DATA-DRIVEN FORWARD-INVERSE PROBLEMS FOR THE VARIABLE COEFFICIENTS HIROTA EQUATION USING DEEP LEARNING METHOD

HUIJUAN ZHOU, JUNCAI PU, AND YONG CHEN

ABSTRACT. Data-driven forward-inverse problems for the variable coefficients Hirota (VC-Hirota) equation are discussed in this paper. An improved physics-informed neural networks (IPINN) algorithm is used to recover the data-driven soliton solutions, high-order soliton, as well as the data-driven parameters discovery for the VC-Hirota equation. We propose a PINN algorithm with sub-neural networks to learn the data-driven functions discovery of VC-Hirota equation with unknown functions under noise of different intensity. Numerical results are shown to demonstrate the facts: (i) data-driven soliton solutions and parameters discovery of the VC-Hirota equation are successfully learned by adjusting the network layers, neurons, the original training data, spatiotemporal regions and other parameters of the IPINN algorithm; (ii) the data-driven functions discovery of VC-Hirota equation can be trained stably and accurately via the PINN algorithm with sub-neural networks. The results achieved in this work verify that the forward-inverse problems including the data-driven function discovery of the variable coefficients equation can be solved based on deep learning method.

1. INTRODUCTION

As an important branch of nonlinear systems, solitons have been widely studied in many fields, such as plasma physics, fluid mechanics, optical fiber communication, condensed matter physics and so on [1–6]. In the field of optical fiber communication, it is well known that a modified NLS equation, which is called Hirota equation, can be used to describe the propagation of subpicosecond or femtosecond optical pulse in fibers thanks to it taking into account higher-order dispersion and time-delay corrections to the cubic nonlinearity. Though there are extensive studies about Hirota equation and some types of exact soliton solutions in optical have been obtained, it is worth noting that these investigations of optical solitons have revolved mainly around homogeneous fibers. However, considering long-distance communication and manufacturing problems in realistic fiber transmission lines, the inhomogeneous variable coefficient model needs to be considered.

Taking into account the higher-order effects influenced by spatial variations of the fiber parameters, the inhomogeneous variable coefficient Hirota (VC-Hirota) equation which can be used to describe the certain ultrashort optical pulses propagating in a nonlinear inhomogeneous fiber is as follows [7],

\[ iq_t + \alpha_1(z)q_{tt} - \frac{1}{3}i\alpha_2(z)q_{ttt} + \delta\alpha_2(z)|q|^2q_t - i\alpha_3(z)q = 0, \]

(1.1)

where \( \alpha_2(z) = \frac{\alpha_1(z) - \alpha_1(0)z^\delta}{\alpha_1(z) - \alpha_1(0)^z} \), \( q = q(t,z) \) is complex-valued solution about the space \( t \) and time \( z \), \( \delta \) is a real number, \( \alpha_1(z) \) and \( \alpha_2(z) \) are dispersion and nonlinear effects, respectively. In order to understand the nonlinear phenomena of optical pulse propagation in inhomogeneous fiber media, it is necessary to analyze the analytical and numerical solutions of the VC-Hirota model. Due to the important application of the VC-Hirota model, there has been much research on this model, such as the exact bright and dark solitary wave solutions near the zero dispersion point is derived in [8]. Three combined solitary wave solutions are given in the same expression, and the properties of bright and dark solitary waves are described respectively. Furthermore, the features of the solutions are analyzed, and numerically discuss the stability of these solitary waves under slight violations of the parameter conditions and finite initial perturbations [9]. It has also been extensively studied by many other authors and some types of localized waves solution have been obtained [10–15]. Recently, we derive the exact form of \( N \)-soliton and high-order soliton solutions for the VC-Hirota equation by utilizing the Riemann-Hilbert approach [16].

In the past few decades, many methods and techniques have been presented to solve the nonlinear evolution equations, including Hirota bilinear method, Darboux transformation, Riemann-Hilbert method, physics-informed neural networks (PINN) deep learning method, etc. In scientific computing, the neural network (NN) method provides an ideal representation for the solution of differential equations due to its universal approximation properties. Recently, a PINN method which is controlled by mathematical physical systems based on the multi-layer NNs has been proposed and proved to be particularly suitable for dealing with both the forward problems and highly ill-posed inverse problems by obtaining the approximate solutions of governing equations and discovering parameters involved in the governing equation are inferred from the training data. Numerical results show that the PINN architecture is able to obtain remarkably accurate solutions with remarkably little data [17]. At the same time, this method also provides a better physical explanation for predicted solutions because of the underlying physical constraints, which are usually explicitly described by differential equations. Later on, using the PINN method to obtain data-driven solutions, parameters discovery and reveal the dynamic behavior of nonlinear partial differential equations with physical constraints has attracted extensive attention and raised a hot wave of research. Recently, PINN has played an important role in many
physical applications [18]. Afterwards, global adaptive activation functions and locally adaptive activation functions are proposed to approximate smooth and discontinuous functions as well as solutions of linear and nonlinear partial differential equations by introducing a scalable parameters in the activation function and adding a slope recovery term based on activation slope to the loss function of locally adaptive activation functions. It demonstrates the locally adaptive activation functions further improve the training speed, performance and speed up the training process of NNs. [19][20].

Due to the good properties of integrable systems, it is possible to solve the data-drive forward and inverse problems for abundant classical integrable nonlinear evolution equations via the PINN method. Since Chen’s team first proposed the concept of integrable deep learning in 2020, a large number related literature has been published. In particular, Li and Chen construct abundant numerical solutions of second-order and third-order nonlinear integrable equations with different initial and boundary conditions by deep learning method based on the PINN model [21][22]. Pu, Peng et al. recover the solitons, breathers, rogue wave solutions and rogue wave on the periodic background of the nonlinear partial differential equation with the aid of the PINN model [23][24]. Miao and Chen use the PINN method to high-dimensional system to solve the $(N+1)$-dimensional initial boundary value problem with $2N+1$ hyperplane boundaries [25]. Lin and Chen devise a two-stage PINN method which is tailored to the nature of equations by introducing features of physical systems into NNs based on conserved quantities [26]. In addition, many important work on data-driven solutions and parameter discovery of different nonlinear systems have been done by other scholars [27][32].

In the previous literature, many soliton solutions of constant coefficients integrable systems have been accurately simulated numerically. The soliton solutions of variable coefficients integrable systems studied in this paper are quite different from the soliton solutions of constant coefficients equations. From the perspective of the solution dynamics, the center trajectory equation of the one-soliton solution of the constant coefficient equation is often a straight line, while the one-soliton solution of the variable coefficient equation presents a richer and more complex shape. For example, in this paper, we learned the case of the center trajectory equations of the one-solitons are parabola, “S-shape” and cosine wave type. For the two-soliton solution of variable coefficients equation, it can take on a more complex shape. To our knowledge, there is rare study about the data-driven soliton solutions, parameter and function discovery for the variable coefficient equation by the IPINN method. To fill this gap, the main purpose of this paper is to recover various solitons, high-order soliton, as well as parameters and functions discovery for the variable coefficient equation when only know the initial-boundary value conditions. It is note that we use PINN algorithm with sub-neural networks to achieve data-driven function discovery of variable coefficient equations.

The rest of this paper is built up as follows. In Section 2, the IPINN method for the VC-Hirota equation is introduced. To provide a direct and precise description of the IPINN method, the algorithm flow schematic and algorithm steps are given. Section 3 provides the data-driven multi-soliton and high-order soliton solutions of the VC-Hirota equation using the IPINN approach, and related plots and dynamic analyses are revealed in detail. Section 4 presents experimental results of learning data-driven parameter discovery of VC-Hirota system. Furthermore, we learn the unknown function of the VC-Hirota via a PINN algorithm with sub-neural networks successfully. The conclusions and discussions are given in Section 5.

2. The improved physics-informed neural networks method for the variable coefficient Hirota equation

An improved PINN (IPINN) approach with neuron-wise locally adaptive activation function was presented to derive data-driven localized waves and learn unknown parameters of integrable systems in complex space, and numerical results demonstrated the improved approach has faster convergence and better simulation effect than the classical PINN method [33][34]. In the following, we will introduce the IPINN method in detail and extend it to the application of variable coefficient Hirota equations.

Considering the $(1+1)$-dimensional nonlinear time-dependent systems with unknown constant parameter $\delta$ and variable coefficients $\alpha_1(z), \alpha_2(z)$ and $\alpha_3(z)$ in complex space as below.

$$q_z + \mathcal{N}[q; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)] = 0, \quad (2.1)$$

where $\mathcal{N}[\cdot; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)]$ is nonlinear differential operators in space. In order to simplify the structure of the complex-valued solutions $q(t, z)$ in Eq. (2.1), we decompose $q(t, z)$ into $u(t, z) + iv(t, z)$, where $u(t, z)$ and $v(t, z)$ are real-valued functions. Then substituting $q(t, z) = u(t, z) + iv(t, z)$ into Eq. (2.1), and letting the real and imaginary parts be equal to 0, we have

$$u_z + \mathcal{N}_u[u; v; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)] = 0, \quad (2.2)$$

$$v_z + \mathcal{N}_v[u; v; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)] = 0,$$

where the $\mathcal{N}_u$ and $\mathcal{N}_v$ are nonlinear differential operators in space. Defining the physics-informed part $f_u(x, t)$ and $f_v(x, t)$ as

$$f_u := u_z + \mathcal{N}_u[u; v; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)],$$

$$f_v := v_z + \mathcal{N}_v[u; v; \delta, \alpha_1(z), \alpha_2(z), \alpha_3(z)],$$

which play a role of regularization.
The IPINN of depth $D$ corresponding to the NN with an input layer, $D-1$ hidden-layers and an output layer. $N_d$ presents the number of neurons in the $d$th hidden-layer, and each hidden-layer of the IPINN receives an output $x^{d-1} \in \mathbb{R}^{N_{d-1}}$ from the previous layer. Denote an affine transformation as follows:

$$
\mathcal{L}_d(x^{d-1}) = W^d x^{d-1} + b^d,
$$

(2.4)

where the network weights $W^d \in \mathbb{R}^{N_d \times N_{d-1}}$ and bias term $b^d \in \mathbb{R}^{N_d}$ are associated with the $d$th layer. Then define neuron-wise locally adaptive activation function as

$$
\sigma \left( na^d_i \left( \mathcal{L}_d (x^{d-1}) \right) \right), \quad n > 1, \cdots, d = 1, 2, \cdots, D - 1, i = 1, 2, \cdots, N_d,
$$

where $\sigma$ is the activation function, $n$ is a scaling factor and $\{a^d_i\}$ is additional $\sum_{d=1}^{D-1} N_d$ parameter to be optimized. When $n$ greater than or equal to critical scaling factor $n_c$ in each problem set, the optimization algorithm will become sensitive. The neuron activation function acts as a vector activation function in each hidden layer, and each neuron has its own activation function slope.

The IPINN method with neuron-wise locally adaptive activation function can be expressed as:

$$
q(x; \Theta) = \left( \left( \mathcal{L}_D \right)_i \circ \sigma \circ na^{D-1}_i \left( \mathcal{L}_{D-1} \right)_i \circ \cdots \circ \sigma \circ na^1_i \left( \mathcal{L}_1 \right)_i \right) (x), \quad i' = 1, 2,
$$

(2.5)

where $x$ represent the two inputs and $q(x; \Theta)$ represent the two outputs in the IPINN. The trainable parameters set $\Theta \in \mathcal{P}$ consists of $\{W^d, b^d\}_{d=1}^D$ and $\{a^d_i\}_{d=1}^{D-1}, \forall i = 1, 2, \cdots, N_d$, where $\mathcal{P}$ is a parameter space. In order to minimize the loss function below certain tolerance $\varepsilon$ until a prescribed maximum number of iterations, seek a optimal values of weights $W$, biases $b$ and scalable parameter $a^d_i$ is necessary. Here, we initialize the scalable parameters in the case that $na^d_i = 1, \forall n \geq 1$ (we fixed $n=10$ in this paper).

Define the loss function as follows:

$$
\mathcal{L}'(\Theta) = \text{Loss} = \text{Loss}_u + \text{Loss}_v + \text{Loss}_{f_u} + \text{Loss}_{f_v} + \text{Loss}_a,
$$

(2.6)

where $\text{Loss}_u, \text{Loss}_v, \text{Loss}_{f_u}, \text{Loss}_{f_v}$ and $\text{Loss}_a$ are defined as below:

$$
\text{Loss}_u = \frac{1}{N_u} \sum_{j=1}^{N_u} \left| \hat{u}(t^j, z^j) - u^j \right|^2,
$$

$$
\text{Loss}_v = \frac{1}{N_v} \sum_{j=1}^{N_v} \left| \hat{v}(t^j, z^j) - v^j \right|^2,
$$

$$
\text{Loss}_{f_u} = \frac{1}{N_f} \sum_{l=1}^{N_f} \left| f_u(t^l_f, z^l_f) \right|^2,
$$

(2.7)

$$
\text{Loss}_{f_v} = \frac{1}{N_f} \sum_{l=1}^{N_f} \left| f_v(t^l_f, z^l_f) \right|^2,
$$

$$
\text{Loss}_a = \frac{N_a}{D-1 \sum_{d=1}^{D-1} \exp \left( \sum_{i=1}^{N_a} a^d_i \right)}.
$$

$\{t^j, z^j, u^j, v^j\}_{j=1}^{N_u}$ denote the inputs data of initial-boundary value on Eqs. (2.2) and (2.3). $\hat{u}(t^j, z^j)$ and $\hat{v}(t^j, z^j)$ represent the optimal training outputs data through the IPINN. Furthermore, $\{t^l_f, z^l_f\}_{l=1}^{N_f}$ represent the collocation points on networks $f_u(t, z)$ and $f_v(t, z)$. $N_u$ is the hyper-parameter for slope recovery term $\text{Loss}_u$, and we take $N_u = \frac{1}{100}$ for dominating the loss function to ensure that the final loss value is not too large in this paper. $\text{Loss}_v$ and $\text{Loss}_a$ correspond to the loss on the initial and boundary data. $\text{Loss}_{f_u}$ and $\text{Loss}_{f_v}$ penalize the collocation points which not satisfy the VC-Hirota equation. The $\text{Loss}_a$ changes the topology of $\text{Loss}$ function forces the NN to increase the activation slope value quickly, which ensures the non-vanishing gradient of the loss function and improves the convergence speed and network optimization ability. Therefore, the loss function is evaluated using the contribution from the NN part as well as the residual from the governing equation given by the physics-informed part. The resulting optimization algorithm will attempt to find the optimized parameters including the weights, biases and additional coefficients in the activation to minimize the new loss function. In addition, the $L_2$ norm error is introduced to measure the training error better. $L_2$ norm error is defined as follows:

$$
\text{Error} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} \left| q^\text{exact}(x_k) - q^\text{predict}(x_k; \Theta) \right|^2},
$$

where $\{q^\text{exact}(x_k)\}_{k=1}^{N}$ denote the exact data, and $\{q^\text{predict}(x_k; \Theta)\}_{k=1}^{N}$ denote the predicted data.
where $q_{\text{predict}}(x_k; \Theta)$ represent the model training prediction solution and $q_{\text{exact}}(x_k)$ represent the exact analytical solution at point $x_k = (t_k, z_k)$.

The physics-informed parts of the IPINN for VC-Hirota equation (1.1) can be defined as:

$$f_u := -v_z + \alpha_1(z)u_{tt} + \frac{1}{3\delta} \alpha_1(z)u_t + \delta \alpha_2(z)(u^3 + uu^2) + \alpha_2(z)(u_t^2 u_t + v^2 u_t) + \alpha_3(z)v,$$

$$f_v := u_z + \alpha_1(z)v_t - \frac{1}{3\delta} \alpha_1(z)u_t + \delta \alpha_2(z)(uv^2 + v^3) - \alpha_2(z)(u_t^2 u_t + v^2 u_t) - \alpha_3(z)u.$$

The physics-informed parts of the IPINN for VC-Hirota equation (1.1) is integrable due to the equation (1.1) have the lax pair as follows:

$$U = \begin{bmatrix} \lambda & \sqrt{\frac{\delta \alpha_2(z)}{2\alpha_1(z)}} q(z, t) \\ -q^*(z, t) \sqrt{\frac{\delta \alpha_2(z)}{2\alpha_1(z)}} & -\lambda \end{bmatrix},$$

$$V = \begin{pmatrix} A & B \\ C & -A \end{pmatrix},$$

where

$$A = \frac{4\alpha_1(z)}{3\delta} \lambda^3 + 2i\alpha_1(z)\lambda^2 + \frac{\delta \alpha_2(z)(q(z, t))^2}{3\delta} \lambda + \frac{\delta \alpha_2(z)(q^*(z, t)q(z, t) - q(z, t)q^*(z, t))}{6\delta} + \frac{i\delta \alpha_2(z)q(z, t)q^*(z, t)}{2},$$

$$B = \sqrt{\frac{\delta \alpha_2(z)}{2\alpha_1(z)}} \left( \frac{4\alpha_1(z)q(z, t)}{3\delta} \lambda^2 + \frac{\alpha_1(z)q(z, t)}{3\delta} + i\alpha_1(z)q(z, t) \right) + \frac{\alpha_1(z)}{3\delta} (q^*_t(z, t) + \frac{\delta \alpha_2(z)(q(z, t))^2 q^*(z, t)}{\alpha_1(z)}),$$

and

$$C = \sqrt{\frac{\delta \alpha_2(z)}{2\alpha_1(z)}} \left( -\frac{4\alpha_1(z)q^*(z, t)}{3\delta} \lambda^2 + \frac{\alpha_1(z)q^*(z, t)}{3\delta} - i\alpha_1(z)q^*(z, t) \right) + \frac{\alpha_1(z)}{3\delta} (q^*_t(z, t) + \frac{\delta \alpha_2(z)(q(z, t))^2 q^*(z, t)}{\alpha_1(z)})].$$

In order to describe the IPINN algorithm for the VC-Hirota equation more intuitively, we give the flow chart of the algorithm in Fig. 1. Since the complex function $q$ is decomposed into $u + iv$, one can see that the “NN” part has two output functions $\{u, v\}$, and there are two nonlinear equation constraints in the physics-informed part. What need to note is that in the physics-informed part, we introduce the constant parameter $\delta$ and variable coefficients $\alpha_1(z), \alpha_2(z), \alpha_3(z)$ into the governing equation (2.3). Note $f(z) = \alpha_1(z), \alpha_2(z), \alpha_3(z)$ and $\alpha_3(z) = \frac{\alpha_1(z)\alpha_2(z) - \alpha_3(z)}{2\alpha_1(z)\alpha_2(z)}$ is a necessary constraint for the integrability of the equation (2.3). In addition, we also show the corresponding procedure steps of the IPINN algorithm for the VC-Hirota equation in Tab. 1.

![Figure 1](image-url) (Color online) Schematic of IPINN for the VC-Hirota equation. The left NN is the universal approximation network while the right one induced by the governing equation is the physics-informed network. The two NNs share hyper-parameters and they both contribute to the loss function.
Table 1. IPINN algorithm of the VC-Hirota equation.

| Step one | Introduce constant parameter δ and variable coefficients α₁(z), α₂(z), α₃(z) into the governing equation (2.3) and take α₃(z) = \( \frac{α₁α₂ - α₁α₃}{2α₁α₂} \) to insure Eq. (2.3) completely integrable. |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step two | Specification of training set in computational domain: |
| Training data: \( \{t^j, z^j, w^j, v^j\}_{j=1}^{N_x} \), Residual training points: \( \{t^j, z^j\}_{j=1}^{N_x} \). |
| Step three | Construct NN \( q(x; Θ) \) with random initialization of parameters Θ. |
| Step four | Construct the residual NN \( \{f_u, f_v\} \) by substituting surrogate \( q(x; Θ) \) into the governing equations using automatic differentiation and other arithmetic operations. |
| Step five | Specification of the loss function \( ℒ(Θ) \) that includes the slope recovery term. |
| Step six | Find the best parameters \( Θ^* \) using a suitable optimization method for minimizing the loss function \( ℒ(Θ) \) as \( Θ^* = \arg\min_{Θ∈Ω} ℒ(Θ) \). |

Adam and L-BFGS algorithms are used to optimize all loss functions in the IPINN method. The Adam optimization algorithm is a variant of the traditional stochastic gradient descent algorithm, while the L-BFGS optimization algorithm is a full-batch gradient descent optimization algorithm based on the quasi-Newton method [35, 36]. In particular, unless otherwise stated, the scalable parameters in the adaptive activation function are typically initialized as \( n = 10, a_i^d = 0.1 \). In addition, we chose feedforward NNs with Xavier initialization and hyperbolic tangent (tanh) as activation functions. All the code in this article is based on Python 3.7 and Tensorflow 1.15. All the numerical experiments reported in this article were run on a DELL Precision 7920 Tower computer. The computer is equipped with a 2.10 GHz 8-core Xeon Silver 4110 processor, 64 GB memory and an 11 GB NVIDIA GeForce GTX 1080 Ti video card.

3. DATA-DRIVEN FORWARD PROBLEMS OF THE VC-HIROTA EQUATION

In this section, we will focus on the data-driven forward for the VC-Hirota equation with Dirichlet boundary conditions and initial conditions by means of IPINN method. The VC-Hirota equation with initial-boundary value conditions is as follows.

\[
\begin{aligned}
  iq_z + α_1(z)qtt - \frac{i}{3}α_1(z)qttt + δα_2(z)q|q|^2 - iα_2(z)|q|^2q_t - iα_3(z)q &= 0, \\
  q(L_0, z) &= q^b(z), \quad q(L_1, z) = q^{ab}(z), \\
  q(t, T_0) &= q^0(t), \quad t \in [L_0, L_1], \quad z \in [T_0, T_1],
\end{aligned}
\]

(3.1)

where “\( i \)” is an imaginary number, the subscripts denote the partial derivatives of the complex fields \( q(t, z) \) with respect to the space \( t \) and time \( z \), while the \( L_0 \) and \( L_1 \) represent the lower and upper boundaries of \( t \) respectively. Similarly, \( T_0 \) and \( T_1 \) represent the initial and final times of \( z \) respectively. Moreover, the \( q^0(t) \) represents initial value of the \( q(t, z) \) at \( z = T_0 \), the \( q^{b}(z) \) and \( q^{ab}(z) \) are the lower and upper boundaries of the \( q(t, z) \) corresponding to \( t = L_0 \) and \( t = L_1 \) respectively.

The \( N \)-solitons of the VC-Hirota have been derived by the Riemann-Hilbert method in [16], which can be expressed as follows:

\[
q = -2 \left| \frac{F}{M} \right| f(z)e^{iφ(t,z)},
\]

(3.2)

where \( F \) is the following \((N + 1) \times (N + 1)\) matrix

\[
\begin{pmatrix}
  0 & e^{-θ_1} & \cdots & e^{-θ_N} \\
  c_1 e^{θ_1} & M_{11} & \cdots & M_{1N} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_N e^{θ_N} & M_{N1} & \cdots & M_{NN}
\end{pmatrix},
\]

(3.3)

the elements of the \( N \times N \) matrix \( M \) are given by

\[
M_{jk} = \frac{e^{-i(θ_k + θ_j)} + c_k^{*} c_k e^{iθ_k + θ_j}}{c_j^{*} - c_k},
\]

\[
θ_k = -iκ_k T - (4iβ|ζ_k|^2 + 2γ|ζ_k|^2)Z,
\]

\[
f(z) = \sqrt{\frac{α_1(z)}{α_2(z)}} Z = -\frac{\sqrt{25}}{12β} ∫ α_1(z)dz, \quad T = \frac{25}{2} (t - (\frac{γ^2}{36β^2} - δ) ∫ α_1(z)dz),
\]

and \( g(t, z) = \frac{6βδ + γ\sqrt{25}}{6β} t - \frac{216δ^2β^3 + 54γβ^2δ\sqrt{25} - γ^3\sqrt{25}}{324β^3} ∫ α_1(z)dz, \)
and the symbol $*$ represents complex conjugate, $\beta$ and $\gamma$ are real numbers.

Note that we can obtain the $N$-solitons from formula Eq. (3.2). As the number $N$ gets larger, the form and dynamic behavior of $N$-solitons are more complex, with more parameters and stronger adjustability. In there, we only consider the case of $N=1$ and 2 as examples to solve the data-driven one-solitons and two-solitons of the VC-Hirota equation.

3.1. Data-driven one-solitons of the VC-Hirota equation. In order to get the one-soliton solution, we set $N=1$ and $\xi_1 = \xi + i\eta$ in the above formula (3.2), the one-soliton solution of VC-Hirota equation is shown as

$$q_1 = 2\sqrt{\frac{\alpha_1(z)}{\alpha_2(z)}} \eta e^{\frac{\alpha_1(z)}{\alpha_2(z)}} \text{sech}(B),$$

$$A = (\xi + \frac{\gamma}{6})(216 i \sqrt{25}(\xi^2 \beta^2 - 3 \eta^2 \beta^2 + \frac{\gamma \beta \xi}{3} + \frac{\gamma^2}{30}) - 324 i \sqrt{25} \delta^2 - 216 i \beta^3 \delta^2) \int \alpha_1(z) \, dz$$

$$- 324 i \beta^3 (\sqrt{25}(\xi + \frac{\gamma}{6}) + \beta \delta) t,$$

$$B = \eta \sqrt{25}(\frac{2}{3} \eta^2 - 2 \xi^2 + \delta - \frac{2\gamma \xi}{3\beta} - \frac{\gamma^2}{18\beta^2}) \int \alpha_1(z) \, dz + t).$$

For simplicity, we take $\gamma = 0$ in the following unless otherwise stated.

- Data-driven one-soliton solution I

Taking $\alpha_1(z) = \alpha_2(z) = z$ and $\xi = \eta = 1$, then the center trajectory equation of the solution (3.5) is $t = \frac{1}{2}(\frac{\beta}{\alpha} - \delta) z^2$. It shows that the value of $\delta$ has a great influence on the propagation path of the solution. In order to adjust the curvature of the curve in order to get better training results, here we take $\delta = 10$, then the explicit one-soliton solution is as follows:

$$q_{11}(t, z) = \frac{4e^{\frac{z^2}{2}}((6-6t)+(26-34i)t) - \frac{1}{16}\frac{1}{2}i z^2-10it}{e^{\frac{z^2}{2}}\sqrt{5}(13i+3i) + 1}.$$

Then we will focus on the data-driven one-solitons with the initial conditions $q_0(t)$ and Dirichlet boundary conditions $q^b(z)$ and $q^{ab}(z)$ of the VC-Hirota equation. Taking $[L_0, L_1] = [-3.0, 1.0]$ and $[T_0, T_1] = [-1.0, 1.0]$ in Eq. (3.1), then the corresponding initial condition and Dirichlet boundary condition are shown as follows:

$$q_0(t) = q_{11}(t, -1.0), \quad t \in [-3.0, 1.0],$$

$$q^b(z) = q_{11}(-3.0, z), \quad q^{ab}(z) = q_{11}(1.0, z), \quad z \in [-1.0, 1.0].$$

We employ the traditional finite difference scheme on even grids in Matlab to simulate one-solitons which contains the initial data (3.6) and boundary data (3.7) to acquire the original training dataset. Specifically, we divide spatial region $[-3.0, 1.0]$ and temporal region $[-1.0, 1.0]$ into 1000 points, respectively. Accordingly, the one-solitons (3.5) is discretized into 1000 snapshots. Using the Latin Hypercube Sampling method (LHS) [37], a smaller training dataset containing initial-boundary data by randomly extracting $N_q = 1500$ from original dataset and $N_f = 20000$ collocation points is generated. After giving a dataset of initial and boundary points, the data-driven one-soliton solution I have been successfully learned by tuning all learnable parameters of the 9 layers IPINN with 40 neurons per layer and regulating the loss function, in which we utilize the 20000 steps Adam firstly and subsequently use 19108 steps L-BFGS optimizations for minimizing the loss function [47]. The relative $L_2$ errors of the IPINN is $1.168275 \times 10^{-2}$ with the train time of 10732.7923 seconds and 39108 times iterations.

Figs. 2-3 display the deep learning results of data-driven one-solitons of the VC-Hirota equation (3.1) based on the IPINN method related to the initial boundary value problem (3.6) and (3.7). Specifically, the density plots with the corresponding peak scale for diverse dynamics which contain exact dynamics, learned dynamics and error dynamics have been exhibited in detail, and the sectional drawings of exact solitons and their corresponding data-driven solutions at different moments arising from the IPINN are displayed in Fig. 2. From the bottom panels of Fig. 2 one can indicate one-solitons (3.5) propagate from right to left along the $t$-axis, and show that the amplitudes of solitons are constant 2 with the development of time $z$. It can be seen that the predicted solution agrees well with the exact solution. Fig. 3(a) exhibits the three-dimensional plot with contour map on three planes of the predicted one-solitons based on the IPINN. From the three-dimensional plot Fig. 3(a) we can see the center trajectory of $q_{11}(t, z)$ takes the shape of a parabola, which is also intuitively reflected in the $(z, t)$ plane of a soliton prediction graph. The loss function curve figures account for the solitons I have also been given out, where Fig. 3(b) demonstrates that Adam is an optimizer with oscillatory loss function curves, Fig. 3(c) illustrates L-BFGS is an optimization algorithm with linear loss function curves. It is worth mentioning that the $N_q = 1500$ training data points involved in the initial-boundary condition are marked by mediumorchid symbol “$\times$” in the learned density graphs.
For recovering the data-driven one-solitons II of the VC-Hirota equation in the 9-layer IPINN with 40 neurons per layer, we consider the initial value and Dirichlet boundary conditions:

\[ q^0(t) = q_{12}(t, -0.75), \quad t \in [-3.0, 3.0], \]

\[ q^{\text{lb}}(z) = q_{12}(-3.0, z), \quad q^{\text{ub}}(z) = q_{12}(3.0, z), \quad z \in [-0.75, 0.75]. \]

With the aid of Matlab, the sampling points are selected by means of the finite difference method with spatial-temporal region \([-3.0, 3.0] \times [-0.75, 0.75]\), and the one-soliton (3.8) is discretized into 1000 \times 1000 data points which contain initial-boundary value condition (3.9) and (3.10). Then the train dataset is obtained by randomly extracting \(N_q = 1500\) from original initial-boundary value condition dataset and \(N_f = 20000\) collocation points produced via the LHS in spatial-temporal region. Through 20000 Adam iterations and 16150 L-BFGS iterations to optimize loss function \(L(\Theta)\), the data-driven one-soliton has been successfully recovered by using the IPINN method. The relative \(L_2\) error is 1.466678 e\(^{-2}\), and spend 9521.1221 seconds doing 36150 times iterations.
Figs. 4–5 exhibit the corresponding training results stemmed from the IPINN with the initial boundary value problem (3.9) and (3.10). In Fig. 4, the exact, learned and error dynamics density plots have been given out. Fig. 5(a) displays the three-dimensional plots with contour map on three planes of the predicted one-soliton II based on the IPINN. From the $(z, t)$ plane of the soliton prediction graph, we can see its center trajectory takes the shape of a "S". Fig. 5(b) and 5(c) exhibit the loss function curve figures of the data driven one-soliton II arising from the IPINN with the 20000 steps Adam and 16150 steps L-BFGS optimizations on the loss function $\mathcal{L}(\Theta)$, respectively.

Figure 4. (Color online) The density plots and sectional drawings for the one-soliton $q_{12}(t, z)$: The one-solitons $q_{12}(t, z)$ resulted from the IPINN with the randomly chosen initial and boundary points $N_{e} = 1500$ which have been shown by using mediumorchid “×” in learned dynamics, and $N_{f} = 20000$ collocation points in the corresponding spatiotemporal region. The exact, learned and error dynamics density plots for the one-solitons $q_{12}(t, z)$ with five distinct training moments $z = -0.50, -0.25, 0.00, 0.25$ and 0.50 (darkturquoise dashed lines), and the sectional drawings which contain the learned and explicit one-solitons $q_{12}(t, z)$ at the aforementioned five distinct moments.

Figure 5. (Color online) (a) three-dimensional plot and corresponding contour maps of the predicted one-soliton solutions II stemmed from the IPINN. (b) The loss function curve figures of the predicted one-solitons II with 20000 Adam optimization iterations; (c) The loss function curve figures of the predicted one-solitons II with 16150 L-BFGS optimization iterations.

- **one-soliton solution III**

  $|q_{1}|^{2} = 4 \frac{\sin(kz)}{\cosh(kz)} \text{sech}^{2}(\sqrt{25}\left(\frac{1}{4} - \delta \cos(kz) + t\right))$ is obtained when considering periodic functions $\alpha_{1}(z) = \sin(kz)$ as excitation function in (3.4). Now the center trajectory of the solution $q_{1}$ is a cosine wave, where $k$ determines the period and $\delta$ has an influence on the shape of the trajectory. Taking $k = 5$, $\delta = 6$ and $\xi = \eta = 1$, then the explicit one-soliton solution is as follows:

  $$q_{12}(t, z) = \frac{4e^{\frac{i}{25}(\cos(5z)(72i-28-44i)\sqrt{7})-2r(3i-15i)\sqrt{7}}}{e^{-\frac{i}{25}r(14\cos(5z)-15r)}+1}. \quad (3.11)$$
In order to obtain the data-driven one-soliton \( q_{13}(t,z) \) of the VC-Hirota equation by using the 9-layer IPINN with 40 neurons per layer, we are committed to studying the initial value and the Dirichlet boundary conditions in the following.

\[
q_{13}^0(t) = q_{13}(t,-1.0), \quad t \in [-3.0,3.0],
\]

\[
q_{13}^h(t) = q_{13}(-3.0,z), \quad q_{13}^{hb}(t) = q_{13}(3.0,z), \quad z \in [-1.0,1.0].
\]

Using the same method and procedure as above to obtain the original train date and smaller train data set of the data-driven one-soliton III. After that, introducing the dataset of initial and boundary points into the IPINN, and taking 20000 Adam iterations and 21755 L-BFGS iterations to optimize loss function \( \mathcal{L}(\Theta) \), the predicted one-soliton have been successfully learned by tuning all learnable parameters of the IPINN, and the corresponding network achieved relative \( L_2 \) error is \( 1.885151 \times 10^{-2} \) and the total number of iterations is 41755 with the 10671.1341 seconds training time.

Figs. 6-7 provide the training results arising from the IPINN for the one-soliton (3.11) of the VC-Hirota equation with the initial boundary value problem (3.12) and (3.13). In Fig. 6 the exact, learned and error dynamics density plots have been exhibited, it is worth mentioning that the \( N_{q} = 1500 \) training data points involved in the initial-boundary condition are marked by mediumorchid symbol “×” in the learned density plots. Meanwhile, the sectional drawings which include the learned and exact one-soliton (3.11) have been shown at the five distinct moments pointed out in the exact, learned and error dynamics density plots by using darkturquoise dashed lines in the bottom panel of Fig. 6. The three-dimensional plots with contour map on three planes of the predicted one-soliton III based on the IPINN is drawn in Fig. 7(a). From the \((z,t)\) plane of the soliton prediction graph, the center trajectory of one-soliton III takes the shape of cosine wave. Fig. 7(b) and 7(c) respectively, display the loss function curve figures of the one-soliton III arising from the IPINN with the 20000 steps Adam and 21755 steps L-BFGS optimizations.

![Figure 6.](image-url) (Color online) The density plots and sectional drawings for the one-soliton: The one-solitons III resulted from the IPINN with the randomly chosen initial and boundary points \( N_q = 1500 \) which have been shown by using mediumorchid “×” in learned dynamics, and \( N_f = 20000 \) collocation points in the corresponding spatiotemporal region. The exact, learned and error dynamics density plots for the one-solitons with five distinct training moments \( t = -0.67, -0.33, 0.00, 0.33 \) and 0.67 (darkturquoise dashed lines), and the sectional drawings which contain the learned and explicit one-solitons at the aforementioned five distinct moments.

### 3.2. Data-driven two-solitons of the VC-Hirota equation

In the formula for \( N \)-solitons formula (3.2), take \( N = 2, \alpha = 0, e_1 = e_2 = 1, \zeta_1 = 1 + i \) and \( \zeta_2 = 2 + 2i \), then the exact expression of the corresponding two-soliton solution is

\[
q_2(t,z) = \frac{\Delta_1}{\Delta_2},
\]

(3.14)
The original training data is obtained by dividing the spatial region $[-1.5, 1.0]$ into 1000 points and the temporal region $[-0.3, 0.3]$ into 1000 points, the remaining data will be used to obtain training errors by comparing with predicted two-solitons. After that, we generate a smaller training dataset containing initial-boundary data by randomly extracting $N_q = 1600$ from original training dataset and $N_f = 20000$ collocation points produced via LHS in the corresponding spatiotemporal region. Then, the predicted two-solitons 1 have been successfully learned by imposing an 8-hidden-layer IPINN with 50 neurons per layer, and the related loss functions are optimized through 20000 Adam iterations and 29359 L-BFGS iterations. The relative $L_2$ error of the IPINN model is 4.121996e$^{-2}$, the total number of iterations is 49359 with 12990.2661 seconds training time.

Figs. 7 display the training results of the two-soliton $q_{21}(t, z)$ based on the IPINN related to the initial-boundary value problem (3.16) and (3.17) of the VC-Hirota equation. Fig. 8 depicts the exact, learned and error dynamic density plots and sectional drawing at different moments for the exact and prediction two-soliton. The three-dimensional plot and corresponding contour maps of the predicted two-soliton solution 1 is shown in Fig. 9(a). In the IPINN framework, curve plot of the loss function after 20000 Adam optimization iterations is displays in Fig. 9(b) the loss function plot with 26239 L-BFGS optimization iterations is showcases in Fig. 9(c). The shape of the two-soliton solution of variable...
coefficients equation shown in this paper is more complex than that of the rogue wave solution of the traditional constant coefficient equation.

\[ q(t, z) = \Delta(t, z) \]

\[ \Delta = -4e^{-\frac{1}{2} \sqrt{3}(32i-82)z^3-45ti} + (2i+1)e^{-\frac{1}{2} \sqrt{3}(32i-26)z^3-9ti}) + 5e^{\frac{3}{2} \sqrt{3}(2i+108)z^3+(9i+54)t} + 5e^{\frac{3}{2} \sqrt{3}(2i+56)z^3+(9i+36)t} \]

\[ \Delta_{22} = e^{\frac{3}{2} \sqrt{3}(2i+20)z^3+(9i+94)t} + 5e^{\frac{3}{2} \sqrt{3}(2i+56)z^3+(9i+36)t} - 4e^{-\frac{3}{2} \sqrt{3}(5i+3^3-54z^3-27i)} + e^{\frac{3}{2} \sqrt{3}(2i+18)z^3+6it+9t} + e^{\frac{3}{2} \sqrt{3}(2i+9)z^3+6i\phi+9t} \]

\[ q_{22} = \frac{\Delta_{12}}{\Delta_{22}} \]

**two-soliton solution II**

Fix the coefficient \( a_1(z) = a_2(z) = z^2 \) and \( \delta = 10 \), then the explicit express of the two-soliton solution is derived form (3.2) as follows:

\[ q_{22} = \frac{\Delta_{12}}{\Delta_{22}} \]

\[ \Delta_{12} = -4e^{-\frac{1}{2} \sqrt{3}(32i-82)z^3+3^3+(9i-36)t} - (2i+4)e^{-\frac{3}{2} \sqrt{3}(9i-28)z^3+(9i-18)t} + (1-2i)e^{-\frac{3}{2} \sqrt{3}(32i-82)z^3-45ti} \]

\[ \Delta_{22} = e^{\frac{3}{2} \sqrt{3}(2i+108)z^3+(9i+54)t} + 5e^{\frac{3}{2} \sqrt{3}(2i+56)z^3+(9i+36)t} - 4e^{-\frac{3}{2} \sqrt{3}(5i+3^3-54z^3-27i)} + e^{\frac{3}{2} \sqrt{3}(2i+18)z^3+6it+9t} + e^{\frac{3}{2} \sqrt{3}(2i+9)z^3+6i\phi+9t} \]

\[ q_{22} = \frac{\Delta_{12}}{\Delta_{22}} \]

Figure 8. (Color online) The two-solitons I resulted from the IPINN with the randomly chosen initial and boundary points \( N_q = 1500 \) which have been shown by using mediumorchid “×” in learned dynamics, and \( N_f = 20000 \) collocation points in the corresponding spatiotemporal region. The exact, predicted and error dynamics density plots for the two-solitons with five distinct training moments \( z = -0.2, -0.1, 0.0, 0.1 \) and 0.2 (darker turquoise dashed lines), and the sectional drawings which contain the learned and explicit two-solitons at the aforementioned five distinct moments.

Figure 9. (Color online) (a) three-dimensional plot and corresponding contour maps of the predicted two-soliton solutions I stemmed from the IPINN. (b) The loss function curve figures of the two-solitons I with 20000 Adam optimization iterations; (c) The loss function curve figures of the two-solitons I with 26239 L-BFGS optimization iterations.

· two-soliton solution II
Now, we consider the initial-boundary value problem of the VC-Hirota equation for obtaining the data-driven two-soliton II by applying the multilayer IPINN. Similarly, taking \([L_0, L_1]\) and \([T_0, T_1]\) in Eq. (3.1) as \([-2.0, 2.0]\) and \([-0.5, 0.5]\) respectively, then the corresponding initial condition \(q^0(t)\) and Dirichlet boundary conditions \(q^{ib}(z)\) and \(q^{ub}(z)\) are shown in the following:

\[
q^0(t) = q_{22}(t, -3.0), \quad t \in [-2.0, 2.0],
\]

and

\[
q^{ib}(z) = q_{22}(-0.5, z), \quad q^{ub}(z) = q_{22}(2.0, z), \quad z \in [-0.5, 0.5].
\]

Dividing the spatial region \([-2.0, 2.0]\) into 1000 points and the temporal region \([-0.5, 0.5]\) into 1000 points, and setting \(N_q = 1500\) and \(N_f = 20000\). Then, the predicted two-solitons II have been successfully learned by imposing a 8-hidden-layer IPINN with 50 neurons per layer, and the related loss functions are optimized through 20000 Adam iterations and 26239 L-BFGS iterations. The relative errors of the IPINN model are \(1.770931e^{-2}\) and the total number of iterations is 23461 with training time of 11106.1975 seconds.

Figs. 10-11 display the training results of the two-solitons \(q_{22}(t, z)\) based on the IPINN related to the initial-boundary value problem (3.19) and (3.20) of the VC-Hirota equation (3.1). Fig. 10 shows various dynamic density plots for the two-solitons \(q^{lb}(z)\) with five distinct training moments \(z = -0.50, 0.00, 0.25, 0.50\) and sectional drawings which contain the learned and exact \(q(t, z)\) at the aforementioned five distinct moments.

Figure 10. (Color online) The density plots and sectional drawings for the two-solitons: the two-solitons resulted from the IPINN with the randomly chosen initial and boundary points \(N_q = 1500\) which have been shown by using mediumorchid “×” in learned dynamics, and \(N_f = 20000\) collocation points in the corresponding spatiotemporal region. The exact, predicted and error dynamics density plots for the two-solitons \(q_{22}(t, z)\) with five distinct training moments \(z = -0.33, -0.17, 0.00, 0.17\) and \(0.33\) (darkturquoise dashed lines), and the sectional drawings which contain the learned and exact two-solitons \(q_{22}(t, z)\) at the aforementioned five distinct moments.

From the figure above, it can be seen intuitively that the dynamic behavior of the two-soliton solution of the VC-Hirota equation is more complex than that of the traditional rogue wave solutions of the constant coefficients integrable equation. This brings a greater challenge to the deep learning process of two-solitons. In this paper, an 8-layer IPINN algorithm with 50 neurons in each layer is adopted to successfully train the 2-solitons of the VC-Hirota equation after several experiments to optimize the relevant parameters.

3.3. Data-driven higher-order solitons of the VC-Hirota equation. In complex systems, the research of high-order soliton of the IVC model has more appropriate practical significance. In this section, we will focus on the data-driven high-order soliton of the VC-Hirota equation with \(\Theta_1 = 1 + i\) and \(\Theta_2 = 1 - i\) based on the IPINN method. The formula of the high-order solitons for the VC-Hirota equation is obtained by means of Riemann-Hilbert method in [16]. For simple, we just use a second-order soliton as an example to verify the effective of the IPINN method. The express of the second-order soliton is as follows.

\[
q_h = \frac{\Theta_1}{\Theta_2},
\]
\[ \Theta_1 = -8i \sqrt{\frac{\alpha_1(z)}{\alpha_2(z)}} \left( ((-i\sqrt{\sqrt{2} + 4\sqrt{2}t}) \int \alpha_1(z)dz - i\sqrt{2}t + i)e^{(2\sqrt{2}t + \frac{i}{\sqrt{2}}((-8-8i)\sqrt{2})} t \alpha_1(z)dz - (\alpha_2 - 2\sqrt{2}t) + (i\sqrt{2}t) + (i\sqrt{2}t + 4\sqrt{2}t) \int \alpha_1(z)dz + 4\sqrt{2}t) e^{-\frac{i}{\sqrt{2}}((8\sqrt{2}t + 2\alpha_1(z))dz + 3\alpha_1(z))}, \]

\[ \Theta_2 = (8\delta^3 + 128\delta)(\int \alpha_1(z)dz)^2 + 16\delta^2 \int \alpha_1(z)dz + 8t^2 \delta + 2)e^{(1 + \frac{i}{\sqrt{2}})\sqrt{2}t((3\alpha_1(z))dz + 3t)} + e^{(1 + \frac{i}{\sqrt{2}})\sqrt{2}t((3\alpha_1(z))dz + 3t)} + e^{(-\frac{1}{\sqrt{2}} + \frac{i}{\sqrt{2})}((3\alpha_1(z))dz + 3t))}. \]

From the expression of (3.21), we can construct abundant second-order solitons by adjustment \( \alpha_1(z) \) and \( \alpha_2(z) \). Next, we give an expression of the second-order soliton with \( \alpha_1 = \alpha_2 = \sinh(z) \) as follows:

\[ q_{h1}(x,t) = \frac{\Theta_1}{\Theta_2}, \]  

(3.22)

\[ \Theta_{h1} = 8i((5 \cosh(z) + t)i\sqrt{10}e^{(\frac{3}{\sqrt{2}} \sqrt{10} - \frac{3}{2} \sqrt{10} - \frac{3}{2} t) \cos(z) + 2\sqrt{10}t - 5st} - (5 \cosh(z) + 1)i \sqrt{10}e^{-(1 + \frac{i}{\sqrt{2}}((8\sqrt{10} + 50) \cosh(z) + 15t)} \]

\[ - (4 \cosh(z) i\sqrt{10} + i)e^{-\frac{i}{\sqrt{2}}((8\sqrt{10} + 50) \cosh(z) + 15t)} + e^{(\frac{3}{\sqrt{2}} \sqrt{10} - \frac{3}{2} \sqrt{10} - \frac{3}{2} t) \cosh(z) + 2\sqrt{10}t - 5st}), \]

\[ \Theta_{h2} = (1640 \cosh(z)^2 + 400t \cosh(z) + 40t^2 + 2)e^{\frac{2\pi}{\sqrt{10}}((1 + t) \cos(z) + 3t)} + e^{\frac{2\pi}{\sqrt{10}}(11 \cos(z) + 3t) + (1 + t)} + e^{\frac{2\pi}{\sqrt{10}}(11 \cos(z) + 3t)(1 + 1)}}. \]

In order to recover the data-driven second-order soliton solution via eq. (3.22), we commit to introducing the initial boundary value conditions of the VC-Hirota equation to the 9-layer PINN with 40 neurons per layer. Selecting \([L_0, L_1]\) and \([T_0, T_1]\) in Eq. (3.1) as \([-10, 0.0]\) and \([-0.75, 0.75]\) respectively, then we have the corresponding initial value and Dirichlet boundary conditions:

\[ q^0(t) = q_{h1}(t, -0.75), \quad t \in [-10, 0.0], \]  

(3.23)

\[ q^{0}(z) = q_{h1}(-10.0, z), \quad q^{0b}(z) = q_{h1}(0.0, z), \quad z \in [-0.75, 0.75]. \]  

(3.24)

In order to obtain the original training data set of the above initial boundary value conditions (3.23) and (3.24), we discretize the exact second-order soliton solution (3.22) based on the finite difference method by dividing the spatial region \([-10, 0.0]\) into 1000 points and the temporal region \([-0.75, 0.75]\) into 1000 points in Matlab. Furthermore, in addition to the data set composed of the aforementioned initial boundary value conditions, the residual data set is used to calculate the \( L_2 \) norm error by comparing with the predicted second-order soliton solution (3.22). After that, a smaller training dataset that containing initial-boundary data is generated by randomly extracting \( N_y = 1500 \) from original dataset and \( N_f = 20000 \) collocation points which are produced by the LHS. According to 20000 Adam iterations and 25117 L-BFGS iterations, the latent second-order soliton \( q_{h1}(t, z) \) have been successfully learned by employing the IPNN, and the network achieved relative \( L_2 \) error of 1.622956e-2 for the second-order soliton \( q_{h1}(t, z) \), and the total number of iterations is 20932 with the training time is 11016.8943 seconds.

Figs. (12)(13) present the deep learning results of the second-order soliton based on the PINN for the VC-Hirota equation with the initial-boundary value problem (3.23) and (3.24). Fig. (12) displays the exact, learned and error dynamics of the second-order soliton, and exhibits the sectional drawings which contain the learned and explicit...
second-order soliton solution at five different moments. From the density plots of learned dynamics and profiles which reveal amplitude and error of exact and prediction second-order soliton in Fig. [12]. The 3D plots of the predicted second-order soliton is shown in Fig. [13(a)]. Graphically, it can be seen that the dynamic behavior of second-order solution is similar to two parallel one solitons \( q(t, z) \) with different amplitudes. The loss function curve figures of the second-order soliton solutions with 20000 Adam optimization iterations is shown in Fig. [13(b)]. The loss function curve figures of the second-order soliton solutions with 25117 L-BFGS optimization iterations is displayed in Fig. [13(c)].

![Figure 12](image1.png)

**Figure 12.** (Color online) The density plots and sectional drawings for the second-order soliton \( q_h(t, z) \): The second-order soliton \( q_h(t, z) \) resulted from the IPINN with the randomly chosen initial and boundary points \( N_q = 1500 \) which have been shown by using mediumorchid “ × ” in learned dynamics, and \( N_f = 20000 \) collocation points in the corresponding spatiotemporal region. The exact, learned and error dynamics density plots for the second-order soliton \( q_h(t, z) \) with five distinct training moments \( t = -0.50, -0.25, 0.00, 0.25 \) and 0.50 (darkturquoise dashed lines), and the sectional drawings which contain the learned and explicit second-order soliton \( q_h(t, z) \) at the aforementioned five distinct moments.

![Figure 13](image2.png)

**Figure 13.** (Color online) (a) three-dimensional plot and corresponding contour maps of the predicted second-order soliton stemmed from the IPINN. (b) The loss function curve figures of the second-order soliton solutions with 20000 Adam optimization iterations; (c) The loss function curve figures of the second-order soliton solutions with 25117 L-BFGS optimization iterations.

Through a lot of numerical training, we found that the IPINN algorithm has a good effect when describing relatively smooth solutions in a small range, but when the dynamic behavior of the solution becomes complex, the learning effect will become worse. In order to obtain better numerical results, it becomes extremely dependent on the choice of parameters, which means that the training difficulty will increase with the increase of the complexity of the solution. A summary of the aforementioned data-driven forward problems of the VC-Hirota equation is shown in the following Tab. [2]
Table 2. Results of the different forward problems for the VC-Hirota equation by IPINN model

| Results | solution Types | one-soliton I | one-soliton II | one-soliton III | two-soliton I | two-soliton II | second-order soliton |
|---------|----------------|---------------|---------------|----------------|---------------|----------------|----------------------|
| Relative error | 1.169275e-04 | 1.466678e-04 | 1.885154e-04 | 4.121996e-04 | 1.770901e-04 | 1.622268e-04 |                         |
| Training time | 10732.923s | 9521.1221s | 10671.1341s | 12990.2661s | 11061.1975s | 11106.8943s |                         |
| Iterations | 39108 | 36150 | 41755 | 49359 | 41755 | 40042 |                         |

4. Data-driven inverse problems of the VC-Hirota equation

In the inverse problem of VC-Hirota equation, we mainly consider two cases: first, data-driven parameters discovery of VC-Hirota equation by using IPINN system when the variable coefficient \( \alpha(z) \) of VC-Hirota equation are known; Second, the data-driven variable coefficient equation discovery, namely, data-driven function discovery.

4.1. Data-driven parameters discovery of the VC-Hirota equation. The inverse problems of the VC-Hirota equation with unknown parameters \( \delta \) is considered by using the IPINN model in this part. First, initialize the unknown parameters \( \delta \) to \( \delta_0 = 1 \), then we can get the predicted parameter \( \delta \) from the IPINN algorithm. Define the relative error of unknown parameters as

\[
RE = \frac{|\delta - \hat{\delta}|}{\delta} \times 100\%,
\]

where the \( \hat{\delta} \) and \( \delta \) represent predicted value and true value, respectively. All noise interference in this section is added to the randomly chosen small data set, the specific form is as below:

\[
Data_{train} = Data_{train} + noise \times np.std(Data_{train}) \times np.random.randn(Data_{train}.shape[0], Data_{train}.shape[1]),
\]

where \( noise \) and \( Data_{train} \) indicate the noise intensity and a small randomly chosen training data set, respectively. The \( np.std(\cdot) \) returns the standard deviation of an array element, and \( np.random.randn(\cdot, \cdot) \) returns a set of samples with a standard normal distribution.

In order to accurately and effectively learn unknown parameters, we try to use \( L^2 \) norm parameter regularization into IPINN to study the parameter discovery problem of VC-Hirota equation. Now we construct a new loss function \( \mathcal{L}(\Theta) \) with \( L^2 \) norm parameter weight decay as follows [34]

\[
\mathcal{L}(\Theta) = Loss_{PR} = \mathcal{L}(\Theta) + \frac{\alpha}{2} \|W\|_2^2,
\]

where \( \mathcal{L}(\Theta) \) and \( W \) have been defined in Eq. (4.7) and Eq. (4.1).

For learning the parameters \( \delta \) in Eq. (4.3) with the aid of one-solitons solution (3.5), we reset the initial conditions and Dirichlet boundary conditions as follows:

\[
q_0^0(t) = q_{11}(t, -0.5), \quad t \in [-3.0, 1.0],
\]

\[
q_{1h}(t) = q_{11}(-3.0, z), \quad q_{2h}(t) = q_{11}(1.0, z), \quad z \in [-0.5, 0.5].
\]

The original data set can be obtained by dividing the spatial region \([-3.0, 1.0]\) into 1000 points and temporal region \([-0.5, 0.5]\) into 1000 points. Then a smaller training dataset that containing initial-boundary data [4.3] and [4.4] by randomly extracting \( N_q = 1700 \) from original dataset and \( N_f = 20000 \) collocation points which is generated by the LHS method.

When we set \( \alpha = 0 \), namely, recover the unknown parameter \( \delta \) under the IPINN without \( L^2 \) norm parameter regularization. We find that the parameter \( \delta \) error learned under the IPINN without \( L^2 \) norm parameter regularization is not very satisfactory by tuning various parameters. As an example here we only show a relatively good result in the following. The training results of unknown parameters \( \delta \) under the above initial boundary value conditions by using the IPINN with 30000 Adam iterations are shown in Fig. [14]. Fig. [13 (a)] exhibit the variation curves of unknown coefficients \( \delta \) with different noise intensity. When we use the clean initial-boundary data (noise=0%) in IPINN, the relative error of \( \delta \) is about 7.821617%. With the increase of noise, the relative error of unknown parameter values also increases. When the noise is 3%, the relative error reaches 11.8413753%. The error variation plots of unknown coefficients \( \delta \) under different interference noise are revealed in Fig. [13 (b)].

Recall the smaller training dataset that containing initial-boundary data [4.3] and [4.4] by randomly extracting \( N_q = 1700 \) from the original dataset and \( N_f = 20000 \) collocation points which generated by the LHS method. We successfully used IPINN method with parameter regularization to learn the value of the unknown parameter \( \alpha \) with a small relative error by adjusting the iteration number of Adam, increasing or decreasing the weight \( \alpha \) and other tuning methods. When we set the 30000 times Adam iterations, Fig. [15 (a)] displays the variation curves of unknown coefficients \( \alpha \) with different noise intensity by using the IPINN with \( \alpha = 0.00005 \) weight decay. The noise intensity and relative error plots are shown in Fig. [15 (b)]. The specific numerical results show that \( \alpha \) obtains the minimum relative error of 0.991810% as the noise intensity is 0%, and maximum relative error 6.2832007% as the noise intensity is 3%. Compared with the numerical results obtained by IPINN without parameter regularization in Fig. [14] the relative
Figure 14. (Color online) Training results of parameter discovered by means of the IPINN with 30000 Adam iterations: (a) the variation curves of unknown coefficients $\delta$ with different noise intensity; (b) unknown coefficients $\delta$ error variation plots under different interference noise.

In other words, the training effect has been very good in the case of clean data. However, when using the data with noise interference for training, the relative error of the prediction parameters is larger than that of the training results of clean data. Therefore, we need to reset the weight decay coefficient to improve the training effect of using the data with noise interference. When we put the L-BFGS iterations into the above method, one can find that the training effect with data of various noise is much better than that with clean data. Compare with the Adam iteration without L-BFGS iterations, we find that the parameter discovery result is worse in the pure data, but significantly better in the 3% noise, the generalization training shows that IPINN with coefficient $\alpha = 0.00005$ of weight decay has excellent noise resistance. The detail can be seen in the Fig. 16.

At the same time, we also consider the influence of weight $\alpha$ and the number of Adam iterations on the parameter discovery results, and the specific results can be seen in the following Fig. 17. From the above figures, we can find that the training effect is better once setting the appropriate weight coefficients, noise intensity data and other associated parameters.
Figure 16. (Color online) Training results of parameter discovered by means of the IPINN with $\alpha = 0.00005$ weight decay: (a) the variation curves of unknown coefficients $\delta$ with different noise intensity; (b) unknown coefficients $\delta$ error variation plots under different interference noise.

Figure 17. (Color online) Training results of parameter discovered by means of the IPINN with different parameters: (a) with a fixed number of 30000 Adam iterations, unknown coefficients $\delta$ error variation plots under different weight decay; (b) with a fixed weight decay $\alpha = 0.00005$, the variation curves of unknown coefficients $\delta$ with different iteration numbers of Adam; (c) the unknown coefficients $\delta$ error variation plots under different interference noise after using only 30000 Adam iterations and adding L-BFGS algorithm.
4.2. Data-driven function discovery of the VC-Hirota equation. In this subsection, we will use a PINN algorithm with sub-neural networks to study the unknown function for the VC-Hirota equation. The flow chart of the PINN algorithm with sub-neural networks is given in Fig. 18 to describe it more intuitively.

Figure 18. (Color online) Schematic of the PINN algorithm with sub-neural networks for study the unknown function of VC-Hirota equation.

Compared with the flow chart of the algorithm in the NNs part, we not only use a main neural network to train the data-driven solutions $u$ and $v$, but also add a sub-neural network to train the unknown functions $\alpha_1(z)$, $\alpha_2(z)$ and $\alpha_3(z)$ in the equation (1.1). At this point, $\text{Loss}_{\alpha_1(z)}$, $\text{Loss}_{\alpha_2(z)}$ and $\text{Loss}_{\alpha_3(z)}$ are added to the loss function $L(\bar{\Theta})$ where $\text{Loss}_{\alpha_i(z)}$ are defined as below:

$$\text{Loss}_{\alpha_i(z)} = \left| \hat{\alpha_i}(T_0) - \alpha_i(T_0) \right|^2,$$

where $\alpha_i(T_0)$ and $\hat{\alpha_i}(T_0)$ (i=1, 2 and 3) represent predicted value and true value, respectively.

Without loss of generality, we can consider a simple case: $\alpha_1(z) = \alpha_2(z) = \Gamma(z)$ and $\delta = 10$ in the VC-Hirota equation (1.1), then $\alpha_3(z) = 0$ and the physics-informed parts of the PINN algorithm with sub-neural networks for VC-Hirota equation (1.1) can be rewritten as:

$$f_u := -v + \Gamma(z) v_{tt} + \frac{1}{30} \Gamma(z) v_{ttt} + 10 \Gamma(z)(u^3 + uv^2) + \Gamma(z)(v^2 v_t + v^2 v_t),$$

$$f_v := u + \Gamma(z) v_{tt} - \frac{1}{30} \Gamma(z) u_{ttt} + 10 \Gamma(z)(uv^2 + v^3) - \Gamma(z)(u^2 u_t + u^2 u_t).$$

Now $\text{Loss}_{\Gamma(z)} = \text{Loss}_{\alpha_1(z)} = \text{Loss}_{\alpha_2(z)} = \left| \hat{\Gamma}(T_0) - \Gamma(T_0) \right|^2$.

In this section, we fixed $d = 1$, which has the advantage of ensuring good training results while training fewer parameters. Without loss of generality, we can take $N_{\text{in}} = 0$. Now the loss function which used in the PINNs is shown as follows:

$$\mathcal{L}(\bar{\Theta}) = \text{Loss} = \text{Loss}_u + \text{Loss}_v + \text{Loss}_{f_u} + \text{Loss}_{f_v} + \text{Loss}_{\Gamma(z)},$$

and rewritten the $\text{Loss}_u$ and $\text{Loss}_v$, as

$$\text{Loss}_u = \frac{1}{N_{\text{in}}} \sum_{j=1}^{N_{\text{in}}} \left| \hat{u}(t^j, z^j) - u^j \right|^2,$$

$$\text{Loss}_v = \frac{1}{N_{\text{in}}} \sum_{j=1}^{N_{\text{in}}} \left| \hat{v}(t^j, z^j) - v^j \right|^2,$$

where $N_{\text{in}}$ represents the internal grid point from original dataset.

For learning the unknown function $\Gamma(z)$, $z \in [z_0, z_1] = [-1.0, 1.0]$ in Eq. (2.8) with the aid of one-solitons solution (3.5), we reset the initial conditions as follows:
where $c = 1$. The temporal region $[-1.0, 1.0]$ of function $\Gamma(z)$ is divided into 500 points. In the main neural network, the original data set can be obtained by dividing the spatial region $[-3.0, 1.0]$ into 512 points and temporal region $[-1.0, 1.0]$ into 200 points. Then a smaller training dataset is obtained by randomly extracting $N_{q} = 2000$ and $N_{f} = 40000$ which is generated by the random sampling method.

Define $L_2$ norm error of unknown function as:

$$\text{Error} = \frac{\sqrt{\sum_{k=1}^{N} |\hat{\Gamma}(z_k) - \Gamma(z_k)|^2}}{\sqrt{\sum_{k=1}^{N} |\Gamma(z_k)|^2}},$$

where $\hat{\Gamma}(z_k)$ and $\Gamma(z_k)$ represent predicted value and true value, respectively. $z_k$ denotes the equidistant point after dividing the interval $[-1, 1]$ into 500 equal parts.

The training results of unknown function $\Gamma(z)$ under the above conditions by using the PINN algorithm with sub-neural networks with 5000 times Adam iterations and a number of L-BFGS iterations are shown in Fig. 19. Since unknown function $\Gamma(z)$ can obtain very good training results under both pure data and noisy conditions, the images of predictive solutions under different noises are visually indistinguishable. Therefore, in Fig.19(a) we only present the images of predictive function solution and real function solution $\Gamma(z)$ under pure data. When we use the clean data (noise= 0%) in the PINN algorithm with sub-neural networks, the $L_2$ norm error of $\Gamma(z)$ is about $1.124902e^{-4}$. With the addition of different noise, we find that the PINN algorithm with sub-neural networks has excellent anti-noise ability. For example, the $L_2$ norm error of $\Gamma(z)$ is about $1.449694e^{-4}$ when we add 3% noise into the data. Error variation plots of unknown function $\Gamma(z)$ under different interference noise is revealed in Fig.19(b). In error variation plots Fig.19(b) we convert scientific counting to percentages.

**Figure 19.** (Color online) Training results of function discovery by means of the PINN algorithm with sub-neural networks: (a) the plot of predicted and true function $\Gamma(z)$; (b) error variation plots of predicted function $\Gamma(z)$ under different interference noise.

5. Conclusions and discussions

The conclusion drawn from the numerical evidence presented in this paper is the IPINN framework with neuron-wise locally adaptive activation function and slope recovery term is an efficient method for solving the data-driven solutions including the one-solitons, two-solitons and second-order soliton of the VC-Hirota equation with a small sample data set. Meanwhile, we also study the parameter discovery problem of VC-Hirota equation through the IPINN method and found that the PINN with neuron-wise locally adaptive activation function and $L^2$ norm parameter regularization shows amazing effect in studying the inverse problem of VC-Hirota equation. Furthermore, we learn the data-driven unknown function discovery stably and efficiently in the variable coefficient equation via the PINN algorithm with sub-neural networks, and find that the network has superior anti-noise ability. The data-driven forward and inverse problems of variable coefficients equation are given for the first time by using the PINN methods mentioned above in this work, which provides an essential theoretical basis and experimental experience for variable coefficients model to conduct more professional research.
Due to the soliton solutions of the variable coefficient equation present more complex shapes including parabola type, “S-shape” and cosine wave type etc., which brings more challenges in the training process. The IPINN method displays well training effect and convergence speed in training the one-soliton solution of VC-Hirota equation. Even though the two-soliton solution of variable coefficients equation, the dynamic behavior shown in this paper is more complex than that of the rogue wave solution of the traditional constant coefficient equation, we also obtain the data driven two-solitons of the VC-Hirota successful through a large number of training experiments. The experiments result that in addition to the number of network layers and neurons, other influencing factors, such as randomly extracting from original dataset and collocation points may have a great impact on the learning results. During the parameter discovery, we find that the training error is not ideal either using clean data or noisy data based on earlier IPINN method. However, the training results improve greatly under different parameters when the parameter regularization strategy is added into the IPINN framework. In the progress of function discovery, we can obtain the data-driven function solution with relatively small error whether using clean data or data with different levels of noise by means of the IPINN algorithm with sub-neural networks.

The IPINN algorithm has good effect when describing relatively smooth solutions in a small range, but as the solution range becomes larger or the dynamic behavior of the solution is steeper, the learning effect becomes worse. In future work, we will focus on overcoming this problem by optimization and improvement of the PINN algorithms. Since the variable coefficient model used in this paper is relatively complex, in the process of data-driven function discovery in this paper, we only give the example of linear function. In future work, we will also consider how to construct an appropriate PINN network to solve the data-driven problem of nonlinear unknown function.

REFERENCES

[1] Zabusky NJ, Kruskal MD. Interaction of "solitons" in a collisionless plasma and the recurrence of initial states. Phys Rev Lett. 1965;15:240-3.
[2] Hasegawa A, Tappert F. Transmission of stationary nonlinear optical pulses in dispersive dielectric fibres I. Anomalous dispersion. Appl Phys Lett. 1973;23:142-4.
[3] Gedalin M, Scott TC, Band YB. Optical solitary waves in the higher order nonlinear Schrödinger equation, Phys Rev Lett. 1997;78:448-51.
[4] Palacios SL, Guinea A, Fernández-Díaz JM, Crespo RD. Dark solitary waves in the nonlinear Schrödinger equation with third order dispersion, self-steepening, and self-frequency shift. Phys Rev E. 1999;60:45-7.
[5] Khaykovich L, Schreck F, Ferrari G, Bourdel T, Cubizolles J, Carr LD, Castin Y, Salomon C. Formation of a Matter-Wave Bright Soliton. Science. 2002;296:1290-3.
[6] Grelu Y, Akhmediev N. Dissipative solitons for mode-locked lasers. Nature photonics. 2012;6:84-92.
[7] Kodama Y. Optical Solitons in a Monomode Fiber. J Stat Phys. 1985;39:597-613.
[8] Papaioannou E, Frantzeskakis DJ, Hizanidis K. An analytical treatment of the effect of axial inhomogeneity on femtosecond solitary waves near the zero dispersion point. IEEE J Quantum Electron. 1996;32:145-54.
[9] Yang RC, Li L, Hao RY, Li ZH, Zhou GS. Combined solitary wave solutions for the inhomogeneous higher-order nonlinear Schrödinger equation. Phys Rev E. 2005;71:036616.
[10] Dai CQ, Zhang JF. New solitons for the Hirota equation and generalized higher-order nonlinear Schrödinger equation with variable coefficients. J Phys A: Math Gen. 2006;39:723-37.
[11] Wang P, Tian B, Liu WJ, Li M, Sun K. Soliton Solutions for a Generalized Inhomogeneous Variable-Coefficient Hirota Equation with Symbolic Computation. Stud Appl Math. 2010;125:213-22.
[12] He JS, Tao YS, Porsezian K, Fokas AS. Rogue wave management in an inhomogeneous nonlinear fibre with higher order effects. J Nonlinear Math Phy. 2013;20:407-19.
[13] Rajan M, Mahalingam A. Nonautonomous solitons in modified inhomogeneous Hirota equation: soliton control and soliton interaction. Nonlinear Dyn. 2015;79:2469-84.
[14] Gao XY. Looking at a nonlinear inhomogeneous optical fiber through the generalized higher-order variable-coefficient Hirota equation. Appl Math Lett. 2017;73:143-9.
[15] Yang DY, Tian B, Hu CC, Liu SH, Shan WR, Jiang Y. Conservation laws and breather-to-soliton transition for a variable-coefficient modified Hirota equation in an inhomogeneous optical fiber. Wave Random Complex. 2021;1:1-17.
[16] Zhou HJ, Chen Y. High-order soliton solutions and their dynamics in the inhomogeneous variable coefficients Hirota equation. arXiv: 2207.06248v1 (2022)
[17] Raisi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686-707.
[18] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang SF, Yang L. Physics-informed machine learning. Nat Rev Phys. 2021;3:422-40.
[19] Jagtap AD, Kawaguchi K, Karniadakis GE. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. J Comput Phys. 2020;404:109136.
[20] Jagtap AD, Kawaguchi K, Karniadakis GE. Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks. Proc R Soc A. 2020;476:20200334.
[21] Li J, Chen Y. Solving second-order nonlinear evolution partial differential equations using deep learning. Commun Theor Phys. 2020;72:105005.
[22] Li J, Chen Y. A deep learning method for solving third-order nonlinear evolution equations. Commun Theor Phys. 2020;72:115003.
[23] Pu JC, Li J, Chen Y. Soliton, breather and rogue wave solutions for solving the nonlinear Schrödinger equation using a deep learning method with physical constraints. Chin Phys B. 2021;30:060202.
[24] Peng WQ, Pu JC, Chen Y. PINN deep learning for the Chen-Lee-Liu equation: rogue wave on the periodic background. Commun Nonlinear Sci Numer Simul. 2022;105:106067.
[25] Miao ZW, Chen Y. Physics-informed neural network method in high-dimensional integrable systems. Mode Phys Lett B. 2021;36:2150531.
[26] Lin SN, Chen Y. A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions. J. Comput. Phys. 2022;457:111053.

[27] Fang Y, Wu GZ, Wang YY, Dai CQ. Data-driven femtosecond optical soliton excitations and parameters discovery of the high-order NLSE using the PINN. Nonlinear Dyn. 2021;105:603-16.

[28] Wang L, Yan ZY. Data-driven rogue waves and parameter discovery in the defocusing nonlinear Schrödinger equation with a potential using the PINN deep learning. Phys Lett A. 2021;404:127-108.

[29] Wu GZ, Fang Y, Wang YY, Wu GC, Dai CQ. Predicting the dynamic process and model parameters of the vector optical solitons in birefringent fibers via the modified PINN. Chaos, Solitons and Fractals. 2021;152:111393.

[30] Mo YF, Ling LM, Zeng DL. Data-driven vector soliton solutions of coupled nonlinear Schrödinger equation using a deep learning algorithm. Phys Lett A. 2022;421:127739.

[31] Li JH, Chen JC, Li B. Gradient-optimized physics-informed neural networks (GOPINNs): a deep learning method for solving the complex modified KdV equation. Nonlinear Dyn. 2022;107:781-92.

[32] Zhou ZJ, Yan ZY. Solving forward and inverse problems of the logarithmic nonlinear Schrödinger equation with symmetric harmonic potential via deep learning. Phys Lett A. 2021;387:127010.

[33] Pu JC, Li J, Chen Y. Solving localized wave solutions of the derivative nonlinear Schrödinger equation using an improved PINN method. Nonlinear Dyn. 2021;105:1723-39.

[34] Pu JC, Chen Y. Data-driven vector localized waves and parameters discovery for Manakov system using deep learning approach. Chaos, Solitons and Fractals. 2022;160:112182.

[35] Kingma DP, Ba JL. Adam: a method for stochastic optimization. arXiv: 1412.6980 (2014)

[36] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Program. 1989;45:503-28.

[37] Stein M. Large sample properties of simulations using Latin hypercube sampling. Technometrics. 1987;29:143-51.

(HZ) SCHOOL OF MATHEMATICAL SCIENCES, SHANGHAI KEY LABORATORY OF PURE MATHEMATICS AND MATHEMATICAL PRACTICE, AND SHANGHAI KEY LABORATORY OF TRUSTWORTHY COMPUTING, EAST CHINA NORMAL UNIVERSITY, SHANGHAI 200241, PEOPLE’S REPUBLIC OF CHINA

(YC) SCHOOL OF MATHEMATICAL SCIENCES, SHANGHAI KEY LABORATORY OF PURE MATHEMATICS AND MATHEMATICAL PRACTICE, AND SHANGHAI KEY LABORATORY OF TRUSTWORTHY COMPUTING, EAST CHINA NORMAL UNIVERSITY, SHANGHAI 200241, PEOPLE’S REPUBLIC OF CHINA

(YC) COLLEGE OF MATHEMATICS AND SYSTEMS SCIENCE, SHANDONG UNIVERSITY OF SCIENCE AND TECHNOLOGY, QINGDAO 266590, PEOPLE’S REPUBLIC OF CHINA

Email address: ychen@sei.ecnu.edu.cn