1}^{N_{\text{f}}}$ to generate n small subsets $\left\{S_k = \left\{(x_k^i, t_k^i)\right\}_{j=1}^{\frac{k}{n}N_{\text{f}}} | k = 1,\ldots,n\right\}$, such that

$$S_{k-1} \subseteq S_k, \quad k = 2,\ldots,n, \tag{11}$$

$$\frac{1}{\frac{k-1}{n}N_{\rm f}}\sum_{j=1}^{\frac{k-1}{n}N_{\rm f}}{\rm dis}(x_{\rm f}^j,t_{\rm f}^j)<\frac{1}{\frac{k}{n}N_{\rm f}}\sum_{j=1}^{\frac{k}{n}N_{\rm f}}{\rm dis}(x_{\rm f}^j,t_{\rm f}^j),\quad k=2,\ldots,n, \tag{12}$$

where

$${\rm dis}(x_{\rm f}^j,t_{\rm f}^j)=\min_{i=1,\ldots,N_0}\|(x_{\rm f}^j,t_{\rm f}^j)-(x_0^i,t_0^i)\|_2^2,\quad j=1,\ldots,N_{\rm f}. \tag{13}$$

Equations (11)–(13) mean that the sample point set at each stage contains all sample points from the previous stage. Additionally, the sampling region gradually expands. This approach enables the model to progressively approximate the solution across the entire domain.

We add a conservation loss term $\text{loss}_{\rm c}$

$$\text{loss}_{\rm c}=\frac{1}{N_{\rm c}}\sum_{j=1}^{N_{\rm c}}\left(\frac{\partial C(x_{\rm f}^j,t_{\rm f}^j)}{\partial t}+\nabla\cdot F(x_{\rm f}^j,t_{\rm f}^j)\right)^2, \tag{14}$$

to ensure the physical consistency of the model during training. $C(x_j^i,t_j^i)$ represents the conserved quantity at the collocation points $(x_{\rm f}^j,t_{\rm f}^j)$. $\frac{\partial C}{\partial t}$ denotes the time derivative of C, and $\nabla\cdot F$ represents the divergence of the flux F. $N_{\rm c}$ is the number of collocation points. $\text{loss}_{\rm c}$ balances the discrepancy between the conserved quantity at each collocation point and its theoretical value.

The complete diagram of the CM algorithm is shown in Figs. 1 and 2(a). For each k-th stage ($k=1,\ldots,n$), the PINN model solve the Zakharov equation within the subset S_k (like the blue points in Fig. 2(b)). Once the model's loss within the S_k drops below the threshold ϵ or exceeds the maximun iterations M, we select N_k collocation points (like the red points in Fig. 2(b)) from S_k to get the pre-fixed points $P_k=\{(x_k^j,t_k^j)\}_{j=1}^{N_k}$, which minimize the model's loss as Eq. (10). The number of fixed points per stage N_k, is determined by evenly distributing the total number of collocation points $N_{\rm f}$ across n training stages $N_k\approx N_{\rm f}/n$. The trained model subsequently use these pre-fixed points $\{(x_k^j,t_k^j)\}_{j=1}^{N_k}$ to predict $\{\epsilon_{{\rm real},k}^j,\epsilon_{{\rm imag},k}^j,n_k^j\}_{j=1}^{N_k}$, and add a new enhanced fixed term in the loss function as

$$\begin{aligned}\text{loss}_{\rm p}=\frac{1}{N_k}\sum_{i=1}^{N_k}\Big(&\left|\varepsilon_{\rm real}(x_i^k,t_i^k)-\varepsilon_{\rm real}\right|^2\\&+\left|\varepsilon_{\rm imag}(x_i^k,t_i^k)-\varepsilon_{\rm imag}\right|^2+\left|n(x_i^k,t_i^k)-n\right|^2\Big).\end{aligned} \tag{15}$$

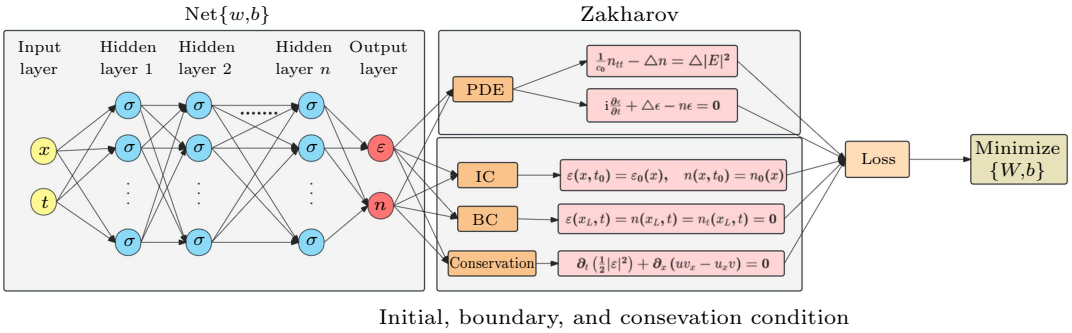


Fig. 1. The CM algorithm of Zakharov equation.

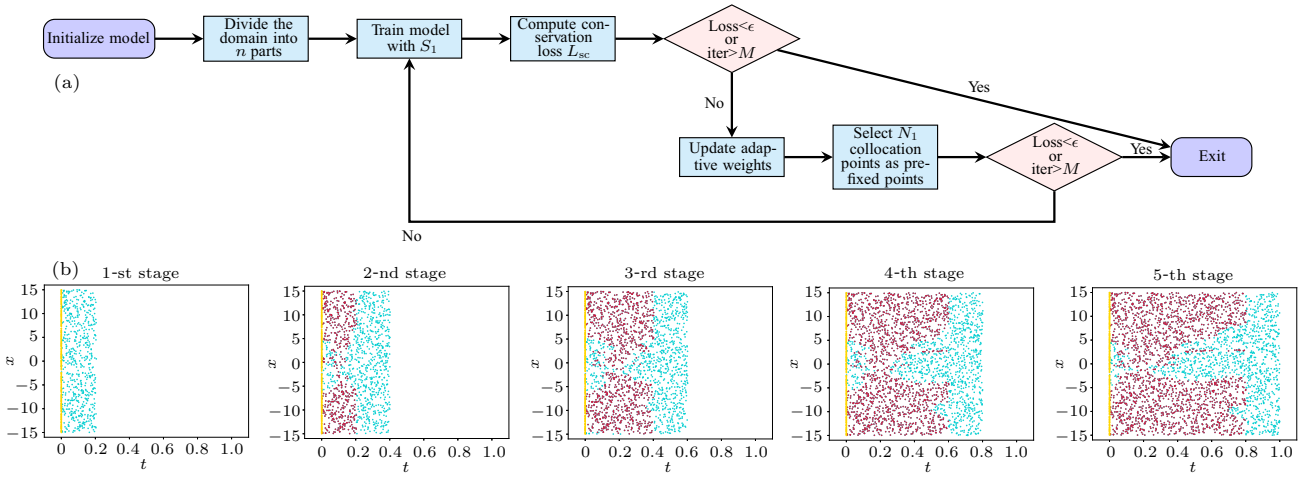


Fig. 2. Panel (a) presents the flow diagram of the CM algorithm. Panel (b) illustrates example distributions of collocation points and pre-fixed points across various training stages.

Moreover, to address the gradient propagation issues arising from the imbalance among different loss terms during training, we introduce an adaptive weighting strategy.[28] With this method, the overall loss function is re-defined as

$$\text{Loss}=\text{loss}_{\rm f}+\sum_{i=1}^{4}\lambda_i\text{loss}_i, \tag{16}$$

where λ_1, λ_2, λ_3, λ_4 are the adaptive weights, $\text{loss}_1 = \text{loss}_{\text{ic}}$, $\text{loss}_2 = \text{loss}_{\text{bc}}$, $\text{loss}_3 = \text{loss}_{\text{p}}$, $\text{loss}_4 = \text{loss}_{\text{c}}$. The Loss not only helps the model to accurately fit the data but also ensures compliance with physical conservation laws. In addition, the adaptive weighting mechanism dynamically adjusts during training to address the gradient imbalances between different loss terms, thus optimizing the model's training efficiency and predictive accuracy. The detailed process is presented in Algorithm 1.

(i) Efficiency analysis By reducing the number of neural network layers, the CM algorithm significantly shortens training time while maintaining good learning capabilities. Combined with the pre-fixed multi-stage strategy, the algorithm focuses computational resources on key regions, avoiding unnecessary computation time and maintaining high training efficiency even in large-scale domains. Although efficiency may slightly decrease as the computational domain expands, overall efficiency remains high. Detailed data can be found in Section 3, Table 2, Figs. 3 and 4.

(ii) Accuracy analysis The CM algorithm incorporates conservation laws as physical constraints, which speeds up initial convergence and ensures the solution remains physically consistent during training. It also improves accuracy by adjusting the weights of the loss function terms, prioritizing the components that most influence the solution. For more details, refer to Section 3, Table 2, Figs. 3 and 4.

Algorithm 1: Conservation multi-stage algorithm

Initialize: Initialize a PINN model $\phi_\theta(x,t)$ with parameters θ and the weights of loss terms λ_i for $i = 1,2,3,4$.
Input: Collocation points $\{(x_j^{\text{f}}, t_j^{\text{f}})\}_{j=1}^{N_{\text{f}}}$, divided into subsets $S_1, \ldots, S_n$. Pre-fixed points $E_1 = \emptyset$.
for $k = 1$ **to** n **do**
  for iteration = 1 **to** M **do**
    Compute loss_{f} by Eq. (10) with S_k.
    Compute loss_{ic}, loss_{bc} by Eqs. (8) and (9) with initial and boundary data.
    Compute conservation loss loss_{c} by Eq. (14).
    Compute loss_{p} by Eq. (15)
    Compute adaptive weights $\hat{\lambda}_i$ by

$$\hat{\lambda}_i = \frac{\max_\theta(|\nabla_\theta \text{loss}_{\text{f}}|)}{\overline{|\nabla_\theta \text{loss}_i|}}, \quad i = 1,2,3,4$$

    where $\overline{|\nabla_\theta \text{loss}|}$ denotes the mean of $|\nabla_\theta \text{loss}|$ with respect to parameters θ.
    Update the weights λ_i using a moving average of the form

$$\lambda_i = (1-\alpha)\lambda_i + \alpha\hat{\lambda}_i.$$

    Compute total loss by Eq. (16).
    Update the model parameters θ via optimizer.
    if loss $< \mu$ **then**
      break
    end if
  end for
  Select N_k pre-fixed points E_{k+1} from S_k which contribute the smallest loss_{f} computed by Eq. (10).
end for
Recommended hyper-parameter values: $\alpha = 0.9$, $N_k = 0.8 \cdot \frac{k}{n} \cdot N_{\text{f}}$.

3. Numerical simulation of Zakharov equation

This section presents the numerical simulation results of Zakharov equation, evaluates the performance of the CM algorithm, and compares it with A1, A2, and A3 listed in Table 1. Specifically, we focus on the stability and consistency of the solutions under different initial conditions. Furthermore, this analysis lays the foundation for the upcoming numerical simulations of the NLS equation and the exploration of the limit behavior between the two equations. In subsequent sections, we will examine how the solutions of the Zakharov equation asymptotically approximate those of the NLS equation as $c_0 \to \infty$.

Table 1. Algorithms considered in the numerical studies presented in Section 3.

Algorithm	Description
A1	Original PINN: Original formulation of physics-informed neural networks.
A2	Conservation law: Incorporate conservation laws into the loss function
A3	Pre-fixed multi-stage algorithm: Multi-stage adaptive training.
A4	Conservation multi-stage algorithm (Subsection 2.2): Combine the strategies of A2 and A3.

The Zakharov equation is complex and sensitive to initial conditions, so maintaining stability is important. To test the reliability of the solutions and the algorithm, we performed several experiments with different initial parameters. The model with the lowest loss value in each experiment was chosen as the final solution. We assessed the algorithm's overall performance by averaging the results of all experiments. The hyperbolic tangent (tanh) function was used as the activation func-

tion in the simulations. Python 3.7 and PyTorch 1.8.1 were used to maintain consistency in the experiments. The code is available at https://github.com/DSJ414/CM-algorithm.

The soliton solution of Eq. (3) is given in the form

$$\epsilon = A\,\mathrm{sech}\left(B(x-ct-x_0)\right)\mathrm{e}^{\mathrm{i}\left(\frac{c}{2}x+Dt+\theta\right)}, \tag{17}$$

$$n = -A^2\,\mathrm{sech}^2\left(B(x-ct-x_0)\right), \tag{18}$$

where the parameters are defined as

$$A = m/\sqrt{2}\left(1-c^2/c_0^2\right),$$
$$B = m/\left(2\left(1-c^2/c_0^2\right)\right),$$
$$D = c^2/4 - m^2/\left(4\left(1-c^2/c_0^2\right)^2\right).$$

Here, x_0 represents the initial position of the wave packet at $t=0$, and θ denotes the initial phase. The primary control parameters are m and c, where c represents the propagation speed of the soliton. From conditions (17) and (18), we obtain the limiting condition $c^2/c_0^2 < 1$. This implies that the speed of the soliton must be less than c_0, indicating that the soliton is subsonic. The term A^2 represents the depth of the density depression, and $1/B$ represents the width of the depression. The depth and width of the depression are inversely proportional, meaning that a deeper depression corresponds to a narrower width. Furthermore, the local mass conservation law provides an essential constraint for the dynamics of the soliton. It is given by

$$\partial_t\left(\frac{1}{2}|\epsilon|^2\right) + \partial_x\left[\epsilon_{\mathrm{real}}\frac{\partial\epsilon_{\mathrm{imag}}}{\partial x} - \epsilon_{\mathrm{imag}}\frac{\partial\epsilon_{\mathrm{real}}}{\partial x}\right] = 0, \tag{19}$$

where ϵ_{real} and ϵ_{imag} denote the real and imaginary parts of ϵ, respectively. This equation ensures that the total mass of the system is conserved during the propagation of the soliton, reflecting the physical consistency of the model.[43]

In practice, physical experiments involving high-cost and complex setups make numerical simulations a feasible alternative to expensive physical experiments. In this study, we attempt to adopt the CM algorithm to simulate experimental values and model Langmuir wave solutions.

In this study, we use a fully connected neural network with 2 hidden layers, each containing 30 neurons, to perform numerical simulations. The network parameters $\theta = (w, b)$ are initialized with Xavier initialization for w and zeros for b. The training dataset consists of $N_0 = 300$ initial points, $N_b = 300$ boundary points, and $N_f = 3000$ collocation points. The initial conditions can be obtained from Eqs. (17) and (18) at $t=0$, and the boundary conditions are zero at $x = \pm 15$. To improve training efficiency and model accuracy, we divide the computational domain into 5 parts, meaning that the CM algorithm consists of 5 training stages. At each stage, we select 80% of the collocation points from the previous stage's collocation point set. The number of fixed points at each stage is defined as $N_j = 0.8N_f/5 = 480$, $j = 1,\ldots,5$, increasing as the stage progresses. This progressive training approach allows the model to gradually adapt to the characteristics of the entire computational domain. At each training stage, in addition to the standard loss terms (initial condition loss, boundary condition loss, and collocation point loss), we also incorporate the conservation loss term $\mathrm{loss_c}$ to ensure that the model outputs satisfy the corresponding conservation laws.

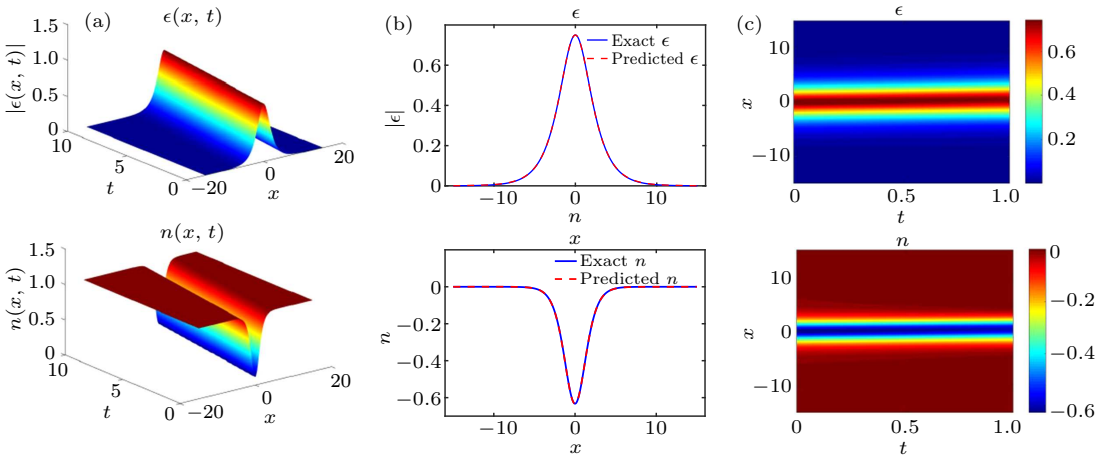


Fig. 3. Solutions of the Zakharov equation using the CM algorithm with parameters $m=1$, $c=1/2$, $x_0=0$, $\theta=0$. Panel (a) presents the three-dimensional (3D) evolution plots for variables ϵ and n over time t and spatial position x, illustrating the dynamic evolution and spatial distribution respectively. Panel (b) shows the comparison of predicted and exact solutions for ϵ and n at $t=0$, where the exact solutions are depicted in blue lines and the predictions in dashed red lines. Panel (c) displays the density maps for ϵ and n, with the color bars indicating the range of values for each variable.

We set a minimum loss threshold of $\mu = 1\times10^{-6}$ for each training phase and defined the maximum number of iterations per phase as $M_2 = 2000$. This ensures the training stops once the accuracy requirement is met or the iteration limit is reached. We also included the conservation constraint term in the loss function, optimizing it with other loss terms to align the training with physical principles. The CM algorithm captures the dynamic behavior of the solution to Eq. (3), as shown

in Figs. 3 and 4.

In the experiments, we set the parameters $m=1$, $c=1/2$, $x_0=0$, and $\theta=0$. Figure 3 shows the simulation results of the CM algorithm. In panel (a), 3D images of $\epsilon(x,t)$ and $n(x,t)$ are displayed. $\epsilon(x,t)$ evolves with energy concentration in the central region, while $n(x,t)$ shows wave amplitude attenuation in the same area. Panel (b) compares the predicted solutions of ϵ and n with the actual solutions at $t=0$. The predicted solutions (in red) are close to the actual solutions (in blue), showing that the CM algorithm effectively captures the solution's behavior. Panel (c) presents density plots of ϵ and n, showing their wave dynamics over time, especially in the central region. Figure 4 compares the absolute errors between the original PINNs and the CM algorithm. The CM algorithm has better accuracy, with the original model showing larger errors in the central region, which grow over time. In contrast, the CM algorithm reduces these errors, showing clear improvement in dynamic conditions. Tables 2 and 3 show the relative errors, computation time, and relative errors of conserved quantities for the four algorithms under the same conditions.

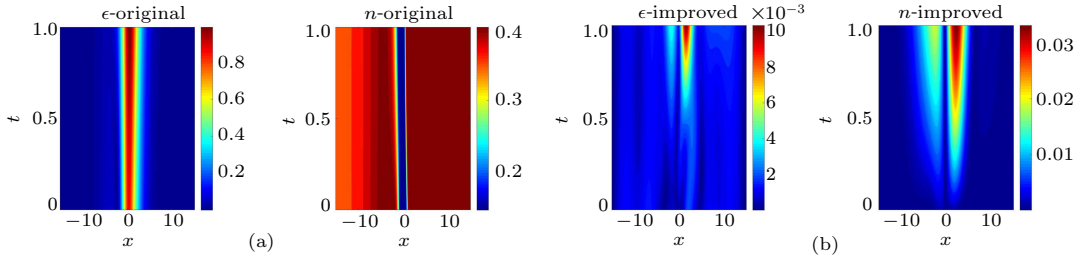


Fig. 4. (a) The errors of the two variables ϵ and n trained by the original PINN method for solving the Zakharov equation, where the error of ϵ is calculated as the sum of the errors $\|\epsilon_{\text{real}}\|+\|\epsilon_{\text{imag}}\|$. The error of ϵ is shown on the left, and the error of n is shown on the right. (b) The errors of the two variables ϵ and n trained by the CM algorithm for solving the Zakharov equation, where the error of ϵ is calculated as the sum of the errors $\|\epsilon_{\text{real}}\|+\|\epsilon_{\text{imag}}\|$. The error of ϵ is shown on the left, and the error of n is shown on the right.

Table 2. The comparisons of different algorithms (error metrics).

Algorithm	ϵ_{real} (relative L_2 error)	ϵ_{imag} (relative L_2 error)	n (relative L_2 error)
A1	1.4282	1.9556	1.7092
A2	1.6176×10^{-1}	2.7202	1.3253
A3	7.0024×10^{-3}	3.6723×10^{-2}	8.5469×10^{-2}
A4	3.3723×10^{-3}	1.4005×10^{-2}	3.1218×10^{-2}

For model parameter optimization, we used the L-BFGS optimizer.[44] When adding 0.5% white noise to the initial data and boundary conditions, the results closely resemble those in Fig. 3. This confirms that the high-frequency–low-frequency coupled soliton remains stable under small perturbations. It also shows that such coupled solitons can be observed in real physical systems, extending beyond specific exact solutions.

Table 3. The comparisons of different algorithms (time and conservation error).

Algorithm	Time (s)	Conservation error
A1	1014.32	–
A2	1014.32	3.3241×10^{-5}
A3	743.49	–
A4	321.38	2.4400×10^{-7}

4. Limit behavior from Zakharov equation to NLS equation

In this section, we first explore the NLS equation through numerical simulations to verify the practical performance of our method in handling typical nonlinear wave problems. Subsequently, we focus on the limit behavior from Zakharov equation to NLS equation as $c_0\to\infty$. This part of study not only demonstrates the theoretical applicability of the CM algorithm but also provides a practical numerical tool for analyzing wave behaviors in complex nonlinear systems.

4.1. Numerical simulation of the solution to the NLS equation

Among the various applications of quantum mechanics, the NLS equation plays a particularly important role in explaining and predicting soliton wave phenomena.[45–49] Soliton solutions are a special class of solutions to this wave equation, demonstrating how stable localized wave packets can form in a nonlinear medium.[47] Our study investigates the following NLS equation:

$$\mathrm{i}\,\epsilon_t+\epsilon_{xx}+|\epsilon|^2\epsilon=0, \tag{20}$$

where one of the solutions of Eq. (20) is as follows:

$$\begin{aligned}\epsilon \;=\; & \frac{m}{\sqrt{2}}\operatorname{sech}\left(\frac{m}{2}(x-ct-x_0)\right)\\ & \times\exp\left[\mathrm{i}\left(\frac{c}{2}x+\left(\frac{c^2}{4}-\frac{m^2}{4}\right)t+\theta\right)\right].\end{aligned} \tag{21}$$

The NLS equation also satisfies the mass conservation law as the Zakharov equation, see Eq. (19).

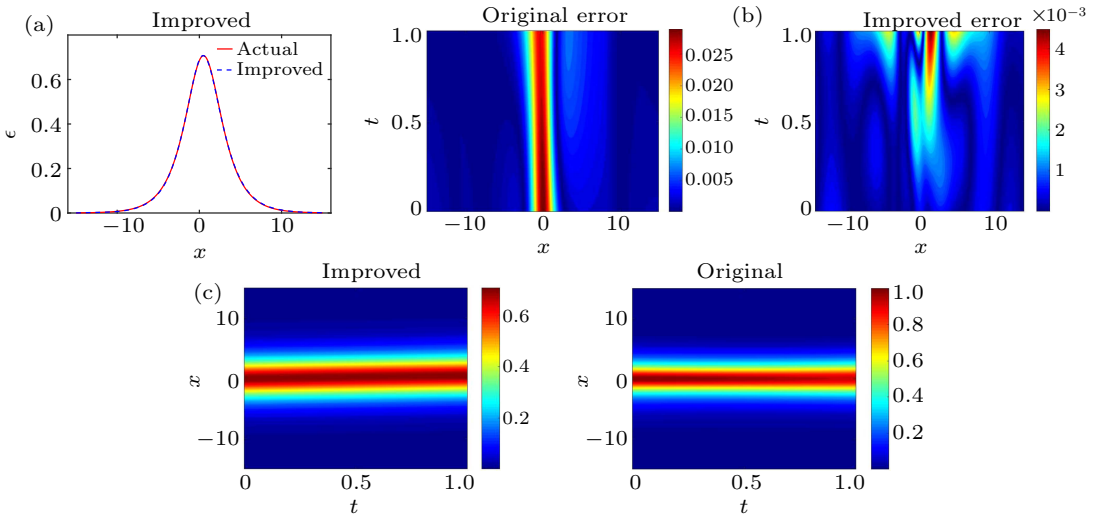


Fig. 5. Comparisons of solutions for NLS equation using both original PINNs method and the CM algorithm. (a) Intensity profiles at $t = 0$ showing the actual soliton and its approximation by the CM algorithm. (b) Error distribution maps for the original PINNs method and the CM algorithm, illustrating the regions of high and low errors in simulating the NLS equation. (c) Evolution over time, displayed as density plots with x on the horizontal axis and t on the vertical; color intensity represents the magnitude of ϵ, with separate plots for the CM algorithm and original PINNs method.

In the experiments, we set the following parameters as $m = 1$, $c = 1/2$, $x_0 = 0$, and $\theta = 0$. To ensure fair comparisons, both of the CM algorithm and traditional PINNs were implemented with an identical simplified network (2 hidden layers, 30 neurons per layer). The results are shown in the following Fig. 5.

Figure 5(a) shows the intensity profile at $t = 0$, comparing the actual soliton solution with the approximation from the CM algorithm. The CM algorithm closely approximates the actual solution, particularly in the central region, where the predicted peak value is nearly identical to the actual value. Figure 5(b) presents the error distribution maps for both the original PINNs model and the CM algorithm. In the central region of the soliton, the CM algorithm shows a very small error, almost negligible, while the original model has larger errors in the same region. As the soliton propagates, the errors in the original model grow. The discrepancy arises because the CM algorithm accurately captures soliton gradient features even in a simplified network with 2 hidden layers. In contrast, traditional PINNs rely on deep architectures. They need stacked layers to fit steep soliton profiles. Shallow network architectures limit their ability to fit gradients in such networks. Figure 5(c) shows how the solution evolves over time in density plots for both models. The CM algorithm captures the propagation features of the soliton more effectively, especially in the central region, where the soliton's energy remains concentrated and stable.

Overall, the CM algorithm shows clear advantages in the numerical simulation of solitons, especially in predicting the initial wave packet and accurately modeling its evolution over time. Compared to the original PINNs method, the CM algorithm achieves higher accuracy in high-energy regions. By using a multi-stage training strategy and conservation constraints, the CM algorithm maintains good accuracy even with fewer training samples. The original PINNs method, on the other hand, has limitations under the same training settings, especially regarding error accumulation over time, which affects the overall representation of soliton behavior. Therefore, the CM algorithm is more reliable and robust when simulating complex systems like solitons.

4.2. Numerical simulation from Zakharov equation to NLS equation

When the speed of sound $c_0 \to \infty$ in Eq. (1), the influence of the low-frequency sound waves diminishes, and the system dynamics simplify to

$$n = -|\epsilon|^2. \tag{22}$$

Substituting this relation into the Zakharov equation (1), the NLS equation is derived

$$\mathrm{i}\epsilon_t + \epsilon_{xx} + |\epsilon|^2\epsilon = 0. \tag{23}$$

This approximation indicates that the Zakharov equation serves as a generalized form of the NLS equation, while the NLS equation accurately captures the high-frequency behavior of the system in the high sound-speed limit. Specific theoretical proofs can be found in Refs. [38–42].

In previous numerical experiments, we verified that the CM algorithm can accurately capture the solutions of both equations, with predictions closely matching their exact solutions. In the following, we analyze the variation trends of the solutions by comparing the predicted solutions of the Zakharov equation and the NLS equation as c_0 increases, further

revealing the intrinsic connection and unification between the two equations in the high sound speed limit.

Figure 6 illustrates the process by which the solution of Zakharov equation gradually approximates the solution of NLS equation as c_0 increases. Figure 6(a) compares the predicted solutions of Zakharov equation for different values of c_0 with the solution of NLS equation. When $c_0 = 1$, the solution of Zakharov equation shows a noticeable deviation from that of NLS equation. However, as c_0 increases (*e.g.*, $c_0 = 10$), the solutions of the two equations converge and eventually almost completely overlap. Figure 6(b) shows the distribution of the differences between the predicted solutions of Zakharov equation and NLS equation as c_0 increases. It can be observed that as c_0 increases, the difference gradually decreases, and at $c_0 = 10$, the difference approximates to zero, indicating that the solutions of the two equations are numerically almost identical.

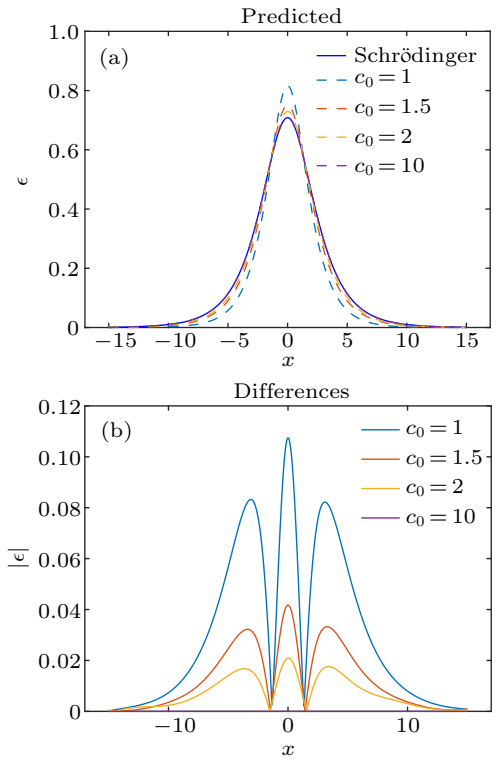


Fig. 6. Comparisons of numerical simulations between the Zakharov equation and the NLS equation. (a) Comparisons of the numerical simulation results of Zakharov equation and NLS equation under different parameter values of c_0. The blue solid line represents the numerical solution of NLS equation, and the dashed lines in different colors represent the numerical solutions of Zakharov equation under different c_0 values. (b) Absolute error of the numerical solutions of Zakharov equation and NLS equation under different c_0 values.

These results validate the theoretical properties of the Zakharov equation under limiting conditions. They also quantify, through difference analysis, the dynamic process by which its solution approximates that of the NLS equation. This analysis shows the intrinsic connections between the two equations and highlights the efficiency and applicability of the CM algorithm in solving complex nonlinear wave problems.

5. Conclusion

This study proposes a conservation multi-stage (CM) algorithm, which incorporates conservation laws[33,34] into the multi-stage training process.[29] Using this algorithm, we numerically simulated the Zakharov equation and the NLS equation, verifying that as c_0 becomes sufficiently large, the solution of Zakharov equation gradually approximates that of NLS equation. This study reveals the intrinsic connection between these two equations and demonstrates the effectiveness of the CM algorithm in solving complex nonlinear problems. Finally, by comparing with methods A1, A2, and A3, the advantages of the CM algorithm are highlighted.

Although our focus is on the soliton solutions of the Zakharov equation, the proposed algorithm can also be extended to solve other types of solutions and complex coupled systems. Extending the CM algorithm to other nonlinear equations, exploring its computational efficiency in high-dimensional systems, and optimizing related strategies will be the focus of our future research.

Acknowledgement

Lijia Han is supported in part by the National Natural Science Foundation of China (Grant No. 11971166).

References

[1] Chen Z, Segev M and Christodoulides D N 2012 *Rep. Prog. Phys.* **75** 086401
[2] Zakharov V E 1972 *J. Exp. Theor. Phys.* **35** 1086
[3] Ablowitz M A and Clarkson P A 1991 *Solitons, Nonlinear Evolution Equations and Inverse Scattering* (Cambridge: Cambridge Univ. Press)
[4] Hirota R 1971 *Phys. Rev. Lett.* **27** 1192
[5] Gu C H, Zhou Z H and Tan H 1990 *Soliton Theory and Its Application* (Hangzhou: Zhejiang Science and Technology Press)
[6] Ern A, Guermond JL 2004 *Theory and Practice of Finite Elements* (New York: Springer) *Appl. Math. Sci.* **159**
[7] Eymard R, Gallouët T and Herbin R 2000 *Handb. Numer. Anal.* **7** 713
[8] Trefethen L N 2000 *Spectral Methods in MATLAB* (Philadelphia: Society for Industrial and Applied Mathematics)
[9] Canale R P and Chapra S C 2014 *Numerical Methods for Engineers* (Europe: McGraw-Hill Education)
[10] Washington W M, Buja L and Craig A 2008 *Philos. Trans. R. Soc. A* **366** 2157
[11] Raissi M, Perdikaris P and Karniadakis G E 2019 *J. Comput. Phys.* **378** 686
[12] Karniadakis G E, Kevrekidis I G, Lu L, *et al.* 2021 *Nat. Rev. Phys.* **3** 422
[13] Liu Y, Liu W, Yan X, Guo S and Zhang C A 2023 *J. Comput. Phys.* **490** 112291
[14] Lu L, Meng X, Mao Z and Karniadakis G E 2021 *SIAM Rev.* **63** 208
[15] Cuomo S, Di Cola V S, Giampaolo F, Rozza G, Raissi M and Piccialli F 2022 *Phys. Scr.* **92** 88
[16] Nakamura M, He H and Abdikian H 2024 *Comput. Phys. Commun.* **285** 108460

[17] Song J, Zhong M, Karniadakis G E and Yan Z 2024 *J. Comput. Phys.* **505** 112917
[18] Peng Y, Pu J C and Chen Y 2022 *Commun. Nonlinear Sci. Numer. Simul.* **105** 106067
[19] Lin S and Chen Y 2023 *Physica D* **445** 133629
[20] Pu J C and Chen Y 2022 *Chaos, Solitons and Fractals* **160** 112182
[21] Pu J and Chen Y 2023 *Physica D* **454** 133851
[22] Zhou Z and Yan Z 2021 *Phys. Lett. A* **387** 127010
[23] Wu Y, Ling L and Huang Y 2024 *Comput. Math. Appl.* **169** 132
[24] Yan Z 2007 *Phys. Lett. A* **361** 194
[25] De Ryck T and Mishra S 2022 *Adv. Comput. Math.* **48** 79
[26] Jagtap A D and Karniadakis G E 2020 *Commun. Comput. Phys.* **28** 2002
[27] Wight C L and Zhao J 2021 *Commun. Comput. Phys.* **29** 930
[28] Wang S, Teng Y and Perdikaris P 2021 *SIAM J. Sci. Comput.* **43** A3055
[29] Mo Y, Ling L and Zeng D 2022 *Phys. Lett. A* **421** 127739
[30] Jagtap A D, Kharazmi E and Karniadakis G E 2020 *Comput. Methods Appl. Mech. Eng.* **365** 113028
[31] Gurieva J, Vasiliev E and Smirnov L 2022 *Procedia Comput. Sci.* **212** 464
[32] Baez A, Zhang W, Ma Z, Das S, Nguyen L M and Daniel L 2024 arXiv: 2410.17445
[33] Wu G Z, Fang Y, Kudryashov N A, Wang Y Y and Dai C Q 2022 *Chaos Solitons Fractals* **159** 112143
[34] Fang Y, Wu G Z, Kudryashov N A, Wang Y Y and Dai C Q 2022 *Chaos, Solitons and Fractals* **158** 112118
[35] Lin S and Chen Y 2022 *J. Comput. Phys.* **457** 111053
[36] Nakamura A, Sawado N, Shimasaki K, Suzuki Y and Toda K 2024 arXiv: 2410.19014
[37] Abdikian A 2024 *Phys. Plasmas* **31** 102105
[38] Han L, Zhang J, Gan Z and Guo B 2012 *Sci. China Math.* **55** 509
[39] Schochet S H and Weinstein M I 1986 *Commun. Math. Phys.* **106** 569
[40] Added H and Added S 1988 *J. Funct. Anal.* **79** 183
[41] Ozawa T, Tsutsumi Y and Brezis H 1992 *Differ. Integral Equ.* **5** 721
[42] Masmoudi N and Nakanishi K 2005 *J. Hyperbolic Differ. Equ.* **2** 975
[43] He Y, Wang J L and Huang J 2023 *Journal on Numerical Methods and Computer Applications* **44** 1
[44] Liu D C and Nocedal J 1989 *Math. Program.* **45** 503
[45] Ling L and Zhao L C 2013 *Phys. Rev. E* **88** 043201
[46] Zhao L C, Guo B and Ling L 2016 *J. Math. Phys.* **57** 043508
[47] Guo B and Huo Z 2010 *Commun. Partial Differ. Equ.* **36** 247
[48] Ling L, Zhao L C and Guo B 2015 *Nonlinearity* **28** 3243
[49] Pei Y T, Wang J K, Guo B L and Liu W M 2023 *Acta Phys. Sin.* **72** 100201 (in Chinese)