ISSN 0027-1349, Moscow University Physics Bulletin, 2024, Vol. 79, Suppl. 2, pp. S666–S675. © Allerton Press, Inc., 2024.

Neural Network Modeling of Optical Solitons Described by Generalized Nonlinear Schrödinger Equations

S. V. Zavertyaev[1], I. A. Moloshnikov[1,2], A. G. Sboev[1,2*], and M. S. Kuvakin[2,3]

[1]*National Research Nuclear University MEPhI, Moscow, Russia*
[2]*National Research Centre "Kurchatov Institute," Moscow, Russia*
[3]*Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Moscow oblast, Russia*

Received October 1, 2024; revised October 10, 2024; accepted October 20, 2024

Abstract—The paper examines the modeling of pulse propagation in a nonlinear medium using two partial differential equations, namely the second-order Schrödinger equation and the fourth-order generalized nonlinear Schrödinger equation (GNSE). The applicability of physics-informed neural networks (PINN) methods for solving the GNSE is demonstrated for analyzing physical effects involving solitons, using the example of soliton interaction with an isolated wave. A study of the efficiency of balancing methods for the GNSE is conducted on a boundary value problem with zero boundary conditions, as well as an assessment of the accuracy of the PINN method with segmentation by comparing it to the exact solution for single solitons of the second and fourth-order GNSE. The use of conservation laws as a means of verifying the validity of the solution is experimentally justified, in the absence of the possibility of comparing it to an exact solution, thereby suggesting their use as an additional validation metric for the obtained solutions.

Keywords: PINN, nonlinear Schrödinger equation, NSE, optical solitons

DOI: 10.3103/S0027134924702114

1. INTRODUCTION

One of the current significant tasks in the study of mathematical models is the investigation of optical soliton propagation in a nonlinear medium. The importance of this task is related to the fact that optical solitons are a crucial component in information encoding for long-distance transmission. To describe pulse propagation in optical media, several mathematical models have been proposed and intensively studied in recent years. The primary equation used to describe optical solitons in a nonlinear medium is the nonlinear Schrödinger equation, which takes the form

$$i\,q_t + q_{xx} + |q|^2 q = 0. \tag{1}$$

The equation (1) is the simplest nonlinear equation used to describe optical solitons, accounting for the dispersion of the solitary wave and the Kerr nonlinearity of the medium's refractive index. The equation (1) belongs to the class of integrable equations, and its Cauchy problem can be solved using the inverse scattering transform method. This equation possesses a number of interesting and important properties characteristic of integrable equations: Lax pairs, Bäcklund transformations, an infinite number of conservation laws, multisoliton solutions, and some other properties. However, the propagation of pulses in a nonlinear medium is generally described by more complex nonlinear equations than Eq. (1), which take into account more complex dispersion laws and a more intricate refractive index coefficient. Considering these additional factors leads to more complicated nonlinear differential equations, which significantly complicate the study of pulse propagation processes in optical media and necessitate the development of new analytical and numerical methods for investigating the propagation of optical pulses in a nonlinear medium.

In recent years, the physics informed neural networks (PINN) method has gained popularity for modeling processes described by nonlinear partial differential equations, initially proposed in the work by [1]. This approach is attractive for several reasons:

- Neural networks are universal function approximators [2], making them suitable tools for

*E-mail: sag111@mail.ru

solving boundary value problems of nonlinear differential equations.

- In most machine learning methods, the more data available, the better the results. The creators of PINN leveraged the specificity of the subject area, namely the inexhaustible number of points from the domain defined by the boundary value problems.

- The method is highly flexible. Additional knowledge about the solution can potentially be incorporated into the loss function, and the architecture/activation functions can be tailored to the problem.

- The method is mesh-free and does not require interpolation.

- Neural networks can be trained in parallel, which significantly reduces computation time [3].

However, further application of the method has shown that for solving problems described by the Poisson, Schrödinger (1), and Allen–Cahn equations, the classic PINN approach still falls short of the finite element method in terms of speed (including training time) and accuracy [4].

It appears that this circumstance has led to the proposal of numerous modifications to the original PINN concept following its inception by [1]. Research is being conducted on the influence of activation functions and point generation methods on convergence. Loss functions considering the principle of causality (BC-PINN [5], CausalPINN [6]), as well as alternative architectures allowing for greater accuracy in less time (SPINN [7], Spline-PINN [8]), have been developed. For instance, in the article on SPINN, the authors proposed obtaining solutions as a weighted sum of kernel functions with specified centers. For solving the problem described by the Allen–Cahn equation, the method demonstrated comparable accuracy to the PySPH framework for fluid dynamics simulation, with similar computation times.

There is a promising approach in which the boundary value problem is considered a special case of the problem of finding operators that map one function to another (e.g., the initial condition $u_0(x)$ to the solution of the equation $u(x,t)$). This approach was proposed in the article DeepONet [9].

In the articles [10–12], including the original PINN paper [1, 13], it is shown that the standard PINN without modifications yields good results for boundary value problems of the nonlinear Schrödinger equation (NLS) and the Korteweg–de Vries (KdV) equation. Since the goal of this work is to solve boundary value problems described by generalized NLS, we primarily use this closest approach to solve our problems.

The density of points affects the convergence of the PINN method [1]. The study of soliton solutions is complicated by the fact that the pulse shifts, and to model a large area over time, it is necessary to capture a large spatial region. Thus, the density of points decreases quadratically. To compensate for this effect, we use segmentation, similar to FBPINN [14].

In [15], various balancing methods are also discussed using the example of solving problems described by the Poisson, NLS, and Allen–Cahn equations. In Section 3 of the article, we investigate the effectiveness of this approach for boundary value problems of generalized NLS.

For functional optimization in this work, we use the Adam and LBFGS algorithms, similar to the original work [1].

In the works [16, 17], the integral property of conservation laws is used, and in [18, 19], the laws themselves are used (since the equation in conservative form improves convergence). For some equations, considering them in the loss function leads to increased accuracy. However, in our case, there was no definite effect [20]. For small models or a small number of points from the domain, using the integral property of conservation laws can reduce error. In this work, conservation laws are used only as a means of controlling the correctness of the solution.

In this article, we compare for the first time the effectiveness of applying the PINN method with its segmented modification, inspired by the methods proposed in the FBPINN [14] and BC-PINN [5] articles for nonintegrable generalized Schrödinger equations [21].

In this work, to model the propagation of pulses in a nonlinear medium, two partial differential equations are used, namely, a second-order equation [22]:

$$iq_t + q_{xx} + |q|^2 q \left(1 - \alpha|q|^2 + \beta * |q|^4\right) = 0, \quad (2)$$

which has a solution in the form of a bright soliton:

$$q_2(x,t) = \sqrt{\frac{\mu e^{(x-2kt-x_0)\sqrt{\mu}}}{\left(\frac{1}{2}e^{(x-2kt-x_0)\sqrt{\mu}} + 1\right)^2 - \frac{\alpha_0\mu}{3}e^{2(x-2kt-x_0)\sqrt{\mu}}}} \times e^{i(kx-\omega t+\theta_0)},$$

$$\mu = 4\left(k^2 - \omega\right). \quad (3)$$

And the generalized nonlinear fourth-order Schrödinger equation [23]:

$$iq_t + ia_1 q_x + a_2 q_{xx} + ia_3 q_{xxx} + a_4 q_{xxxx} = q\left(b_1|q|^2 + b_2|q|^4 + b_3|q|^6 + b_4|q|^8\right), \quad (4)$$

having an analytical solution under the following parameter restrictions:

$$a_3 = -4ka_4,$$
$$b_4 = \frac{105a_4}{A^4},$$
$$b_3 = \frac{120\mu a_4}{A^3},$$
$$b_2 = \frac{1}{A^2}\left(\left(18k^2 + 24\mu^2 + 78\nu\right)a_4 + 3a_2\right),$$
$$b_1 = \frac{1}{A}\left(\left(12\mu k^2 + 20\mu\nu + 60\chi\right)a_4 + 2\mu a_2\right),$$

where A, k, μ, ν, a_4, a_2—arbitrary, $\chi = 0$.

Bright soliton solution is as follows:

$$q_4(x,t) = \frac{\sqrt{2A\nu}e^{i(kx-tw-\theta_0)}}{\sqrt{-\mu + \sqrt{\mu^2 - 4\nu}\cosh\left(2\sqrt{\nu}\left(x - vt - x_0\right)\right)}},$$
$$v = 8k^3a_4 + 2ka_2 + a_1,$$
$$w = \left(3k^4 - 6\nu k^2 - 12\chi\mu - \nu^2\right)a_4, + \left(k^2 - \nu\right)a_2 + ka_1. \quad (5)$$

Equations (2) and (4) share the common property as shown in [21]: every periodic or decaying solution of these equations must satisfy conservation laws:

$$I_1 = \int_{-\infty}^{\infty} |q|^2 dx = \int_{-\infty}^{\infty}\left(u^2 + v^2\right)dx = \text{const},$$
$$I_2 = \int_{-\infty}^{\infty}\left(q^*q_x - q_x^*q\right)dx = \int_{-\infty}^{\infty}(uv_x - vu_x)dx = \text{const}.$$

Apparently, using conservation laws as additional terms in the loss function does not lead to increased accuracy [20]. However, they can serve as a means of verifying the validity of the solution when there is no possibility of comparing it with an exact solution.

That property would be utilised as an additional prediction validity metric in Section 3:

$$\begin{gathered} iq_t + q_{xx} + |q|^2q\left(1 - \alpha|q|^2 + \beta|q|^4\right) = 0, \\ x \in [x_L, x_R], t \in [0, t_{\max}], \\ q(x,0) = q_0(x), \\ q(x_L,t) = q(x_R,t), \\ q_x(x_L,t) = q_x(x_R,t). \end{gathered} \quad \text{(BVP2)}$$

In Section 2, our implementation of the PINN method is described. Subsection 3.1 demonstrates the effectiveness of segmentation. Subsection 3.2 evaluates the effectiveness of loss balancing strategies. Subsection 3.4 investigates the interaction of single-soliton solutions described by Eqs. (2) or (4) with solitary waves in the form of Gaussian functions, and the interaction of two analytical bright soliton solutions for the model described by Eq. (2).

In this work, machine learning and neural network modeling methods are applied to solve two boundary value problems.

In different experiments, various coefficients of the equation, domains (such as x_R, x_L, and $t_{\max,}$), and different initial conditions $q_0(x)$ are used:

$$\begin{gathered} iq_t + ia_1q_x + a_2q_{xx} + ia_3q_{xxx} \\ + a_4q_{xxxx} - \sum_{j=1}^{4} b_jq|q|^{2j} = 0, \\ x \in [x_L, x_R], t \in [0, t_{\max}], \\ q(x,0) = q_0(x), \\ q(x_L,t) = q(x_R,t), \\ q_x(x_L,t) = q_x(x_R,t), \\ q_{xx}(x_L,t) = q_{xx}(x_R,t), \\ q_{xxx}(x_L,t) = q_{xxx}(x_R,t). \end{gathered} \quad \text{(BVP4)}$$

In Subsection 3.3, the applicability of the PINN method to the considered second-order and fourth-order equations is demonstrated for the case of a single-soliton initial condition. The error with respect to the exact solution is provided.

In Subsection 3.4, the robustness of solutions (3) and (5) is examined when they encounter obstacles in the form of Gaussian functions and for the stationary case ($k = 0$) of solution (3).

2. PINN METHOD

The PINN method involves finding the weights of a fully connected neural network that takes the independent variables x and t as inputs and returns approximate values of the dependent variables (u and v). The error is minimized when these approximations are substituted into the corresponding boundary value problems BVP2 or BVP4:

$$\begin{gathered} \hat{q}(x,t) = \hat{u}(x,t) + i\hat{v}(x,t). \\ MSE_{ic} = ||\hat{q}(x,0) - q(x,0)||. \\ MSE_{bc} = ||\hat{q}(x_L,t) - \hat{q}(x_R,t)|| \\ + ||\hat{q}_x(x_L,t) - \hat{q}_x(x_R,t)||, \\ MSE_{eq} = ||Eq\left[\hat{q}(x,t)\right]||, \\ Loss = MSE_{ic} + MSE_{bc} + MSE_{eq}, \end{gathered}$$

where $\hat{u}(x,t)$ and $\hat{v}(x,t)$ are the neural network outputs.

$Eq\,[q(x,t)] = 0$—nonlinear differential operator from Eq. (2) or (4), all derivatives are calculated using tensorflow automatic differentiation. Here, $||\cdot|| = MSE(\cdot)$.

The algorithm of this method is schematically represented in Fig. 1.

As a result of the modification used in this work, the time domain is divided into segments, with a separate PINN being trained sequentially for each segment. The initial condition for the next segment is provided by the model trained on the previous segment.

When moving to the next segment, the weights are carried over from the previous segment and fine-tuned sequentially using Adam and LBFGS. Depending on the complexity of the problem, slightly different sets of hyperparameters were used in the models. The main differences included the number of segments, the number of neurons in each layer, the number of training iterations, and so on. Default parameters: number of segments is 50, 50k iterations of Adam + LBFGS for each segment, number of points from the previous segment is 20k, 32 neurons, 4 layers and sin activation function.

To evaluate the described algorithm, it is tested on a problem with a known exact solution. As initial conditions, slices of solutions (3) and (5) are used.

The following error metrics were utilized:

$$MSE_q = ||\hat{q}(x,t) - q(x,t)||,$$
$$Rel_h = \sqrt{\frac{\sum_{n=1}^{N} ||\hat{q}(x_n,t_n)| - |q(x_n,t_n)||^2}{\sum_{i=1}^{N} |q(x_i,t_i)|^2}},$$
$$Res = |Eq\,[\hat{q}(x,t)]\,|,$$
$$Lw1(t) = \frac{I_1 - \sum_{j=0}^{N_x} \left(\hat{u}(j\Delta x,t)^2 + \hat{v}(j\Delta x,t)^2\right)}{I_1} \times 100,$$
$$Lw2(t) = \frac{I_2 - \sum_{j=0}^{N_x} (\hat{u}(j\Delta x,t)\hat{v}_x(j\Delta x,t) - \hat{v}(j\Delta x,t)\hat{u}_x(j\Delta x,t))}{I_2} \times 100,$$
$$\Delta x = \frac{x_R - x_L}{N_x}, N_x = 1000.$$

3. COMPUTATIONAL EXPERIMENTS

3.1. Justification for Choosing the PINN Method with Segmentation

To assess the effectiveness of segmentation, an experiment was conducted on the boundary value problem BVP2 with a small domain of definition ($x_{\rm L} = -50$, $x_{\rm R} = 50$, $t_{\rm max} = 10$). Two models without segmentation (32 and 64 neurons) are compared with a model using 5 segments (32 neurons). The initial condition chosen is $q_0(x) = q_2(x, 0)$, $k = 1$, $\omega = 0.88$, $x_0 = -30$, $\theta = 0$, $\alpha_0 = 0.3$. The parameters of the equation are $\alpha = 0.3$ and $\beta = 0$. Thus, in this case, the initial condition is a slice of the exact solution.

The results in Table 1 show that the algorithm with segmentation yields the lowest error for the first law while maintaining comparable values for the other metrics and similar computation time.

3.2. Investigation of Balancing Methods

To determine the effectiveness of dynamic coefficient balancing methods, an experiment was conducted for the boundary value problem BVP2 with the following boundary conditions:

$$q(x_{\rm L}, t) = 0,$$
$$q(x_{\rm R}, t) = 0.$$

Dynamic coefficient balancing is one of the methods used to enhance the training process of neural networks. The idea is to ensure that the neural network learns to satisfy all imposed conditions equally well. To achieve this, coefficients are added to the terms of the loss function, resulting in the following form:

$$loss = \sum_i \lambda_i \times loss_i,$$
$$\sum_i \lambda_i \approx 1,$$

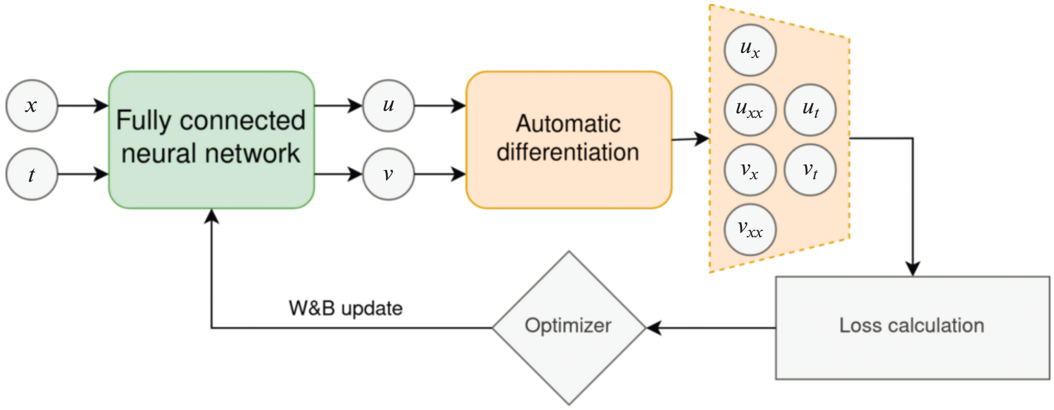

Fig. 1. Original PINN method.

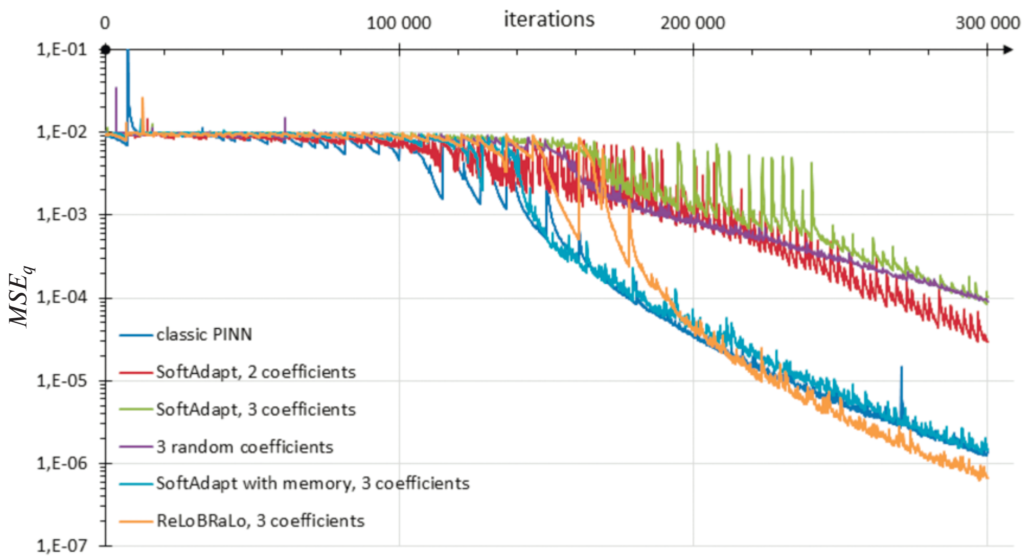

Fig. 2. MSE_q iteration dynamic.

Table 1. Comparison of runs with and without segmentation. Time is given in minutes, $Lw1$ and $Lw2$ are percentages, Res denotes absolute residual values, and Rel_h represents the relative error compared to the analytical solution

n segm	n neur	Res (mean)	$Lw1$ (mean)	$Lw2$ (mean)	Rel_h	Time, min
1	64	0.00008	1.7e-02	1.8e-02	8.3e-04	52.5
5	**32**	0.00011	**9.1e-03**	**1.5e-02**	9.2e-04	57.6
1	32	0.00017	6.3e-02	7.5e-02	1.7e-03	34.3

where $loss_i \in \{MSE_{ic}, MSE_{bc}, MSE_{eq}\}$.

There are different methods for calculating the values of these coefficients. In this work, the SoftAdapt and ReLoBRaLo methods proposed in [24] and [15], as well as some variations of these methods, were used. The results are presented in Fig. 2.

The blue curve corresponds to the absence of dynamic weight balancing. The comparison of curves shows that significant improvements can be achieved only by using the ReLoBRaLo method. Although the SoftAdapt method adjusts coefficients to be more suitable than random ones, the training proceeds more slowly compared to using constant coefficients. ReLoBRaLo demonstrates an advantage only with a large number of iterations (>200 000). In further experiments, we divided the initial domain into segments, with the number of iterations per segment being much smaller, around 50 000. Under these con-

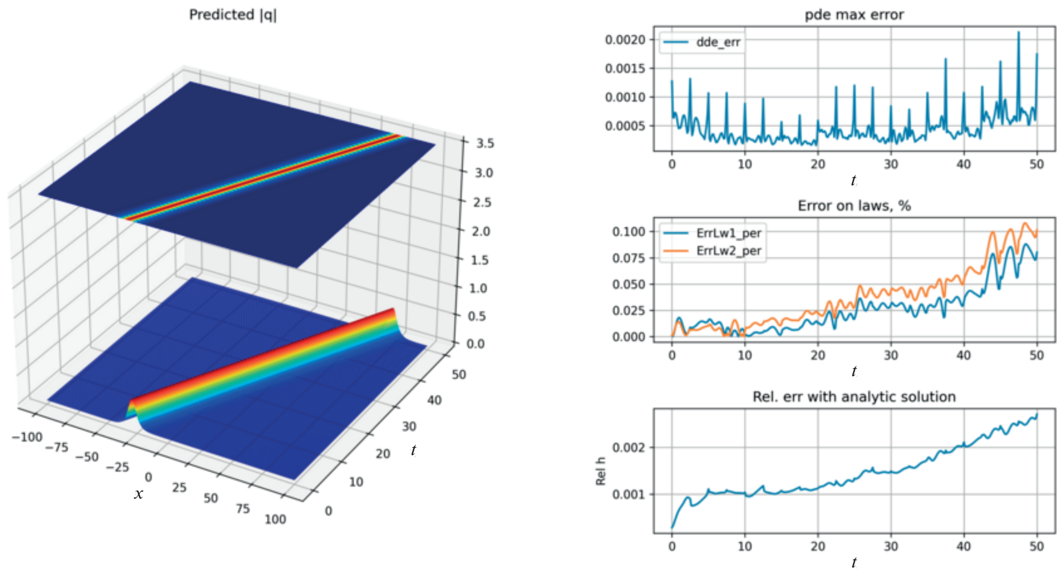

Fig. 3. Problem BVP2, single soliton. Predicted $|q|$—3D visualization with a heatmap (top) showing the magnitude of the function, with various errors detailed by time slices t shown on the right.

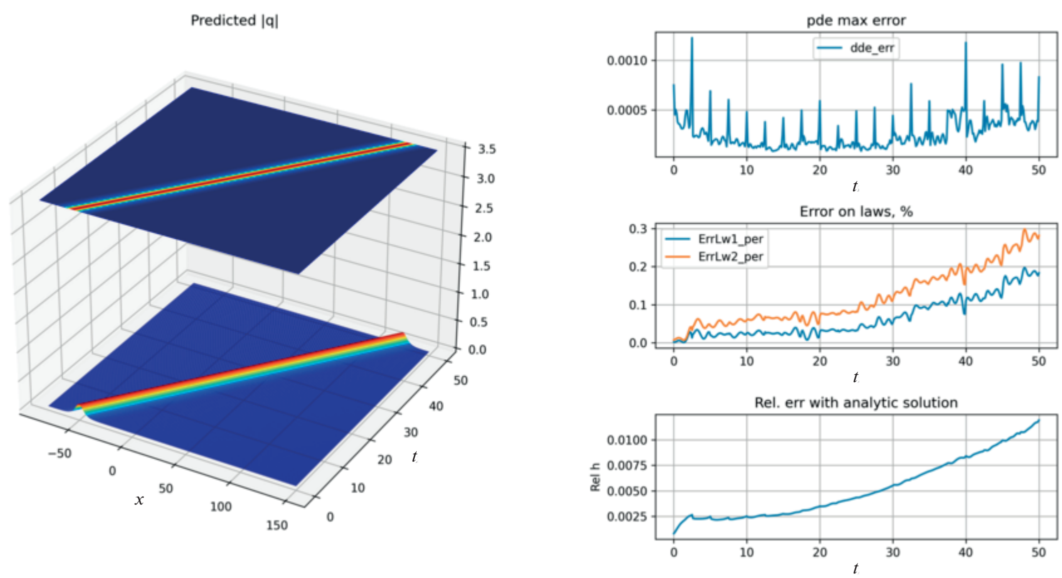

Fig. 4. Task BVP4, single soliton.

ditions, ReLoBRaLo would not provide a noticeable advantage and would only consume computational resources, making the use of dynamic balancing in subsequent experiments appear impractical.

3.3. Accuracy Assessment of the PINN Method with Segmentation by Comparison with the Exact Solution for Single Solitons of the 2nd and 4th Order NLS

When an exact solution is known, it is possible to evaluate how the error relative to the exact solution is associated with the error in the conservation laws for the chosen method. In this experiment, the boundary value problem BVP2 is considered with parameters as described in Subsection 3.1, but with increased domain boundaries $x_{\mathrm{L}} = -100$, $x_{\mathrm{R}} = 100$, $t_{\max} = 50$ to study the behavior of errors. A similar run is conducted for the fourth-order equation: The boundary value problem BVP4 is considered with $a_1 = 1$, $a_2 = 2$, $a_4 = 0.5$ and domain boundaries $x_{\mathrm{L}} = -80$, $x_{\mathrm{R}} = 150$, $t_{\max} = 50$ and initial condition $q_0(x) = q_4(x, 0)$, $A = 1$, $k = 0.5$, $\mu = -2$, $\nu = 0.088$.

The obtained solutions and error behavior for the

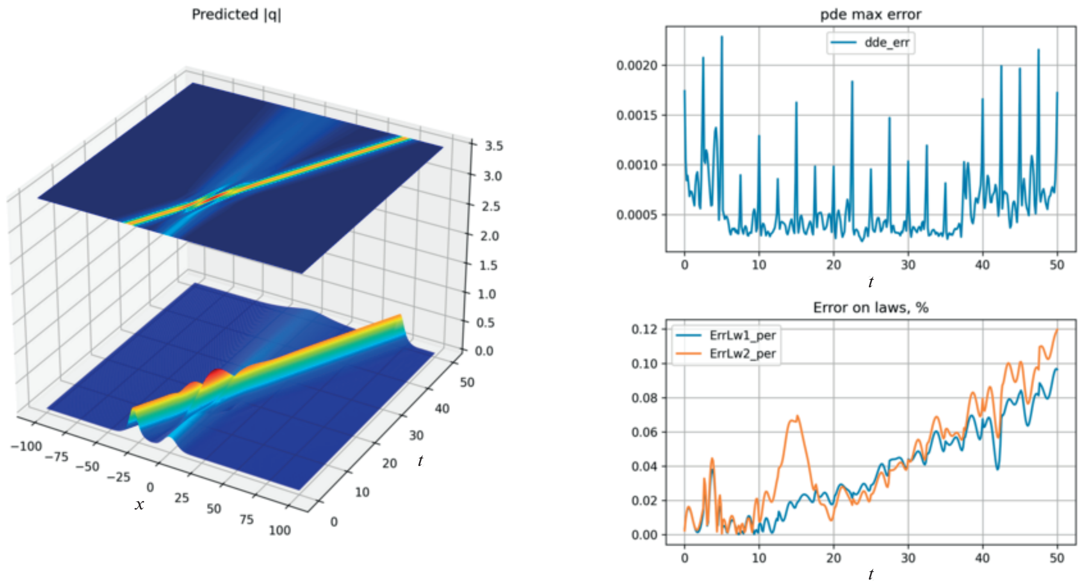

Fig. 5. BVP2, soliton-Gaussian interaction.

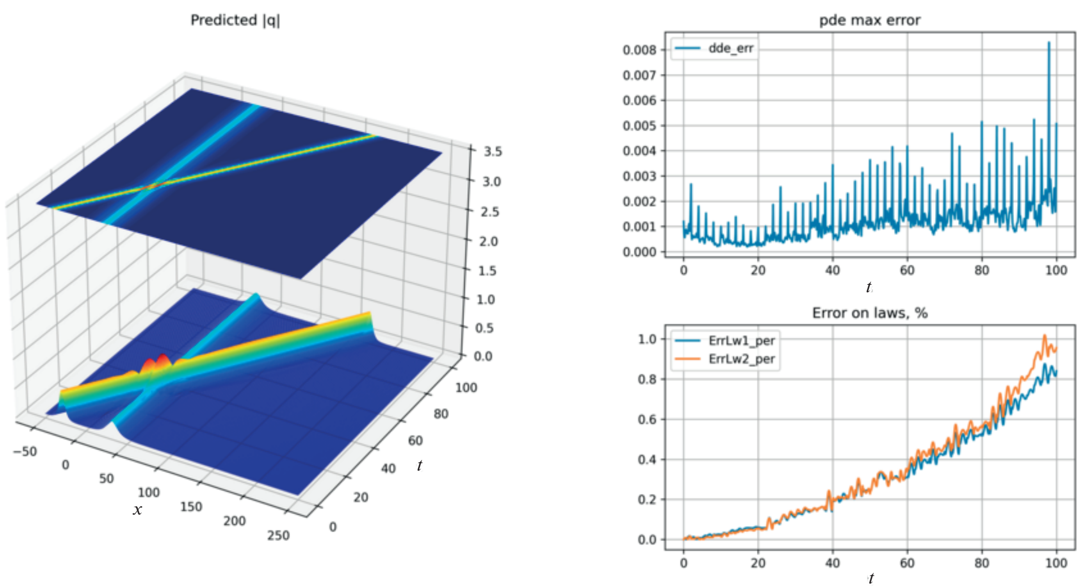

Fig. 6. Problem BVP2 soliton—soliton interaction.

equation, conservation laws, and the exact solution are presented in Figs. 3 and 4.

The obtained calculations show that when applying this method to boundary value problems with a single soliton initial condition, the error relative to the exact solution strongly correlates with the error in the conservation laws. It can be assumed that this correlation will persist, and hence, errors in the conservation laws may be used as error metrics in cases where the exact solutions are unknown.

In the case of an unknown solution (as in the experiments below), it is not possible to calculate the relative error Rel_h, and thus residuals and errors in the conservation laws Res, $Lw1$, and $Lw2$ become the only means for assessing the correctness of the results.

3.4. Investigation of Soliton Stability under Perturbation

3.4.1. Interaction of a soliton with a perturbation in the form of a Gaussian function for the 2nd order NLS equation. To study the interaction of a soliton, described by the 2nd-order equation model, with a solitary wave in the form of a

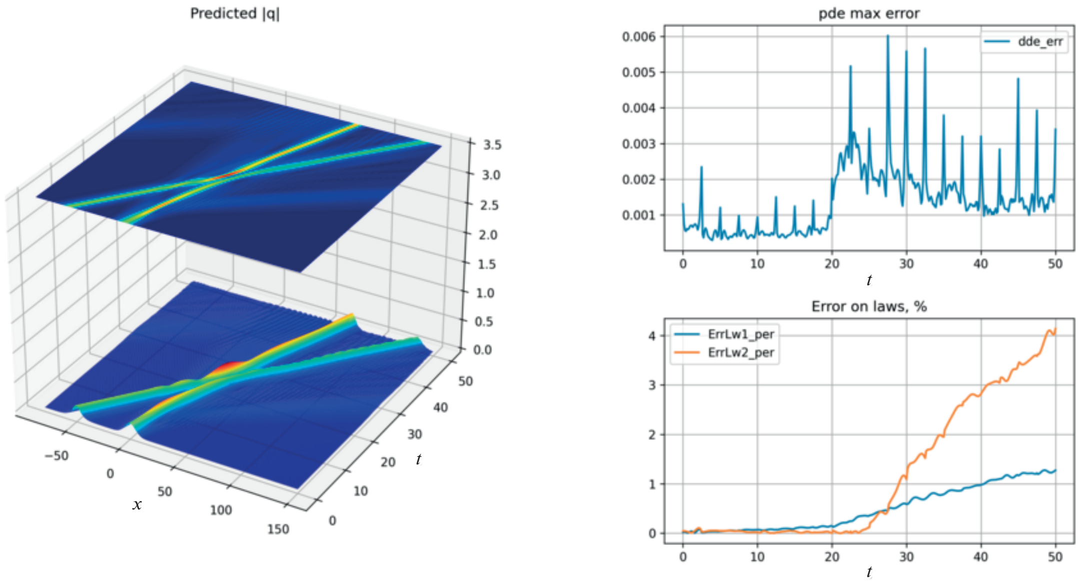

Fig. 7. BVP4 soliton with a Gaussian perturbation.

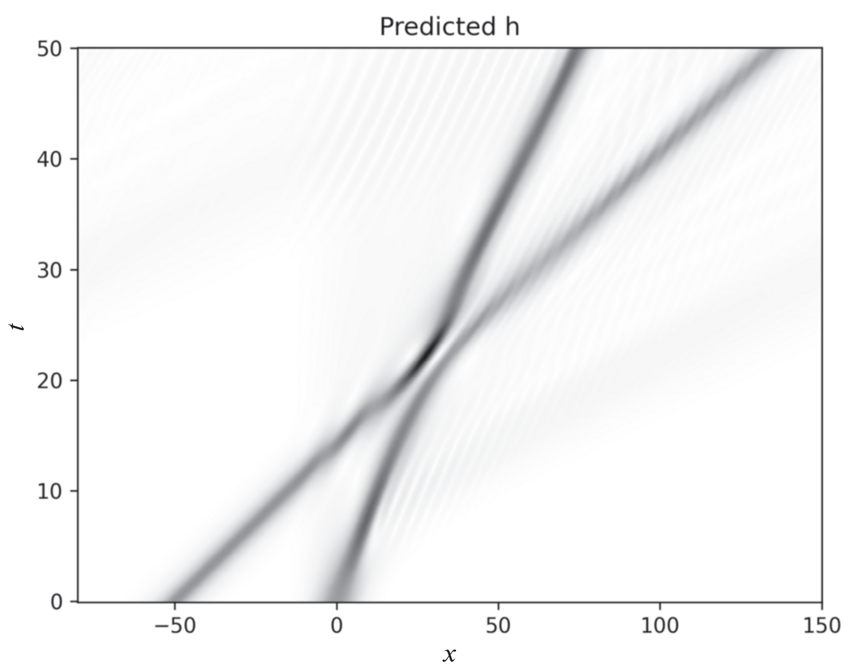

Fig. 8. Problem BVP4, soliton and Gaussian function, heatmap of the magnitude of the function $|\hat{q}|$, with darker regions indicating higher values.

Gaussian function, the solution to the boundary value problem BVP2 is presented with $\alpha = 0.3$, $\beta = 0$, over the domain $x_{\mathrm{L}} = -100$, $x_{\mathrm{R}} = 100$, $t_{\max} = 50$ and the initial condition:

$$q_0(x) = q_2(x,0) + A_g e^{-\frac{(x-B_g)^2}{C_g^2}},$$

where $A_g = 0.2$, $B_g = 0$, $C_g = 5$, $\alpha_0 = 0.3$, $k = 1$, $\omega = 0.88$, and $x_0 = -30$.

The obtained result (Fig. 5) indicates that the soliton interacts minimally with the solitary wave in the form of a Gaussian function.

3.4.2. Interaction of a soliton with a stationary soliton solution ($k = 0$) for the 2nd order NLS equation. To investigate the interaction of solitary solitons described by the model, the boundary value problem BVP2 is considered with $\alpha = 0.3$, $\beta = 0$, over the domain $x_{\mathrm{L}} = -100$, $x_{\mathrm{R}} = 100$, $t_{\max} = 50$ and the initial condition in the form of a sum of sufficiently distant solitons, each of which is a solution:

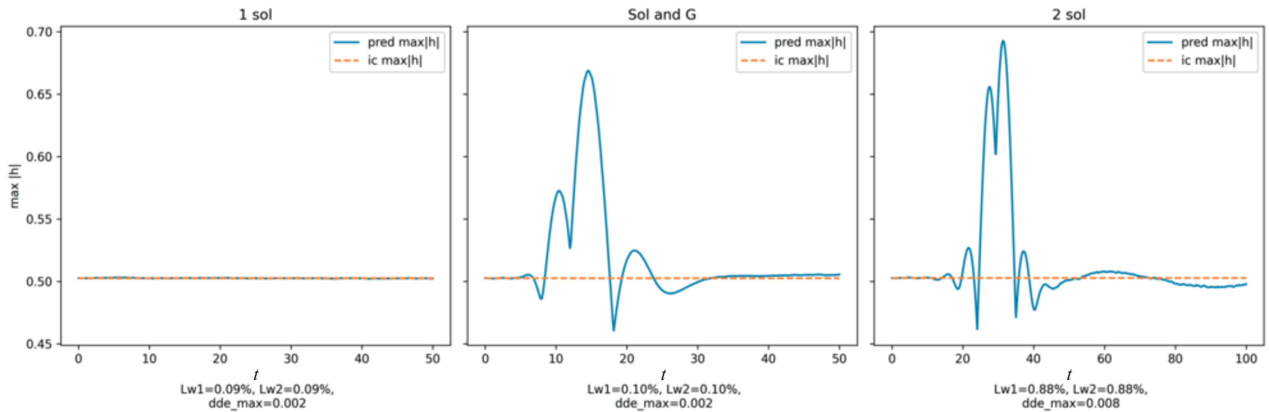

Fig. 9. Problem BVP2, maximum value of the function modulus over time for 3 runs: single soliton, soliton–Gaussian Interaction, soliton–soliton Interaction.

$$\varphi_{\text{2sol2ord}}(x) = \sum_{j=1}^{2} \sqrt{\frac{\mu_j e^{(x-x_j)\sqrt{\mu_j}}}{\left(\frac{1}{2}e^{(x-x_j)\sqrt{\mu_j}}+1\right)^2 - \frac{\alpha\mu_j}{3}e^{2(x-x_j)\sqrt{\mu_j}}}}\, e^{i(k_j x+\theta_j)},$$

where $k_1 = 1$, $k_2 = 0$, $x_1 = -30$, $x_2 = 30$.

The obtained result (Fig. 6) shows that for this model (described by the 2nd-order equation), solitary solitons remain solitary even after interaction, and the interaction does not affect their long-term behavior.

3.4.3. Interaction of a soliton with a perturbation in the form of a Gaussian function for the 4th order NLS equation. To study the interaction of a soliton described by the 4th-order equation with a solitary wave in the form of a Gaussian function, the solution to the boundary value problem BVP4 is presented with $a_1 = 1$, $a_2 = 1$, $a_4 = 0.5$, $k = 0.5$, $A = 1$, $\mu = 2$ over the domain $x_{\text{L}} = -100$, $x_{\text{R}} = 100$, $t_{\max} = 50$, and the initial condition: $q(x,0) = q_4(x,0) + A_g e^{-\frac{(x-B_g)^2}{C_g^2}}$, where $A_g = 0.2, B_g = 0$, and $C_g = 5$.

From Fig. 7, it is evident that the fourth-order soliton, with the given parameters, elastically interacts with the perturbation in the form of a Gaussian function. This is clearly visible in Fig. 8.

For the fourth-order equation (4), the numerical experiment demonstrated elastic interaction of a single bright soliton solution with a perturbation in the form of a Gaussian function.

4. RESULT

The paper demonstrates the applicability of the PINN method for solving boundary value problems of modified Schrödinger equations of the second and fourth order. Approximate solutions were obtained for cases involving interactions between soliton-like solutions with another soliton or a Gaussian function. The results indicate the stability of the found soliton solution for the second-order equation (2), at least for the chosen parameters of the equation and initial conditions.

An illustration of this is shown in Fig. 9, which presents curves of the maximum values of the solution magnitudes over time, demonstrating the stability of the second-order soliton solutions during interactions with another soliton or a solitary perturbation in the form of a Gaussian function.

ACKNOWLEDGEMENTS

The work was carried out using the equipment of the collective use center "Complex for Modeling and Processing Data from Megascience Research Facilities" of the National Research Center "Kurchatov Institute," http://ckp.nrcki.ru/.

FUNDING

The work was financially supported by the Russian Foundation for Basic Research, grant no. 23-41-00070, https://rscf.ru/project/23-41-00070/.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. M. Raissi, P. Perdikaris, and G. E. Karniadakis, arXiv Preprint (2017). https://doi.org/10.48550/arXiv.1711.10561
2. K. Hornik, M. Stinchcombe, and H. White, Neural Networks **2**, 359 (1989). https://doi.org/10.1016/0893-6080(89)90020-8
3. A. Sergeev and M. Del Balso, arXiv Preprint (2018). https://doi.org/10.48550/arXiv.1802.05799
4. T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb, IMA J. Appl. Math. **89**, 143 (2024). https://doi.org/10.1093/imamat/hxae011
5. R. Mattey and S. Ghosh, Comput. Methods Appl. Mech. Eng. **390**, 114474 (2022). https://doi.org/10.1016/j.cma.2021.114474
6. S. Wang, S. Sankaran, and P. Perdikaris, arXiv Preprint (2022). https://doi.org/10.48550/arXiv.2203.07404
7. A. A. Ramabathiran and P. Ramachandran, J. Comput. Phys. **445**, 110600 (2021). https://doi.org/10.1016/j.jcp.2021.110600
8. N. Wandel, M. Weinmann, M. Neidlin, and R. Klein, Proc. AAAI Conf. Artif. Intell. **36**, 8529 (2022).
9. L. Lu, P. Jin, G. Pang, Zh. Zhang, and G. E. Karniadakis, Nat. Mach. Intell. **3**, 218 (2021). https://doi.org/10.1038/s42256-021-00302-5
10. X. Jiang, D. Wang, Q. Fan, M. Zhang, Ch. Lu, and A. P. T. Lau, Laser Photonics Rev. **16**, 2100483 (2022). https://doi.org/10.1002/lpor.202100483
11. J. Kim, K. Lee, D. Lee, S. Y. Jhin, and N. Park, Proc. AAAI Conf. Artif. Intell. **35**, 8146 (2021).
12. Yi. Fang, G.-Z. Wu, N. A. Kudryashov, Y.-Y. Wang, and C.-Q. Dai, Chaos, Solitons Fractals **158**, 112118 (2022). https://doi.org/10.1016/j.chaos.2022.112118
13. M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019). https://doi.org/10.1016/j.jcp.2018.10.045
14. B. Moseley, A. Markham, and T. Nissen-Meyer, Adv. Comput. Math. **49**, 62 (2023). https://doi.org/10.1007/s10444-023-10065-9
15. R. Bischof and M. Kraus, arXiv Preprint (2021). https://doi.org/10.48550/arXiv.2110.09813
16. A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, Comput. Methods Appl. Mech. Eng. **365**, 113028 (2020). https://doi.org/10.1016/j.cma.2020.113028
17. G.-Z. Wu, Yi. Fang, N. A. Kudryashov, Y.-Y. Wang, and C.-Q. Dai, Chaos, Solitons Fractals **159**, 112143 (2022). https://doi.org/10.1016/j.chaos.2022.112143
18. Zh. Mao, A. D. Jagtap, and G. E. Karniadakis, Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020). https://doi.org/10.1016/j.cma.2019.112789
19. J. Gurieva, E. Vasiliev, and L. Smirnov, Procedia Computer Science **212**, 464 (2022). https://doi.org/10.1016/j.procs.2022.11.030
20. A. G. Sboev, N. A. Kudryashov, I. A. Moloshnikov, D. R. Nifontov, S. V. Zavertyaev, and R. B. Rybka, in *Biologically Inspired Cognitive Architectures Meeting*, Ed. by A. V. Samsonovich and T. Liu, Studies in Computational Intelligence, Vol. 1130 (Cham, 2023), p. 759. https://doi.org/10.1007/978-3-031-50381-8_82
21. N. A. Kudryashov and D. R. Nifontov, Chaos, Solitons Fractals **175**, 114076 (2023). https://doi.org/10.1016/j.chaos.2023.114076
22. N. A. Kudryashov, Optik **261**, 169163 (2022). https://doi.org/10.1016/j.ijleo.2022.169163
23. A. A. Bayramukov and N. A. Kudryashov, J. Comput. Appl. Math. **437**, 115497 (2024). https://doi.org/10.1016/j.cam.2023.115497
24. A. A. Heydari, C. A. Thompson, and A. Mehmood, arXiv Preprint (2019). https://doi.org/10.48550/arXiv.1912.12355