INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 19, Number 5, Pages 603–629

COMPARATIVE STUDIES ON MESH-FREE DEEP NEURAL NETWORK APPROACH VERSUS FINITE ELEMENT METHOD FOR SOLVING COUPLED NONLINEAR HYPERBOLIC/WAVE EQUATIONS

XINGWEN ZHU, MINGYAN HE, AND PENGTAO SUN*

Abstract. In this paper, both the finite element method (FEM) and the mesh-free deep neural network (DNN) approach are studied in a comparative fashion for solving two types of coupled nonlinear hyperbolic/wave partial differential equations (PDEs) in a space of high dimension $\mathbb{R}^d$ $(d > 1)$, where the first PDE system to be studied is the coupled nonlinear Korteweg-De Vries (KdV) equations modeling the solitary wave and waves on shallow water surfaces, and the second PDE system is the coupled nonlinear Klein-Gordon (KG) equations modeling solitons as well as solitary waves. A fully connected, feedforward, multi-layer, mesh-free DNN approach is developed for both coupled nonlinear PDEs by reformulating each PDE model as a least-squares (LS) problem based upon DNN-approximated solutions and then optimizing the LS problem using a $(d+1)$-dimensional space-time sample point (training) set. Mathematically, both coupled nonlinear hyperbolic problems own significant differences in their respective PDE theories; numerically, they are approximated by virtue of a fully connected, feedforward DNN structure in a uniform fashion. As a contrast, a distinct and sophisticated FEM is developed for each coupled nonlinear hyperbolic system, respectively, by means of the Galerkin approximation in space and the finite difference scheme in time to account for different characteristics of each hyperbolic PDE system. Overall, comparing with the subtly developed, problem-dependent FEM, the proposed mesh-free DNN method can be uniformly developed for both coupled nonlinear hyperbolic systems with ease and without a need of mesh generation, though, the FEM can produce a concrete convergence order with respect to the mesh size and the time step size, and can even preserve the total energy for KG equations, whereas the DNN approach cannot show a definite convergence pattern in terms of parameters of the adopted DNN structure but only a universal approximation property indicated by a relatively small error that rarely changes in magnitude, let alone the dissipation of DNN-approximated energy for KG equations. Both approaches have their respective pros and cons, which are also validated in numerical experiments by comparing convergent accuracies of the developed FEMs and approximation performances of the proposed mesh-free DNN method for both hyperbolic/wave equations based upon different types of discretization parameters changing in doubling, and specifically, comparing discrete energies obtained from both approaches for KG equations.

Key words. Coupled hyperbolic/wave equations, Korteweg-De Vries (KdV) equations, Klein-Gordon (KG) equations, deep neural network (DNN), finite element method (FEM), space-time sample points (training) set, least-squares (LS), convergence accuracy, energy conservation.

1. Introduction

In this paper, we choose the following two types of coupled nonlinear hyperbolic/wave partial differential equations (PDEs) as two model problems to be studied, numerically: Korteweg-De Vries (KdV) equations and Klein-Gordon (KG) equations, both are coupled hyperbolic system defined in a space of high dimension $\mathbb{R}^d$ $(d > 1)$ and in a one-dimensional time interval $[0, T]$. The KdV equation was

Received by the editors September 16, 2021 and, in revised form, March 24, 2022; accepted May 27, 2022.

2000 *Mathematics Subject Classification.* 65M60, 65N30, 35L57, 35L75, 35L70, 68Q32, 35Q93.

*Corresponding author.

first introduced in 1985 by Korteweg and de Vries for a (1+1)-dimensional case, later the (d+1)-dimensional cases of coupled KdV equations were developed to explain more involved nonlinear phenomena [24, 39, 40, 46, 69, 51]. The KdV equation(s) is a very important hyperbolic PDE model, both mathematically and practically, for the description of small amplitude shallow-water waves with weakly nonlinear restoring forces, long internal waves in a density-stratified ocean, ion acoustic waves in a plasma, solitary waves on the intensity of light in optical fibers, acoustic waves on a crystal lattice, and fluctuation phenomena in biological and physical systems [9, 35, 64]. As for the second hyperbolic problem to be studied in this paper, KG equations, plays a significant role in many scientific applications as well, such as in studying solitons and condensed matter physics [6, 55, 8, 4, 1], in investigating the interaction of solitons in a collisionless plasma [31, 42], and in examining the recurrence of initial states and the nonlinear wave equations [15, 62, 61, 60, 45].

Both KdV and KG equations are targeted together in this paper because not only their numerical studies are of great scientific significance and research interests but also: (1) both of them fall into the same category of nonlinear hyperbolic/wave problems, in general; (2) each one of them owns respective special properties in PDE theories, in particular; (3) a stable, convergent and/or energy-preserving finite element method (FEM) for each PDE model is uneasy and subtle to be developed. While a comprehensive numerical analysis for each one of these two PDE systems is still a hot topic even now and somehow challenging [3, 67, 19, 33, 30, 34, 23, 11, 28], in this paper we are dedicated to developing FEMs and recently emerging mesh-free, deep neural network (DNN) approaches for both coupled nonlinear hyperbolic/wave problems, respectively, to comparing numerical complexities and convergence properties of both FEM and DNN approach, and finally, to reaching a comparative conclusion based on these two numerical approaches for two distinct problems that belong to the same kind of PDEs, largely.

DNN has been demonstrated as a powerful tool to conquer the curse of dimensionality [16, 18, 25, 63], and have been applied to solve PDEs, e.g., the deep BSDE method [20, 27], the deep Galerkin method (DGM) [59], the physics-informed neural networks (PINN) [48, 44, 49, 32], the deep Ritz method (DRM) [21], the weak adversarial networks (WAN) [68], and the deep Nitsche method (DNM) [38]. The deep BSDE reformulates the time-dependent equations into stochastic optimization problems. The DGM and the PINN train neural networks by minimizing the mean squared error loss of the PDE residual, while the DRM trains networks by minimizing the energy functional of the variational problem that is equivalent to the targeted PDE. The WAN adopts the weak form of original PDEs and trains the primary and adversarial network alternatively with the min-max weak form, and, the DNM enhances the DRM with natural treatment of essential boundary conditions. Recently, an additional neural network is trained to impose Dirichlet boundary conditions [57].

In this paper, we employ a likelihood of the DGM and/or of the PINN approaches to solve both coupled nonlinear hyperbolic systems by adopting DNN functionals to approximate unknown variables, reformulating each PDE model and its initial & boundary conditions as a series of least-squares (LS) problems in the mean squared error form, whose summation defines a total loss function, then minimizing this loss function with a standard optimization algorithms such as the stochastic gradient descent (SGD) method [53, 7, 43, 37, 54], the trust region method [13, 52, 65] or the derivative-free method [50, 2, 66] based upon a $(d+1)$-dimensional space-time

sample point (training) set. We will numerically illustrate that the mesh-free DNN-approximated solutions can produce an acceptable accuracy, which however does not actually show a clear convergence pattern with respect to the number of sample points and the DNN's structure (the number of neurons/hidden layers). This is unsurprised since the approximation properties of DNN approach still remain an active and open question. Mathematically, only a universal approximation theory exists for the shallow neural networks [47], which leads to more recent convergence analysis work for DRM [70, 10, 58, 17, 41, 29], but still far from a satisfied convergence theory for the DNN. Even so, the advantage of using the mesh-free DNN approach circumvents the meshing procedure that especially remains as a challenging task for solving high-dimensional problems with complex geometrical domains, greatly reduces difficulties of the methodology development and computational costs that traditional numerical methods (e.g., FEMs) usually have to confront, and makes a real-time simulation/prediction possible, in practice.

In contrast to the uniformly developed mesh-free DNN approach, we will also develop different FEMs for each presented coupled nonlinear hyperbolic system. It is well known that developing a stable and convergent finite element approximation to KdV equations is always subtle due to its highly hyperbolic property. As for the system of KG equations, since it belongs to the Hamiltonian system, an energy-conserved finite element discretization is crucial to preserve the total energy (the sum of kinetic and potential energy) all the time. However, a generic FEM does not usually fulfill this aim for KG equations, an energy-preserving FEM needs to be delicately designed in order to conserve the total energy of KG equations during the time-marching process. In this paper, we propose to develop a stable and convergent FEM for the presented high-dimensional nonlinear system of coupled KdV equations, and a particularly energy-preserving FEM for the presented coupled nonlinear KG equations in a stable and accurate fashion. Both FEMs can numerically deliver optimal convergence rates for corresponding nonlinear hyperbolic system in their respective energy norms.

Then, we come to a conclusion of the comparative study between two approaches for two hyperbolic/wave system, that is, both the DNN approach and the FEM have their respective pros and cons in terms of the methodology complexity, the development difficulty, the approximation accuracy, and/or the energy conservation, which are also validated in numerical experiments by comparing convergent accuracies of the developed FEMs and approximation performances of the proposed mesh-free DNN method for both hyperbolic/wave equations based upon different types of discretization parameters varying in doubling, and specifically, comparing discrete energies obtained from both approaches for KG equations.

The rest of this paper is organized as follows: In Section 2, we first introduce two types of coupled system of nonlinear hyperbolic/wave equations: KdV equations and KG equations. Then, we develop their specific finite element approximations, respectively, in Section 3. Mesh-free DNN approaches are described, generally and particularly, for each nonlinear hyperbolic problem in Section 4. Numerical experiments and comparative studies are carried out for two presented hyperbolic/wave problems by means of the developed FEMs and DNN approaches, respectively, in Section 5.

In what follows, we use the standard notation for Sobolev spaces $W^{l,p}(\Psi)$ $(0 \leq l < \infty, 1 \leq p \leq \infty)$ and time-dependent Sobolev spaces $H^m(0,T;W^{l,p}(\Psi))$ $(0 \leq m < \infty)$ (or $H^m(W^{l,p}(\Psi))$ in abbreviation), and their associated norms defined in the domain Ψ and within the time interval $[0,T]$. For $p=2$, $W^{l,2}(\Psi) = H^l(\Psi)$, the

standard L^2 inner product $(\cdot,\cdot)$ is adopted, and $\|\cdot\|_{s,\Psi} = \|\cdot\|_{H^s(\Psi)}$ for $0 \le s \le l$, where, when $s = 0$, $H^0(\Psi)$ coincides with $L^2(\Psi)$. In particular, if $m = l = 0$, $p = 2$, the time-dependent Sobolev space is denoted as $L^2(0,T;L^2(\Psi))$, which is abbreviated as $L^2(L^2(\Psi))$ and is equipped with the norm $\|\cdot\|_{L^2(L^2(\Psi))}$.

2. Model Descriptions

In what follows, we introduce two coupled nonlinear hyperbolic/wave problems to be studied in this paper, KdV equations and KG equations.

Korteweg-De Vries (KdV) equations: The $(2+1)$-dimensional coupled system of nonlinear KdV equations is defined as follows:

$$
\begin{cases}
u_t + u_{xxx} - 3u_x v - 3u v_x = f, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
u_x - v_y = 0, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
u(\boldsymbol{x},t) = g_1,\ v(\boldsymbol{x},t) = g_3, & (\boldsymbol{x},t) \in \partial\Omega \times (0,T], \\
u_{xx}(\boldsymbol{x},t) = g_2, & (\boldsymbol{x},t) \in \partial\Omega \times (0,T], \\
u(\boldsymbol{x},0) = u^0(\boldsymbol{x}), & \boldsymbol{x} \in \Omega,
\end{cases} \tag{1}
$$

where $\boldsymbol{x} = (x,y) \in \overline{\Omega} \subset \mathbb{R}^2$. This system of KdV equations was first derived by Boiti et al [5] and can also be considered as a model of an incompressible fluid [24] where u is a component of the velocity.

Klein-Gordon (KG) equations: The coupled nonlinear KG (second-order hyperbolic/wave) equations is described by the following system of PDEs:

$$
\begin{cases}
u_{tt} - \kappa^2 \Delta u + a_1 u + b_1 u^3 + c_1 u v^2 = 0, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
v_{tt} - \kappa^2 \Delta v + a_2 v + b_2 v^3 + c_2 u^2 v = 0, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
u(\boldsymbol{x},t) = 0,\ v(\boldsymbol{x},t) = 0, & (\boldsymbol{x},t) \in \partial\Omega \times (0,T], \\
u(\boldsymbol{x},0) = \phi_1(\boldsymbol{x}),\ v(\boldsymbol{x},0) = \phi_2(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\
u_t(\boldsymbol{x},0) = \psi_1(\boldsymbol{x}),\ v_t(\boldsymbol{x},0) = \psi_2(\boldsymbol{x}), & \boldsymbol{x} \in \Omega,
\end{cases} \tag{2}
$$

where $\Omega \subset \mathbb{R}^d$ $(d = 2,3)$, κ, a_i, b_i and c_i $(i = 1,2)$ are all positive constants, and b_i and c_i, $(i = 1,2)$ denote interaction constants, $u(\boldsymbol{x},t)$ and $v(\boldsymbol{x},t)$ represent interacting relativistic fields of masses. This system, first proposed by Segal in [56], describes the motion of charged mesons in a quantum/electromagnetic field.

We can reformulate (2) as the following unified form, equivalently [11]:

$$
\begin{cases}
\alpha u_{tt} - \nabla\cdot(\beta\nabla u) + \frac{\partial G}{\partial u}(u,v) = 0, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
\gamma v_{tt} - \nabla\cdot(\delta\nabla v) + \frac{\partial G}{\partial v}(u,v) = 0, & (\boldsymbol{x},t) \in \Omega \times (0,T], \\
u(\boldsymbol{x},t) = 0,\ v(\boldsymbol{x},t) = 0, & (\boldsymbol{x},t) \in \partial\Omega \times (0,T], \\
u(\boldsymbol{x},0) = \phi_1(\boldsymbol{x}),\ v(\boldsymbol{x},0) = \phi_2(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\
u_t(\boldsymbol{x},0) = \psi_1(\boldsymbol{x}),\ v_t(\boldsymbol{x},0) = \psi_2(\boldsymbol{x}), & \boldsymbol{x} \in \Omega,
\end{cases} \tag{3}
$$

where,

$$
G(u,v) = \frac{b_1}{4c_1}u^4 + \frac{b_2}{4c_2}v^4 + \frac{a_1}{2c_1}u^2 + \frac{a_2}{2c_2}v^2 + \frac{1}{2}u^2v^2, \tag{4}
$$

and $\alpha = 1/c_1$,$\beta = \kappa^2/c_1$, $\gamma = 1/c_2$ and $\delta = \kappa^2/c_2$.

It can be easily verified that the KG equations (3)-(4) preserve the following total energy functional with respect to time:

$$
E(t) = \frac{1}{2}\int_\Omega \left[\alpha(u_t)^2 + \beta(\nabla u)^2 + \gamma(v_t)^2 + \delta(\nabla v)^2 + 2G(u,v)\right] d\boldsymbol{x}, \tag{5}
$$

leading to [11, 28]

$$
\frac{dE(t)}{dt} = 0, \tag{6}
$$

i.e., the total energy $E(t)$ is conserved as a constant, $E(0)$, which is the initial energy given by

$$E(0) = \frac{1}{2}\int_{\Omega} \left[\alpha(\psi_1)^2 + \beta(\nabla\phi_1)^2 + \gamma(\psi_2)^2 + \delta(\nabla\phi_2)^2 + 2G(\phi_1,\phi_2)\right] d\boldsymbol{x}, \tag{7}$$

where the homogeneous Dirichlet boundary condition shown in (3) is necessary to derive (6).

3. Finite element methods for two types of hyperbolic/wave problems

3.1. FEM for the coupled KdV equations. To develop an appropriate finite element approximation to (1) that is convergent in an optimal fashion, we first try a straight variational (weak) form for (1) that is directly derived from the governing equations, i.e., find $(u, v) \in H^2_{g_1}(\Omega) \times H^1_{g_3}(\Omega)$ such that

$$\begin{cases} (u_t, \tilde{u}) - 3(u_x v, \tilde{u}) - 3(u v_x, \tilde{u}) - (u_{xx}, \tilde{u}_x) = (f, \tilde{u}), & \forall \tilde{u} \in H^1_0(\Omega), \\ (u_x, \tilde{v}) - (v_y, \tilde{v}) = 0, & \forall \tilde{v} \in H^1(\Omega), \end{cases} \tag{8}$$

where $H^j_{g_i}(\Omega) := \{u \in H^j(\Omega) : u|_{\partial\Omega} = g_i\}$ for $i = 1, 2, 3$ and $j = 1, 2$, $H^1_0(\Omega) := \{u \in H^1(\Omega) : u|_{\partial\Omega} = 0\}$. However in our preliminary numerical trials, due to the highly hyperbolic characteristics of the presented KdV equations, the finite element discretization based upon the above direct weak form (8) cannot produce a stable and convergent numerical result. In fact, the linear algebraic system derived from the FEM lacks a dominant diagonal block for the finite element solution of the variable v since both v_t and v_y are missing in (1), consequently, the mass matrix block and the diffusion part of stiffness matrix block that are associated with v are missing too in (8). In view of these numerical difficulties, we develop a new FEM for the coupled KdV equations by starting with the following new weak form to overcome flaws existing in (8), specifically.

First of all, we introduce an extra scalar-valued variable $w = u_{xx}$ to (1), and reformulate $(1)_1$ using three variables u, w and v. Then, we differentiate $(1)_2$ with respect to y, i.e., $u_{xy} = v_{yy}$ that is subjecting to an extra boundary condition: $u_x = v_y$ on $\partial\Omega \times (0, T]$, which can actually lead to the original equation $(1)_2$. Thus, we can derive the following weak form for such a reformulated equivalent strong form of (1): find $(u, w, v) \in H^1_{g_1}(\Omega) \times H^1_{g_2}(\Omega) \times H^1_{g_3}(\Omega)$ such that

$$\begin{cases} (u_t, \tilde{u}) + 3(vu, \tilde{u}_x) - (w, \tilde{u}_x) = (f, \tilde{u}), & \forall \tilde{u} \in H^1_0(\Omega), \\ (w, \tilde{w}) + (u_x, \tilde{w}_x) = 0, & \forall \tilde{w} \in H^1_0(\Omega), \\ (u_x, \tilde{v}_y) - (v_y, \tilde{v}_y) = 0, & \forall \tilde{v} \in H^1_0(\Omega). \end{cases} \tag{9}$$

Let $\mathcal{T}_h$ be a shape-regular simplicial decompositions of Ω with the mesh size h, and introduce the following finite element spaces:

$$\begin{aligned} W_h &:= \{u \in H^1(\Omega) : u|_K \in P^k(K), \forall K \in \mathcal{T}_h\}, \\ W_{h,g_i} &:= \{u \in W_h : u|_{\partial\Omega} = g_i\}, \quad i = 1, 2, 3, \\ W^0_h &:= \{u \in W_h : u|_{\partial\Omega} = 0\}, \end{aligned}$$

where P^k represents the k-th degree piecewise polynomial defined in each element $K \in \mathcal{T}_h$. Denote the temporal step size by Δt, namely, for $N \in \mathbb{Z}^+$, let $\Delta t = T/N$, $t_n = n\Delta t$, $u^n = u(t_n)$. Then, we can define the following fully discrete finite element approximation based upon the weak form (9) and the backward Euler scheme. Given $(u^n_h, w^n_h, v^n_h) \in W_{h,g_1} \times W_{h,g_2} \times W_{h,g_3}$, find $(u^{n+1}_h, w^{n+1}_h, v^{n+1}_h) \in$

$W_{h,g_1} \times W_{h,g_2} \times W_{h,g_3}$ for $n = 0, 1, 2, \cdots, N$, such that

$$
(10) \qquad \begin{cases} (\frac{u_h^{n+1}-u_h^n}{\Delta t}, \tilde{u}) + 3(v_h^{n+1}u_h^{n+1}, \tilde{u}_x) - (w_h^{n+1}, \tilde{u}_x) = 0, & \forall \tilde{u} \in W_h^0, \\ (w_h^{n+1}, \tilde{w}) + (u_{h,x}^{n+1}, \tilde{w}_x) = 0, & \forall \tilde{w} \in W_h^0, \\ (u_{h,x}^{n+1}, \tilde{v}_y) - (v_{h,y}^{n+1}, \tilde{v}_y) = 0, & \forall \tilde{v} \in W_h^0. \end{cases}
$$

Noting that $(10)_1$ is a nonlinear equation due to v_h^{n+1} in the coefficient, we employ Picard's scheme to linearize the nonlinear finite element system (10), as described in Algorithm 3.1.

Algorithm 3.1. *The linearized finite element scheme via Picard's iteration.*

(1) *Initialization of the time marching: set the time step $n = 0$ and u_h^0 be the interpolation of u^0 on $\mathcal{T}_h$.*

(2) *Initialization of the nonlinear iteration: let $(u_h^{n+1,0}, w_h^{n+1,0}, v_h^{n+1,0}) = (u_h^n, w_h^n, v_h^n)$ as the initial guess at the $(n+1)$-th time step $(n \geq 0)$.*

(3) *The linearized finite element method at the $(m+1)$-th iteration step: for $m \geq 0$, find $(u_h^{n+1,m+1}, w_h^{n+1,m+1}, v_h^{n+1,m+1}) \in W_{h,g_1} \times W_{h,g_2} \times W_{h,g_3}$ such that*

$$
(11) \qquad \begin{cases} (\frac{u_h^{n+1,m+1}-u_h^n}{\Delta t}, \tilde{u}) + 3(v_h^{n+1,m}u_h^{n+1,m+1}, \tilde{u}_x) - (w_h^{n+1,m+1}, \tilde{u}_x) = 0, & \forall \tilde{u} \in W_h^0, \\ (w_h^{n+1,m+1}, \tilde{w}) + (u_{h,x}^{n+1,m+1}, \tilde{w}_x) = 0, & \forall \tilde{w} \in W_h^0, \\ (u_{h,x}^{n+1,m+1}, \tilde{v}_y) - (v_{h,y}^{n+1,m+1}, \tilde{v}_y) = 0, & \forall \tilde{v} \in W_h^0. \end{cases}
$$

(4) *Check the stopping criteria for the nonlinear iteration: for a given tolerance ε, stop the iteration if*

$$
(12) \qquad \|u_h^{n+1,m+1} - u_h^{n+1,m}\|_0 + \|w_h^{n+1,m+1} - w_h^{n+1,m}\|_0 + \|v_h^{n+1,m+1} - v_h^{n+1,m}\|_0 \leq \varepsilon,
$$

then set $(u_h^{n+1}, w_h^{n+1}, v_h^{n+1}) = (u_h^{n+1,m+1}, w_h^{n+1,m+1}, v_h^{n+1,m+1})$. Otherwise, set $m+1$ to m, go back to Step 3 and continue the nonlinear iteration.

(5) *Time marching: if $n+1 < N$, then set $n+1$ to n, go back to Step 2 and continue the time marching. Otherwise, stop the entire computation.*

Remark 3.1. *In the above development of an optimally convergent FEM for the presented KdV equations, we particularly introduce an extra variable w that equals u_{xx} in order to reduce the weak space to which the weak solution u belongs from $H^2(\Omega)$ down to $H^1(\Omega)$, which is crucial to define a valid weak form with respect to the weak solution u, further, a valid finite element discretization with respect to the finite element solution u_h, in an efficient and accurate manner. On the other hand, to obtain a stable finite element approximation to the other variable v, we reformulate the second equation of KdV system, $u_x = v_y$, to become $u_{xy} = v_{yy}$ that subjects to an extra boundary condition: $u_x = v_y$ on the boundary. They lead to $(9)_2$ and $(9)_3$ in the weak form. and $(10)_2$ and $(10)_3$ in the finite element discretization, where we particularly attain the inner product term $(v_y, \tilde{v}_y)$ for the variable v that can lead to a nonsingular, dominant diagonal block of sub-matrix that is associated with the finite element solution v_h, which results in a stable and optimal convergence for v_h even though it lacks a mass matrix that u_h however holds.*

We will rigorously conduct stability and convergence analyses of the FEM for the presented KdV equations defined in (10) or (11) in our next paper rather than here, so that we can focus on our true mission in this paper: a comparative study of numerical performance between FEM and DNN for nonlinear hyperbolic equations, where a solid convergence analysis for the DNN's approximation is still missing

even these days, so we omit the finite element analysis in this paper as well. But numerical experiments in Section 5.1 will illustrate the optimal convergence rate of the developed FEM for the presented KdV equations.

3.2. FEM for the coupled KG equations. It is well known that a straightforward finite element approximation to the second-order nonlinear wave equation such as the presented system of KG equations (2), cannot produce a conservative energy in a discrete manner if two nonlinear source terms, $\frac{\partial G(u,v)}{\partial u}$ and $\frac{\partial G(u,v)}{\partial v}$, are directly discretized in their original polynomial forms, i.e., without employing a first-order difference scheme, even though we know that (2), or equivalently (3), belongs to the Hamiltonian system and preserves the energy all the time on the continuous level [45]. To preserve the discrete energy for nonlinear KG equations (2) at each time step, it is crucial to treat the nonlinear source terms of (2) as partial derivatives of another differentiable function $G(u, v)$, then discretize $\frac{\partial G}{\partial u}$ and $\frac{\partial G}{\partial v}$ with proper difference schemes of order one. In addition, the temporal discretizations for u_{tt}, v_{tt} as well as for diffusion terms also need to be appropriately defined. To that end, two types of time difference methods, the three-level (leap-frog) method and the two-level method, can be chosen to play the role of temporal discretization, where u_{tt} and v_{tt} are directly discretized by the second-order central difference scheme in the three-level method, while in the two-level method they are reformulated as p_t and b_t by introducing new variables $p = u_t$, $b = v_t$, and, are both discretized by the first-order difference scheme.

A three-level energy-preserving finite element approximation to (3) is feasible to be developed, which is partially verified in [45, 12], where a three-level energy-preserving finite difference method is developed to preserve the induced two-level discrete energy that involves two steps' numerical solutions, and thus more post-processing work are needed to find spatial and temporal derivatives of numerical solutions in order to compute such a two-level discrete energy, let alone the extra discrete initial values, u_h^1 and v_h^1, need to be properly chosen in order not to lose approximation accuracy. Therefore in this paper, we rather choose to develop a two-level energy-preserving finite element approximation to (3) as initiated above, by reformulating (3) in terms of four variables u, p, v and b, which leads to the following weak form of (3) with the homogeneous Dirichlet boundary condition shown in $(3)_3$: find $(u, p, v, b) \in H_0^1(\Omega) \times L^2(\Omega) \times H_0^1(\Omega) \times L^2(\Omega)$ such that

$$
\left\{
\begin{array}{ll}
(\alpha p_t, \tilde{u}) + (\beta \nabla u, \nabla \tilde{u}) + (\frac{\partial G}{\partial u}, \tilde{u}) = 0, & \forall \tilde{u} \in H_0^1(\Omega), \\
(\alpha p, \tilde{p}) = (\alpha u_t, \tilde{p}), & \forall \tilde{p} \in L^2(\Omega), \\
(\gamma b_t, \tilde{v}) + (\delta \nabla v, \nabla \tilde{v}) + (\frac{\partial G}{\partial v}, \tilde{v}) = 0, & \forall \tilde{v} \in H_0^1(\Omega), \\
(\gamma b, \tilde{b}) = (\gamma v_t, \tilde{b}), & \forall \tilde{b} \in L^2(\Omega).
\end{array}
\right. \tag{13}
$$

Then, based on the same spatial triangulation $\mathcal{T}_h$ in Ω and the same temporal partition in $[0, T]$, as shown in Section 3.1, we define the following finite element spaces:

$$
U_h \quad := \quad \{v \in L^2(\Omega) : v|_K \in P^k(K), \forall K \in \mathcal{T}_h\}.
$$

Let $\varphi^n = \phi(t_n)$, $\varphi^{n+\frac{1}{2}} = \frac{\varphi^{n+1}+\varphi^n}{2}$ and $d_t\varphi^{n+\frac{1}{2}} = \frac{\varphi^{n+1}-\varphi^n}{\Delta t}$. We thus define the following fully discrete finite element approximation for (3)-(4) based upon the weak form (13) and the Crank-Nicolson scheme. Given $(u_h^n, p_h^n, v_h^n, b_h^n) \in W_h^0 \times U_h \times W_h^0 \times U_h$, find $(u_h^{n+1}, p_h^{n+1}, v_h^{n+1}, b_h^{n+1}) \in W_h^0 \times U_h \times W_h^0 \times U_h$ for $n = 0, 1, 2, \cdots, N$, such

that

$$
(14)\quad \begin{cases} (\alpha d_t p_h^{n+\frac{1}{2}}, \tilde{u}) + (\beta \nabla u_h^{n+\frac{1}{2}}, \nabla \tilde{u}) + (d_u G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}}), \tilde{u}) = 0, & \forall \tilde{u} \in W_h^0, \\ (\alpha p_h^{n+\frac{1}{2}}, \tilde{p}) = (\alpha d_t u_h^{n+\frac{1}{2}}, \tilde{p}), & \forall p \in U_h, \\ (\gamma d_t b_h^{n+\frac{1}{2}}, \tilde{v}) + (\delta \nabla v_h^{n+\frac{1}{2}}, \nabla \tilde{v}) + (d_v G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}}), \tilde{u}) = 0, & \forall \tilde{v} \in W_h^0, \\ (\gamma b_h^{n+\frac{1}{2}}, \tilde{b}) = (\gamma d_t v_h^{n+\frac{1}{2}}, \tilde{b}), & \forall b \in U_h, \end{cases}
$$

where [11]

(15)
$$
d_u G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}}) = \frac{\left[G(u_h^{n+1}, v_h^{n+1}) + G(u_h^{n+1}, v_h^n)\right] - \left[G(u_h^n, v_h^{n+1}) + G(u_h^n, v_h^n)\right]}{2(u_h^{n+1} - u_h^n)},
$$

(16)
$$
d_v G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}}) = \frac{\left[G(u_h^{n+1}, v_h^{n+1}) + G(u_h^n, v_h^{n+1})\right] - \left[G(u_h^{n+1}, v_h^n) + G(u_h^n, v_h^n)\right]}{2(v_h^{n+1} - v_h^n)}.
$$

Carrying out an analogous analysis as done in [28], we can easily verify that the total discrete energy E_h at the n-th time step defined by

$$
(17)\qquad E_h^n = \frac{1}{2}\int_\Omega \left[\alpha (p_h^n)^2 + \beta (\nabla u_h^n)^2 + \gamma (b_h^n)^2 + \delta (\nabla v_h^n)^2 + 2G(u_h^n, v_h^n)\right] d\boldsymbol{x},
$$

is preserved as the initial energy E_h^0, i.e., $E_h^n = E_h^0$ for $n = 1, 2, \cdots, N$, where

(18)
$$
E_h^0 = \frac{1}{2}\int_\Omega \left[\alpha (\psi_{1,h})^2 + \beta (\nabla \phi_{1,h})^2 + \gamma (\psi_{2,h})^2 + \delta (\nabla \phi_{2,h})^2 + 2G(\phi_{1,h}, \phi_{2,h})\right] d\boldsymbol{x},
$$

and, $\psi_{i,h}, \phi_{i,h}$ are the interpolation values of ψ_i, ϕ_i $(i = 1, 2)$ in the corresponding finite element spaces U_h, W_h^0, respectively. Therefore we confirm that (14) defines an energy-preserving FEM.

We can see that (14) is a nonlinear system including the nonlinear term $d_u G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}})$ and $d_v G(u_h^{n+\frac{1}{2}}, v_h^{n+\frac{1}{2}})$. The following Algorithm 3.2 describes how we adopt the Picard's iteration method to linearize these two nonlinear terms and implement a nonlinear iteration process for the developed energy-preserving FEM (14).

Algorithm 3.2. *The linearized energy-preserving finite element scheme.*

(1) *Initialization of the time marching: set the time step* $n = 0$ *and* $(u_h^0, p_h^0, v_h^0, b_h^0)$ *be the interpolation of* $(\phi_1, \psi_1, \phi_2, \psi_2)$ *on* $\mathcal{T}_h$.
(2) *Initialization of the nonlinear iteration: let* $(u_h^{n+1,0}, p_h^{n+1,0}, v_h^{n+1,0}, b_h^{n+1,0}) = \frac{1}{2}(u_h^n, p_h^n, v_h^n, b_h^n)$ *as the initial guess at the* $(n+1)$*-th time step* $(n \geq 0)$.
(3) *The linearized energy-preserving finite element method at the* $(m+1)$*-th iteration step: for* $m \geq 0$, *find*

$(u_h^{n+1,m+1}, p_h^{n+1,m+1}, v_h^{n+1,m+1}, b_h^{n+1,m+1}) \in W_h^0 \times U_h \times W_h^0 \times U_h$ *such that*

$$\begin{cases} \left(\alpha\frac{p_h^{n+1,m+1}-p_h^n}{\Delta t}, \tilde{u}\right) + \left(\beta\frac{\nabla u_h^{n+1,m+1}+\nabla u_h^n}{2}, \nabla\tilde{u}\right) \\ \quad + \left(\frac{\left[G(u_h^{n+1,m}, v_h^{n+1,m}) + G(u_h^{n+1,m}, v_h^n)\right] - \left[G(u_h^n, v_h^{n+1,m}) + G(u_h^n, v_h^n)\right]}{2(u_h^{n+1,m} - u_h^n)}, \tilde{u}\right) = 0, \\ \left(\alpha\frac{p_h^{n+1,m+1}+p_h^n}{2}, \tilde{p}\right) = \left(\alpha\frac{u_h^{n+1,m+1}-u_h^n}{\Delta t}, \tilde{p}\right), \qquad \forall(\tilde{u}, \tilde{p}) \in W_h^0 \times U_h. \\ \left(\gamma\frac{b_h^{n+1,m+1}-p_h^n}{\Delta t}, \tilde{v}\right) + \left(\delta\frac{\nabla v_h^{n+1,m+1}+\nabla v_h^n}{2}, \nabla\tilde{v}\right) \\ \quad + \left(\frac{\left[G(u_h^{n+1,m}, v_h^{n+1,m}) + G(u_h^n, v_h^{n+1,m})\right] - \left[G(u_h^{n+1,m}, v_h^n) + G(u_h^n, v_h^n)\right]}{2(v_h^{n+1,m} - v_h^n)}, \tilde{v}\right) = 0, \\ \left(\gamma\frac{b_h^{n+1,m+1}+b_h^n}{2}, \tilde{b}\right) = \left(\gamma\frac{v_h^{n+1,m+1}-v_h^n}{\Delta t}, \tilde{b}\right), \qquad \forall(\tilde{v}, \tilde{b}) \in W_h^0 \times U_h. \end{cases}$$

(4) *Check the stopping criteria for the nonlinear iteration: for a given tolerance* ε*, stop the iteration if*

$$\begin{aligned} &\|u_h^{n+1,m+1} - u_h^{n+1,m}\|_0 + \|p_h^{n+1,m+1} - p_h^{n+1,m}\|_0 + \|v_h^{n+1,m+1} - v_h^{n+1,m}\|_0 \\ &\quad + \|b_h^{n+1,m+1} - b_h^{n+1,m}\|_0 \le \varepsilon, \end{aligned} \tag{19}$$

then set $(u_h^{n+1}, p_h^{n+1}, v_h^{n+1}, b_h^{n+1}) = (u_h^{n+1,m+1}, p_h^{n+1,m+1}, v_h^{n+1,m+1}, b_h^{n+1,m+1})$. *Otherwise, set* $m+1$ *to* m*, go back to Step 3 and continue the nonlinear iteration.*

(5) *Time marching: if* $n+1 < N$*, then set* $n+1$ *to* n*, go back to Step 2 and continue the time marching. Otherwise, stop the entire computation.*

Remark 3.2. *To let a nonlinear wave equation preserve the energy with time, continuously, we notice that its nonlinear right hand side (source term)* $f(\varphi)$ *must satisfy* $f(\varphi) = -\frac{\partial G(\varphi)}{\partial \varphi}$*, where* $G(\varphi)$ *is a third-order differentiable potential function with respect to the primary variable* φ*. Numerically, to obtain an energy-preserving discrete scheme for such a nonlinear wave equation, we need to discretize* $\frac{\partial G(\varphi)}{\partial \varphi}$ *using the first-order difference scheme, such as (15) or (16), instead of discretizing* $f(\varphi)$*, directly. Similar theoretical analyses for proving the energy conservation and optimal convergence properties of the proposed FEM (14) can refer to the authors' earlier studies in* [28].

4. Deep neural network approaches for two types of hyperbolic/wave problems

4.1. DNN approach for the coupled KdV equations. To apply the DNN approach to the coupled KdV equations (1), it is natural to consider the least-squares (LS) formulation [14] to minimize all residuals of governing equations and of initial & boundary conditions by defining a LS functional that incorporates all equations introduced in (1) together. Following that idea, we can define the following total LS functional for the coupled nonlinear KdV equations (1) with

respect to any $(\tilde{u},\tilde{v}) \in H^1(0,T;H^3(\Omega)) \times L^2(0,T;H^1(\Omega))$:

$$\begin{aligned}
&\mathcal{R}^{KdV}(\tilde{u},\tilde{v};f,g_1,g_2,g_3,u^0)\\
:=&\int_0^T \Big(\omega_1\|\tilde{u}_t+\tilde{u}_{xxx}-3\tilde{u}_x\tilde{v}-3\tilde{u}\tilde{v}_x-f\|^2_{0,\Omega}\\
&+\omega_2\|\tilde{u}_x-\tilde{v}_y\|^2_{0,\Omega}+\omega_3\|\tilde{u}-g_1\|^2_{0,\partial\Omega}\\
&+\omega_4\|\tilde{u}_{xx}-g_2\|^2_{0,\partial\Omega}+\omega_5\|\tilde{v}-g_3\|^2_{0,\partial\Omega}\Big)dt+\omega_6\|\tilde{u}(\boldsymbol{x},0)-u^0\|^2_{0,\Omega},\\
:=&\sum_{i=1}^{6}\mathcal{R}_i^{KdV}(\tilde{u},\tilde{v};f,g_1,g_2,g_3,u^0),\quad \forall(\tilde{u},\tilde{v})\in H^1(H^3(\Omega))\times L^2(H^1(\Omega)),
\end{aligned}\tag{20}$$

where ω_i $(i=1,\cdots,6)$ are weight coefficients of each corresponding L^2 inner product term. Then, the LS solution associated with the LS functional (20) is to find $(u,v)\in H^1(0,T;H^3(\Omega))\times L^2(0,T;H^1(\Omega))$ such that

$$\begin{aligned}
&\mathcal{R}^{KdV}(u,v;f,g_1,g_2,g_3,u^0)\\
=&\arg\min_{(\tilde{u},\tilde{v})\in H^1(H^3(\Omega))\times L^2(H^1(\Omega))}\mathcal{R}^{KdV}(\tilde{u},\tilde{v};f,g_1,g_2,g_3,u^0).
\end{aligned}\tag{21}$$

Remark 4.1. *An optimal choice for all weight coefficients ω_i in (20) can make the LS optimization process convergent faster, i.e., can accelerate the minimization of each LS term in (20) by letting it approach to zero as quickly as possible. In practice, each weight coefficient associate with its L^2-inner product or LS term is usually chosen as the reciprocal of the sum of the absolute value of all parameters in this LS term, just like a normalization process. On the other hand, a unified dimension for all LS terms in the total LS functional is also another important consideration for choosing these weight coefficients in order to accelerate the LS minimization for a realistic PDE problem.*

In this section, we aim to develop a mesh-free approach using the deep neural network (DNN) technique for the coupled KdV equations based on the above minimization problem (21). We can prescribe the space-time domain $\overline{\Omega}\times[0,T]$ that belongs to $\mathbb{R}^{d+1}$, then a $(d+1)$-th dimensional input sample point (training) set, $\mathbf{S}:=\{\boldsymbol{X}(\boldsymbol{x},t)\subset\mathbb{R}^{d+1}\,|\,\boldsymbol{x}\in\overline{\Omega},\ t\in[0,T]\}$, can be constructed by uniformly or randomly distributing sample points in the $(d+1)$-dimensional domain $\overline{\Omega}\times[0,T]$. Thus the computational mesh, which is always required by traditional numerical methods (e.g., FEMs) and is always troublesome for high-dimensional problems with complex geometrical domains, is completely removed from the following DNN's numerical simulations. Instead, we need the sample point (training) set $\mathbf{S}$ to train a DNN structure, then solve a discrete version of the minimization problem (21) based upon the DNN approximation to the solution of KdV equations. More precisely, we employ a feedforward DNN model to approximate KdV equations based upon a $(d+1)$-dimensional space-time sample point (training) set $\mathbf{S}$, that is, we utilize a fully connected DNN structure to approximate all primary variables (the solution of KdV equations), then adopt the LS formulation to minimize residuals of governing equations and of initial & boundary conditions of the proposed KdV equations (where $d=2$) by utilizing the DNN-approximated solution functions. Thereafter, we sum up all well-defined LS formulations together with physics-dependent weights to establish a total loss functional, $\mathcal{L}^{KdV}(\boldsymbol{X};\Theta)$, where $\boldsymbol{X}\in\mathbb{R}^{d+1}$ is an input vector that can be sampled from the space-time training set $\mathbf{S}$, $\Theta=\{\boldsymbol{W}^l,\boldsymbol{b}^l,\ l=1,\cdots,L\}$ are parameter variables that are used to construct the

L-layer DNN with the weights $\boldsymbol{W}^l$ and biases $\boldsymbol{b}^l$ in the l-th layer for $l = 1, \cdots, L$. A classical fully connected, feedforward DNN structure can be schematically shown in Figure 1.

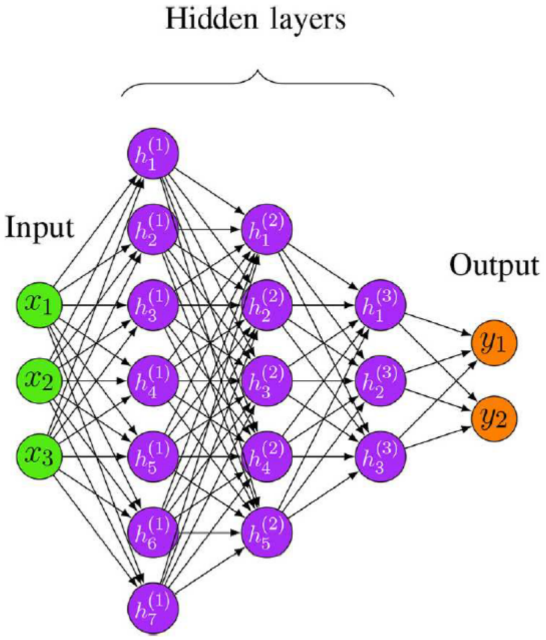


FIGURE 1. Illustration of a classical feedforward DNN structure with 3 hidden layers, where $h_n^{(l)}$ denotes the n-th neuron in the l-th hidden layer [22].

Concretely interpreting, let $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_{L+1}}$ represent the input and output spaces with n_1 and n_{L+1} units (neurons), respectively, and given any input vector $\boldsymbol{X} \in \mathbb{R}^{n_1}$, a feedforward neural network (also termed as a multi-layer perceptron), transforms the $\boldsymbol{X}$ to an output through layers of units (neurons) consisting scalar nonlinear activation functions within units [26], resulting in the following representation of a DNN function

$$\mathbf{NN}(\boldsymbol{X}; \Theta) = \mathbf{T}^L \circ \mathcal{N}^{L-1} \circ \cdots \circ \mathcal{N}^2 \circ \mathcal{N}^1(\boldsymbol{X}), \tag{22}$$

where, $\mathcal{N}^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ is the l-th layer neural network defined as:

$$\mathcal{N}^l(\boldsymbol{X}^l) = \sigma \circ \mathbf{T}^l(\boldsymbol{X}^l), \quad \text{for } l = 1, \cdots, L,$$

here $\sigma : \mathbb{R} \to \mathbb{R}$ is a scalar (nonlinear) activation function of which a large variety have been considered in the machine learning literature [26], popular choices for the activation function σ include the sigmoid function, the tanh function and the ReLU function. And, $\mathbf{T}^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ $(l = 1, \cdots, L)$ is a linear transformation defined as:

$$\mathbf{T}^l(\boldsymbol{X}^l) = \boldsymbol{W}^l \boldsymbol{X}^l + \boldsymbol{b}^l, \quad \text{for } \boldsymbol{W}^l \in \mathbb{R}^{n_{l+1} \times n_l},\ \boldsymbol{X}^l \in \mathbb{R}^{n_l},\ \boldsymbol{b}^l \in \mathbb{R}^{n_{l+1}}, \tag{23}$$

where n_l is the number of neurons in the l-th layer neural network for $l = 1, \cdots, L$, noting that $n_1 = d + 1 = 3$ and $n_{L+1} = 2$ for the proposed coupled KdV equations.

Thus in the terminology of machine learning, the deep neural network (22) consists of an input layer, an output layer and $(L-1)$ hidden layers for some

$1 < L \in \mathbb{Z}^+$. The l-th hidden layer (with n_l neurons) is given an input vector $\boldsymbol{X}^l \in \mathbb{R}^{n_l}$ and transforms it first by a linear transformation $\mathbf{T}^l$ shown in (23) and then by a nonlinear (component wise) activation σ. A straightforward addition shows that such defined DNN contains $\sum_{l=1}^{L} n_l$ neurons in total. It is also straightforward to check that the concatenated set of (tunable) weights and biases, $\Theta = \{\boldsymbol{W}^l, \boldsymbol{b}^l,\ l = 1, \cdots, L\} \in \mathbb{R}^{\mathcal{K}}$, with

$$\mathcal{K} = \sum_{l=1}^{L} n_{l+1}(n_l + 1), \tag{24}$$

denoting the total number of tuning parameters (weights and biases) of a fully connected, feedforward DNN. Additionally, we introduce the following nomenclature for a deep neural network **NN**,

$$\text{size}(\mathbf{NN}) := \mathcal{K}, \qquad \text{depth}(\mathbf{NN}) = L - 1,$$

with $L - 1$ being the number of hidden layers of the network.

With such a DNN structure, we develop a fully connected DNN approach to solve the presented coupled KdV equations (1), where we adopt $n_1 = d+1$, $n_{L+1} = 2$ for the DNN structure to define the following two DNN functions for an approximation to two unknowns (u, v) shown in (21), respectively:

$$\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X}; \Theta) \approx u, \quad \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X}; \Theta) \approx v.$$

Then, the mesh-free DNN approach is described as follows in its continuous version: find $(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X}; \Theta^*), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X}; \Theta^*))$ such that the following minimization problem:

$$\mathcal{L}^{KdV}(\boldsymbol{X}; \Theta^*) = \arg \min_{\Theta \in \Psi^{\mathcal{K}}} \mathcal{L}^{KdV}(\boldsymbol{X}; \Theta), \tag{25}$$

where, $\mathcal{L}^{KdV}(\boldsymbol{X}; \Theta)$ is the total loss functional defined below according to the LS functional (20):

$$\begin{aligned} \mathcal{L}^{KdV}(\boldsymbol{X}; \Theta) &:= \sum_{i=1}^{6} \mathcal{L}_i^{KdV}(\boldsymbol{X}; \Theta) \\ &:= \sum_{i=1}^{6} \mathcal{R}_i^{KdV}(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X}; \Theta), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X}; \Theta); f, g_1, g_2, g_3, u^0), \end{aligned} \tag{26}$$

$\Psi^{\mathcal{K}} := \{\Theta : \Theta|_{\mathcal{D}^i} \in \mathbb{R}^{\mathcal{K}}\}$, and $\mathcal{D}^i$ $(i = 1, \cdots, 6)$ stand for subregions of $\overline{\Omega} \times [0, T]$ associated with each LS functional $\mathcal{R}_i^{KdV}$ defined in (20), including the d-dimensional domain Ω at $t = 0$, the $(d+1)$-dimensional space-time $\Omega \times (0, T]$ and the d-dimensional space-time $\partial\Omega \times (0, T]$. In fact, $\sum_{i=1}^{6} \mathcal{D}^i := \overline{\Omega} \times [0, T]$.

In practice, the loss functional $\mathcal{L}^{KdV}(\boldsymbol{X}; \Theta)$ in its integral form is approximated by its discrete version through a prescribed quadrature rule, say, the Monte Carlo integration, based upon the input sample point (training) set, i.e., each LS functional, $\mathcal{L}_i^{KdV}(\boldsymbol{X}; \Theta)$ $(1 \le i \le 6)$, in (26) is approximated by its mean squared error (MSE) form, $\mathcal{L}_{\text{disc},i}^{KdV}(\boldsymbol{X}; \Theta)$, as:

$$\mathcal{L}_i^{KdV}(\boldsymbol{X}; \Theta) \approx \mathcal{L}_{\text{disc},i}^{KdV}(\boldsymbol{X}; \Theta) := \frac{1}{M_i} \sum_{k=1}^{M_i} |\mathcal{R}_{i,\mathbf{NN}}^{KdV}(\boldsymbol{X}|_{\mathcal{Q}^i}^k; \Theta)|^2,$$

where $\mathcal{R}^{KdV}_{i,\mathbf{NN}} = \mathcal{R}^{KdV}_i(\mathcal{U}_{\mathbf{NN}}, \mathcal{V}_{\mathbf{NN}}; f, g_1, g_2, g_3, u^0)$ stands for the DNN-approximated residual in each subregion $\mathcal{Q}^i \subseteq \overline{\Omega}\times[0,T]$ that is associated with each LS functional defined in (20), $\boldsymbol{X}|^k_{\mathcal{Q}^i}$ is the k-th space-time sample point falling into $\mathcal{Q}^i$, and M_i is the number of all sample points in $\mathcal{Q}^i$. If $M_i = 0$, then the corresponding discrete loss functional $\mathcal{L}^{KdV}_{\text{disc},i}(\boldsymbol{X};\Theta)$ is removed. $M = \sum_{i=1}^{6} M_i$ is the total size of the sample point (training) set. Thus, the total discrete loss functional, $\mathcal{L}^{KdV}_{\text{disc}}(\boldsymbol{X};\Theta)$, is well defined as

$$\mathcal{L}^{KdV}_{\text{disc}}(\boldsymbol{X};\Theta) = \sum_{i=1}^{6} \mathcal{L}^{KdV}_{\text{disc},i}(\boldsymbol{X};\Theta) \approx \mathcal{L}^{KdV}(\boldsymbol{X};\Theta).$$

On the other hand, during the process of computing residuals of governing equations, finding derivatives of DNN can be done by the backward/forward difference, or directly, by employing an automatic differentiation package specifically applying to the DNN function. Finally, standard optimization algorithms with the stochastic gradient descent (SGD) method [53, 7, 43, 37, 54] can be applied to solve the minimization problem:

$$\mathcal{L}^{KdV}_{\text{disc}}(\boldsymbol{X};\Theta^*) = \arg\min_{\Theta\in\Psi^{\mathcal{K}}} \mathcal{L}^{KdV}_{\text{disc}}(\boldsymbol{X};\Theta). \tag{27}$$

When the minimizer Θ^* is reached, we attain the desired DNN-approximated solution, $(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))$, as the DNN's solution of the presented KdV equations.

4.2. DNN approach for the coupled KG equations. First of all, we define a total least-squares (LS) functional for the coupled nonlinear KG equations (2) as follows

$$\begin{aligned}
&\mathcal{R}^{KG}(\tilde{u},\tilde{v};\phi_1,\phi_2,\psi_1,\psi_2)\\
&:= \int_0^T \Big(\omega_1\|\tilde{u}_{tt} - \kappa^2\Delta\tilde{u} + a_1\tilde{u} + b_1\tilde{u}^3 + c_1\tilde{u}\tilde{v}^2\|^2_{0,\Omega} +\\
&\quad \omega_2\|\tilde{v}_{tt} - \kappa^2\Delta\tilde{v} + a_2\tilde{v} + b_2\tilde{v}^3 + c_2\tilde{u}^2\tilde{v}\|^2_{0,\Omega} + \omega_3\|\tilde{u}\|^2_{0,\partial\Omega} + \omega_4\|\tilde{v}\|^2_{0,\partial\Omega}\Big)dt\\
&\quad + \omega_5\|\tilde{u}(\boldsymbol{x},0) - \phi_1\|^2_{0,\Omega} + \omega_6\|\tilde{v}(\boldsymbol{x},0) - \phi_2\|^2_{0,\Omega}\\
&\quad + \omega_7\|\tilde{u}_t(\boldsymbol{x},0) - \psi_1\|^2_{0,\Omega} + \omega_8\|\tilde{v}_t(\boldsymbol{x},0) - \psi_2\|^2_{0,\Omega}\\
&:= \sum_{i=1}^{8} \mathcal{R}^{KG}_i(\tilde{u},\tilde{v};\phi_1,\phi_2,\psi_1,\psi_2), \quad \forall(\tilde{u},\tilde{v}) \in H^2(H^2(\Omega)) \times H^2(H^2(\Omega)).
\end{aligned} \tag{28}$$

Then, the LS solution associated with the LS functional (28) is to find $(u,v) \in H^2(0,T;H^2(\Omega)) \times H^2(0,T;H^2(\Omega))$ such that

$$\begin{aligned}
&\mathcal{R}^{KG}(u,v;\phi_1,\phi_2,\psi_1,\psi_2)\\
&= \arg\min_{(\tilde{u},\tilde{v})\in H^2(H^2(\Omega))\times H^2(H^2(\Omega))} \mathcal{R}^{KG}(\tilde{u},\tilde{v};\phi_1,\phi_2,\psi_1,\psi_2).
\end{aligned} \tag{29}$$

We adopt $n_1 = d+1$, $n_{L+1} = 2$ for the same fully connected, feedforward DNN structure as described in Section 4.1 to define the following two DNN functions for an approximation to two unknowns (u,v) shown in (29), respectively:

$$\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta) \approx u, \quad \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta) \approx v.$$

The total loss functional, $\mathcal{L}^{KG}(\boldsymbol{X};\Theta)$, according to the LS functional (28), is then defined as

(30)
$$\mathcal{L}^{KG}(\boldsymbol{X};\Theta) := \sum_{i=1}^{8} \mathcal{L}_i^{KG}(\boldsymbol{X};\Theta) := \sum_{i=1}^{8} \mathcal{R}_i^{KG}(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta); \phi_1, \phi_2, \psi_1, \psi_2).$$

Therefore, the mesh-free DNN approach for solving the coupled KG equations in its continuous version is to find $(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))$ such that the following minimization problem:

$$\mathcal{L}^{KG}(\boldsymbol{X};\Theta^*) = \arg\min_{\Theta\in\Psi^{\mathcal{K}}} \mathcal{L}^{KG}(\boldsymbol{X};\Theta). \tag{31}$$

To derive its discrete version, we first use the following discrete loss functional,

$$\mathcal{L}_{\text{disc}}^{KG}(\boldsymbol{X};\Theta) = \sum_{i=1}^{8} \mathcal{L}_{\text{disc},i}^{KG}(\boldsymbol{X};\Theta) := \sum_{i=1}^{8} \frac{1}{M_i} \sum_{k=1}^{M_i} |\mathcal{R}_{i,\mathbf{NN}}^{KG}(\boldsymbol{X}|_{\mathcal{Q}^i}^k;\Theta)|^2, \tag{32}$$

to approximate the continuous loss functional, $\mathcal{L}^{KG}(\boldsymbol{X};\Theta)$, in its integral form by a prescribed quadrature rule, e.g., Monte Carlo integration. In (32), $\mathcal{R}_{i,\mathbf{NN}}^{KG} = \mathcal{R}_i^{KG}(\mathcal{U}_{\mathbf{NN}}, \mathcal{V}_{\mathbf{NN}}; \phi_1, \phi_2, \psi_1, \psi_2)$ stands for the DNN-approximated residual in each subregion $\mathcal{Q}^i \subseteq \overline{\Omega}\times[0,T]$ that is associated with each LS functional defined in (28). Then, the discrete version of the mesh-free DNN approach for solving the coupled KG equations is to solve the following minimization problem

$$\mathcal{L}_{\text{disc}}^{KG}(\boldsymbol{X};\Theta^*) = \arg\min_{\Theta\in\Psi^{\mathcal{K}}} \mathcal{L}_{\text{disc}}^{KG}(\boldsymbol{X};\Theta) \tag{33}$$

with a standard optimization algorithms such as the SGD method based upon a $(d+1)$-dimensional (space-time) sample point (training) set. When the minimizer Θ^* is reached, we attain the desired DNN-approximated solution, $(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*)$, $\mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))$, for the presented KG equations.

In order to compare with the energy-preserving FEM specifically designed for KG equations in Section 3.2, we need to compute the DNN's total discrete energies for KG equations, $E_{\mathbf{NN}}(t)$, at each time level t_n $(n = 0, 1, \cdots, N)$ based upon a uniformly distributed space-time sample point set with the total size $M = N_\Omega \times (N+1)$, where N_Ω denotes the number of sample points in space at each time level. According to (5), the DNN-driven total discrete energy, $E_{\mathbf{NN}}$, is defined as

$$\begin{aligned} E_{\mathbf{NN}}(t) \quad = \quad & \frac{1}{2}\int_\Omega \Bigg[\alpha\left(\frac{\partial \mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*)}{\partial t}\right)^2 + \beta(\nabla \mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))^2 \\ & +\gamma\left(\frac{\partial \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*)}{\partial t}\right)^2 + \delta(\nabla \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))^2 \\ & +2G(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))\Bigg] d\boldsymbol{x}, \end{aligned} \tag{34}$$

where $(\mathcal{U}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*), \mathcal{V}_{\mathbf{NN}}(\boldsymbol{X};\Theta^*))$ is the DNN's solution of KG equations with the minimizer Θ^*. Numerically, we will compare $E_{\mathbf{NN}}(t)$ with $E_{\mathbf{NN}}(0)$ for $t \in (0, T]$ in Section 5.2 to see whether or not the energy is conserved, if not, how the energy changes in time.

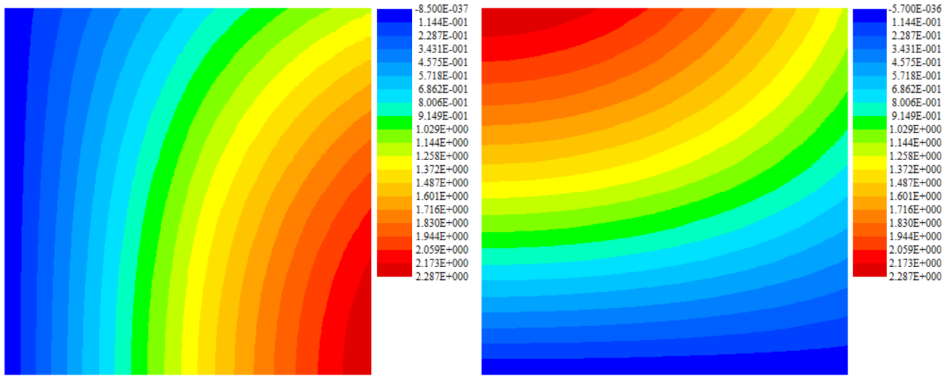
5. Numerical experiments

5.1. FEM and DDN approach for solving the coupled KdV equations. The following two functions

$$\begin{cases} u(x,y,t) = e^t \sin x \cos y, \\ v(x,y,t) = e^t \cos x \sin y, \end{cases} \tag{35}$$

which are defined in a two-dimensional domain $\Omega = [0,1] \times [0,1]$ and the time interval $[0,1]$, can be verified as the unique exact solution to the coupled nonlinear KdV equations (1) by choosing corresponding functions f, g_1, g_2, g_3 and u^0, accordingly. We first adopt the piecewise linear (P^1) polynomial, i.e., $k = 1$, to construct finite element spaces W_h, W_{h,g_i} $(i = 1, 2, 3)$ and W_h^0 that are appropriately employed by the developed FEM (10). Then we implement Algorithm 3.1 to carry out a series of finite element approximations by using the grid doubling, i.e., taking the mesh size $h = 1/10, 1/20, 1/40$ and $1/80$, simultaneously, and letting $\Delta t = h^2$ at different mesh levels. Convergence errors of linear finite element approximations to the solution (u, w, v) at the terminal time $T = 1$ are reported in Table 1 and Figure 3, from which we can see that optimal convergence rates of all three variables are uniformly obtained as: the second order in L^2 norm, and the first-order in H^1 norm, namely

$$\begin{aligned} &\|u^N - u_h^N\|_0 + \|v^N - v_h^N\|_0 + \|w^N - w_h^N\|_0 \\ &\quad + h(\|u^N - u_h^N\|_1 + \|v^N - v_h^N\|_1 + \|w^N - w_h^N\|_1) = O(h^2 + \Delta t). \end{aligned} \tag{36}$$



(a) FEM solution u_h (b) FEM solution v_h

FIGURE 2. Numerical results of the finite element solution for KdV equations on the finest grid at $t = T$.

Next, we apply the mesh-free DNN approach as described in Section 4.1 to solve the same coupled KdV equations (1) defined in the above (2+1)-dimensional space-time domain for the same exact solution (35). A uniformly distributed sample point (training) set in the space is adopted by partitioning the rectangular domain Ω in the same way as the above finite element grid does, however, we do not need any grid element information but only grid points as sample points of the training set. By using such a uniformly generated sample point (training) set that coincides with the finite element grid points, we can not only carry out a training process of the DNN, but also plot the contour figure of DNN's solutions by means of a finite element grid formed by these sample points, as shown in Figure

TABLE 1. Convergence results of FEM solutions for KdV equations at $t = T$.

h	1/10	1/20	1/40	1/80
$\|u - u_h\|_0$	6.961E-03	1.758E-03	4.402E-04	1.101E-04
Order		1.99	2.00	2.00
$\|u - u_h\|_1$	1.121E-01	5.493E-02	2.725E-02	1.357E-02
Order		1.03	1.01	1.01
$\|w - w_h\|_0$	8.045E-02	2.015E-02	5.014E-03	1.249E-03
Order		2.00	2.01	2.01
$\|w - w_h\|_1$	2.490E+00	1.256E+00	6.272E-01	3.128E-01
Order		0.99	1.00	1.00
$\|v - v_h\|_0$	3.317E-03	8.538E-04	2.154E-04	5.399E-05
Order		1.96	1.99	2.00
$\|v - v_h\|_1$	1.021E-01	5.104E-02	2.552E-02	1.276E-02
Order		1.00	1.00	1.00

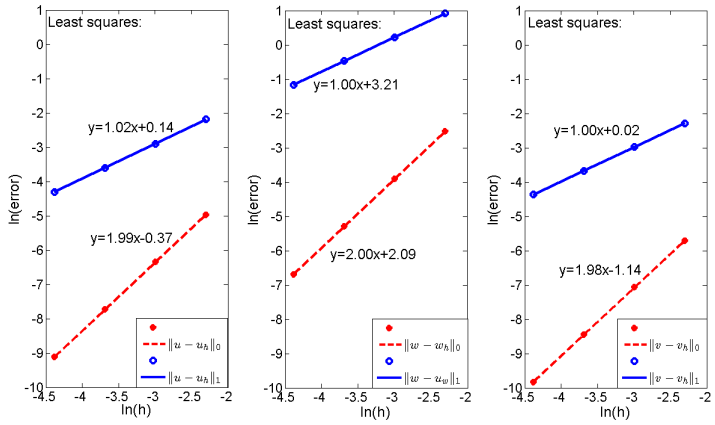
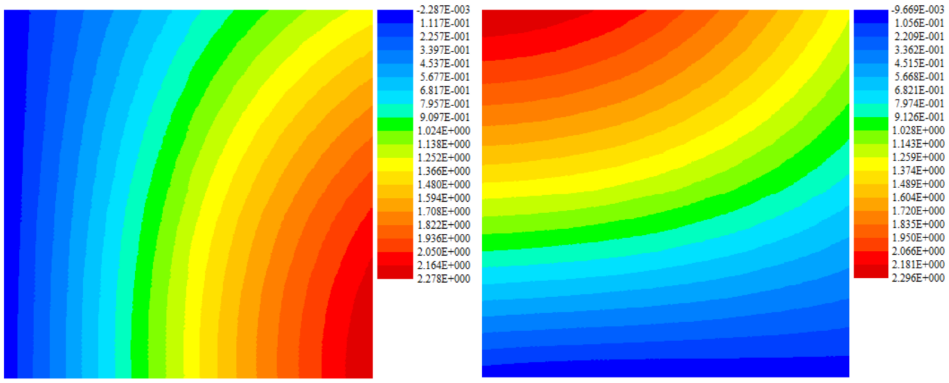


FIGURE 3. Convergence history of the finite element solution for KdV equations at $t = T$.

4. In the meanwhile, a comparison can be easily and accurately conducted between solutions of FEM and DNN at the same grid points and the same time level. On the other hand, the sampling point set along the time dimension is also uniformly distributed that is independent from the spatial distribution. Though, it does not mean a uniformly distributed sample points set is necessary for the DNN's approach, actually, a randomly distributed $(d + 1)$-dimensional sample points set also works well for the DNN approximation which is insensitive to the way of generating sample points, further explaining that the DNN's approach is mesh-free, essentially.

In addition, we adopt a variant of the stochastic gradient descent method, ADAM [36], with an initial learning rate of 0.001 and 2×10^4 epochs to solve the minimization problem (27) formed by the DNN approximation and the LS-formulated loss functional (20). On the finest sample points set in space that is the same finest grid points set of the FEM, Figure 4 shows contour results of the DNN's solution $(\mathcal{U}_{\mathbf{NN}}, \mathcal{V}_{\mathbf{NN}})$ at the highest time level $t_N = T = 1$, which is comparable with that of the finite element solution (u_h^N, v_h^N) shown in Figure 2, and the difference between them is almost invisible. In fact, if exactly computing the total error between



(a) DNN solution u_h (b) DNN solution v_h

FIGURE 4. Numerical results of the DNN's solution for KdV equations on the finest sample points set at $t = T$.

solutions of FEM and DNN on the finest sample points set at $t_N = T = 1$, we obtain $\|u_h^N - \mathcal{U}_{\mathbf{NN}}^N\|_{L^2(\Omega))} + \|v^N - \mathcal{V}_{\mathbf{NN}}^N\|_{L^2(\Omega))} \approx 2.79 \times 10^{-2}$, which holds the same magnitude with DNN's approximation errors over the sample set doubling shown in Table 2 or over the DNN's structure doubling shown in Table 3.

Circumstantiating in detail, along with the doubling size of the sample point (training) set, M, from $10 \times 10 \times 5$ up to $80 \times 80 \times 40$ in the $(2+1)$ dimensional space-time domain, while fixing the DNN structure as 4 hidden layers with 10 neurons in each layer, i.e., 4×10 neurons in total, we carry out the same mesh-free DNN's approximation to the coupled KdV equations (1) with the exact solution (35), and obtain the total approximation errors $|||e_{uv}||| = \|u - \mathcal{U}_{\mathbf{NN}}\|_{L^2(0,T;L^2(\Omega))} + \|v - \mathcal{V}_{\mathbf{NN}}\|_{L^2(0,T;L^2(\Omega))}$ shown in Table 2. On the other hand, Table 3 illustrates $|||e_{uv}|||$ along with different DNN structures by fixing the size of training set $M = 40 \times 40 \times 20 = 32000$ while doubling the number of neurons and the number of hidden layers of the neural network. Both tables show DNN's approximation errors with almost the same magnitude, 10^{-2}, which does not show a distinct convergence pattern in terms of either the sample set size or the number of neurons/layers of the DNN, as the FEM does in terms of mesh size h. But, they do show a distinct convergence trend, i.e., the approximation errors $|||e_{uv}|||$ decrease to some extent if increasing either the size of training set M, or the number of neurons/layers, which leads to the smallest approximation error shown at the lower right corner of Table 3, relatively.

TABLE 2. Approximation errors of DNN's solutions with 4 layers $\times$ 10 neurons for KdV equations.

M	$\|\|\|e_{uv}\|\|\|$
$10 \times 10 \times 5$	1.666×10^{-2}
$20 \times 20 \times 10$	1.872×10^{-2}
$40 \times 40 \times 20$	1.520×10^{-2}
$80 \times 80 \times 40$	1.537×10^{-2}

TABLE 3. Approximation errors of various DNN's solutions with $M = 40 \times 40 \times 20$ for KdV equations.

# of Neurons	1 Layer	2 Layers	4 Layers
4	1.648×10^{-1}	4.609×10^{-2}	4.802×10^{-2}
8	5.707×10^{-2}	3.601×10^{-2}	1.811×10^{-2}
16	2.457×10^{-2}	2.130×10^{-2}	1.395×10^{-2}
32	1.961×10^{-2}	1.507×10^{-2}	1.124×10^{-2}

Remark 5.1. *The above less accurate approximation results of DNN for the presented KdV equations further confirm the following fact, currently the approximation process of the DNN's approach still remain an active and open problem. Although a universal approximation theory exists for the shallow neural networks* [47]*, mathematically, which leads to some recent convergence analysis work* [70, 10, 58, 17, 41, 29]*, they are still far from a satisfactory convergence theory for the DNN in contrast to the optimal convergence property of the FEM. Even so, we have seen that some advantages of the mesh-free DNN approach which is independent of and insensitive of the way of generating the sample points (training) set, thus a randomly distributed sample pints set on which the DNN's loss function is optimized by the SGD method can avoid the meshing procedure that especially remains as a challenging task for solving high-dimensional problems with complex geometrical domains. In addition, as shown in Section 3.1, the development of a stable and convergent FEM for the coupled KdV equations (1) is sophisticated and skillful, failing in developing a stable FEM for a highly hyperbolic system with optimal convergence is possible, unusually, without deep expertise on computational mathematics and long-term numerical trials. In contrast, the DNN's approach is designed for solving PDE problems in a uniform fashion, greatly reduces difficulties of the methodology development and computational costs that FEMs usually have to confront, which may make a real-time simulation/prediction possible with a relatively acceptable accuracy, in practice.*

On the other hand, we also conduct an efficiency comparison in terms of the CPU time cost by both the FEM and the DNN approach with the same and increasing space-time "resolution": [number of mesh nodes $\times$ number of time levels] for the FEM that equals [the size of sample-point training set] for the DNN approach. As illustrated in Table 4, we observe that CPU times of two approaches are not even on the same magnitude. Due to totally different methodology between two approaches, their differences in CPU time are so huge that such difference is amplified much further along with higher space-time "resolution". In fact, as long as a stable FEM equipped with an optimal convergence property is able to be developed for a PDE model bearing a relative low dimension, it is always more efficient than the DNN approach whose efficiency is restricted by the gradient descent optimization algorithm. However, this conclusion is quite opposite for a higher dimensional PDE model (whose dimension is greater than 3), for which traditional numerical methods (such as the FEM) fail in defining corresponding discretization forms, let alone to solve it numerically, whereas the DNN approach can still deal with high dimensional data in the training set with ease as it does for the low dimensional case.

5.2. FEM and DDN approach for solving the coupled KG equations. Consider a coupled system of KG equations (2) defined in a two-dimensional domain

TABLE 4. CPU time (second) cost by the FEM and the DNN approach for KdV equations.

(# of Mesh Nodes) × # of Time Levels ≡≡ # of Sample Points	FEM	DNN
$(10 \times 10) \times 5$	1	77
$(20 \times 20) \times 10$	3	758
$(40 \times 40) \times 20$	14	7428
$(80 \times 80) \times 40$	100	15072

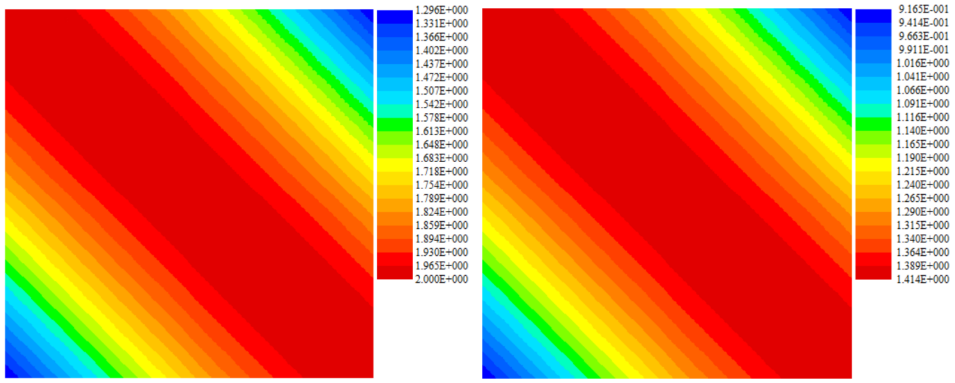
$\Omega = [0,1] \times [0,1]$ and the time interval $[0,1]$ with the following exact solution [11]

$$\begin{cases} u(\boldsymbol{x},t) = a_3 sech(\rho(x+y-\gamma t)), \\ v(\boldsymbol{x},t) = a_4 sech(\rho(x+y-\gamma t)), \end{cases} \tag{37}$$

where $a_3 = \sqrt{\frac{2a_1(c_1-b_2)}{b_1b_2-c_1c_2}}$, $a_4 = \sqrt{\frac{2a_3(c_2-b_1)}{b_2b_1-c_2c_1}}$ and $\rho = \sqrt{\frac{-a_1}{\gamma^2-2\kappa^2}} = \sqrt{\frac{-a_2}{\gamma^2-2\kappa^2}}$, while it is also employed as the initial condition at $t = 0$ and as the Dirichlet boundary condition on $\partial\Omega$. Clearly, such a Dirichlet boundary condition is nonhomogeneous that leads to a nonconservative energy. However, it does not affect the proposed FEM for the coupled KG equations to deliver optimally convergent results.

In this example, we take $a_1 = a_2 = 1, b_1 = -1, b_2 = -2, c_1 = 1, c_2 = 0.5$ and $\kappa = \gamma = 1$, and, we still employ the linear (P^1) finite element, i.e., $k = 1$, to construct finite element spaces W_h^0 and U_h for the developed FEM (14). Then we carry out Algorithm 3.2 to conduct a series of finite element approximations by using the same grid doubling as adopted in Section 5.1, and letting $\Delta t = h$ at different mesh levels. Numerical results at the terminal time $t = T = 1$ are displayed in Table 5 and Figure 5, from which we can see all convergence errors and convergence rates of u_h, v_h, p_h and b_h are optimal in their respective norms, namely, of second order in L^2 norm and of first order in H^1 norm, predicted as

$$\begin{aligned} &\|u^N - u_h^N\|_0 + \|v^N - v_h^N\|_0 + \|p^N - p_h^N\|_0 \\ &\quad + \|b^N - b_h^N\|_0 + h(\|u^N - u_h^N\|_1 + \|v^N - v_h^N\|_1) = O(h^2 + \Delta t^2). \end{aligned} \tag{38}$$



(a) FEM solution u_h (b) FEM solution v_h

FIGURE 5. Numerical results of the finite element solution for KG equations on the finest grid at $t = T$.

TABLE 5. Convergence results of the FEM for the coupled KG system at $t = T$.

h	1/10	1/20	1/40	1/80
$\|u - u_h\|_0$	6.799E-03	1.721E-03	4.315E-04	1.080E-04
Order		1.98	2.00	2.00
$\|u - u_h\|_1$	1.391E-01	6.952E-02	3.475E-02	1.738E-02
Order		1.00	1.00	1.00
$\|p - p_h\|_0$	1.301E-02	3.360E-03	8.469E-04	2.121E-04
Order		1.95	1.99	2.00
$\|v - v_h\|_0$	4.807E-03	1.217E-03	3.051E-04	7.634E-05
Order		1.98	2.00	2.00
$\|v - v_h\|_1$	9.839E-02	4.916E-02	2.457E-02	1.229E-02
Order		1.00	1.00	1.00
$\|b - b_h\|_0$	9.196E-03	2.376E-03	5.989E-04	1.500E-04
Order		1.95	1.99	2.00

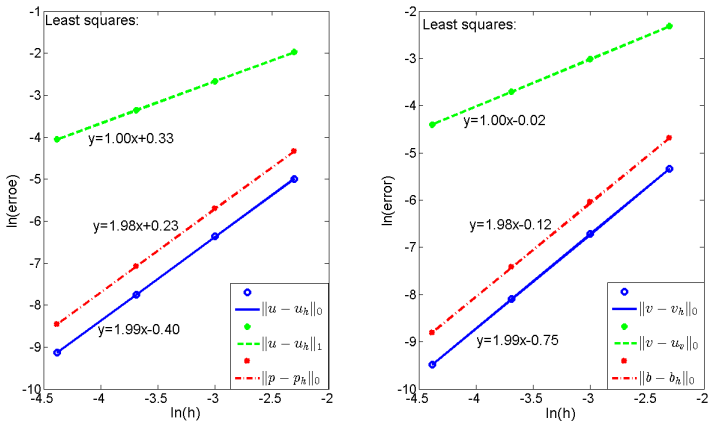


FIGURE 6. Convergence history of the finite element solution for KG equations at $t = T$.

On the other hand, to investigate the energy-preserving property of the proposed FEM for KG equations, we consider the coupled nonlinear KG equation (2) defined in $\Omega \times [0, 10]$ and equipped with the following homogeneous Dirichlet boundary conditions and initial conditions,

$$\begin{cases} u(\boldsymbol{x}, t) = 0, \ v(\boldsymbol{x}, t) = 0, & (\boldsymbol{x}, t) \in \partial\Omega \times (0, 10], \\ u(\boldsymbol{x}, 0) = 2\phi(\boldsymbol{x}), \ v(\boldsymbol{x}, 0) = \phi(\boldsymbol{x}), & \boldsymbol{x} \in \Omega, \\ u_t(\boldsymbol{x}, 0) = 0, \ v_t(\boldsymbol{x}, 0) = 0, & \boldsymbol{x} \in \Omega, \end{cases} \tag{39}$$

where $\phi(\boldsymbol{x}) = \sin(\pi x)\sin(\pi y)$. Discrete energies of finite element solutions are computed at different time steps with a fixed time step size and a fixed mesh size $\Delta t = h = 1/20$, and are illustrated in Table 6, where we can observe that the maximum errors between discrete energies and the initial energy among all time steps show an extremely small magnitude of 10^{-13}, validating that the developed FEM can efficiently preserve discrete energies for the coupled system of nonlinear KG equations all the time.

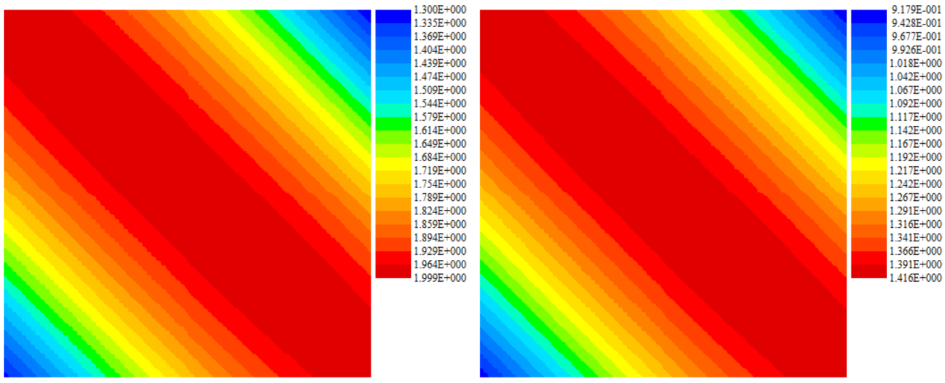
TABLE 6. Discrete energies' change trend of the FEM for the coupled KG system.

t	n	E_h^n	$\lvert E_h^n - E_h^0\rvert$
t=0s	0	15.102856085836247502	0.00E+00
t=1s	20	15.102856085836155131	9.24E-14
t=2s	40	15.102856085836180000	6.75E-14
t=3s	60	15.102856085836160460	8.70E-14
t=4s	80	15.102856085836165789	8.17E-14
t=5s	100	15.102856085836165789	8.17E-14
t=6s	120	15.102856085836142697	1.05E-13
t=7s	140	15.102856085836060984	1.87E-13
t=8s	160	15.102856085836158684	8.88E-14
t=9s	180	15.102856085836132038	1.15E-13
t=10s	200	15.102856085836060984	1.87E-13

Next, we apply the mesh-free DNN approach as described in Section 4.2 to solve the coupled KG equations with the same exact solution (37) defined in the same (2+1)-dimensional space-time domain with the same parameters. So, the same doubling sample point (training) sets are uniformly constructed as done in Section 5.1. In addition, we employ the same stochastic gradient descent method, ADAM, with an initial learning rate of 0.001 and 2×10^4 epochs to solve the minimization problem (33) formed by the DNN approximation and the LS-formulated loss functional (28).

On the finest sample points set in space that is the same finest grid points set of the FEM, Figure 7 shows contour results of the DNN's solution $(\mathcal{U}_{\mathbf{NN}}, \mathcal{V}_{\mathbf{NN}})$ for the coupled KG equations at the highest time level $t_N = T = 1$, which is comparable with that of the finite element solution (u_h, v_h) shown in Figure 5. Again, the difference between FEM's and DNN's solutions is almost invisible, measured by $\|u_h^N - \mathcal{U}_{\mathbf{NN}}^N\|_{L^2(\Omega))} + \|v^N - \mathcal{V}_{\mathbf{NN}}^N\|_{L^2(\Omega))} \approx 6.68 \times 10^{-3}$ after exactly computing the total error between solutions of FEM and DNN on the finest sample points set at $t_N = T = 1$, which holds the same magnitude with DNN's approximation errors over the sample set doubling shown in Table 7 or over the DNN's structure doubling shown in Table 8.

In other words, along with the doubling size of the sampling point set M while fixing the DNN structure as 4 hidden layers with 10 neurons in each layer, Table 7 illustrates total approximation errors for the same estimator, $|||e_{uv}|||$, of the proposed DNN approach for the coupled KG equations (2). And, Table 8 displays total errors of DNN's solutions with the fixed size of training set $M = 32000$ while doubling the number of neurons and the number of hidden layers of the neural network. Both tables show DNN's approximation errors with almost the same magnitude, 10^{-3}. Same numerical phenomenon with that of DNN for KdV equations, they do not show a clear convergence pattern in terms of either the sample set size or the number of neurons/layers of the DNN, for the same reason as explained in Remark 5.1. In addition, the CPU-time comparison cost by both the FEM and the DNN approach for KG equations is shown in Table 9, where the same phenomenon can be observed as that of KdV equations, i.e., the FEM is much more efficient than the DNN approach based upon the same space-time "resolution" for a low dimensional PDE model such as the targeted KG system, and, such an efficiency difference



(a) DNN solution u_h (b) DNN solution v_h

FIGURE 7. Numerical results of the DNN's solution for KG equations on the finest sample points set at $t = T$.

between two approaches is even huger when the space-time "resolution" increases further.

TABLE 7. Approximation errors of DNN's solutions with 4 layers $\times$ 10 neurons for KG equations.

M	$\|\|\|e_{uv}\|\|\|$
$10 \times 10 \times 5$	1.047×10^{-3}
$20 \times 20 \times 10$	1.289×10^{-3}
$40 \times 40 \times 20$	1.999×10^{-3}
$80 \times 80 \times 40$	2.742×10^{-3}

TABLE 8. Approximation errors of various DNN's solutions with $M = 40 \times 40 \times 20$ for KG equations.

# of Neurons	1 Layer	2 Layers	4 Layers
4	1.438×10^{-3}	9.422×10^{-4}	1.288×10^{-3}
8	1.280×10^{-3}	2.356×10^{-3}	2.058×10^{-3}
16	3.520×10^{-3}	2.369×10^{-3}	5.776×10^{-3}
32	6.290×10^{-3}	2.593×10^{-3}	9.201×10^{-3}

In addition, to compare with the energy-preserving FEM, discrete energies of DNN's solutions are also computed at different time levels based upon a fixed DNN structure, 4 layers $\times$ 50 neurons, and a fixed $(2+1)$-dimensional sample point set with $M = 21 \times 21 \times 201$ in the space-time domain $\Omega \times [0, T] := [0, 1] \times [0, 1] \times [0, 10]$, where sample-point sets are uniformly distributed in time with the time step size $1/200$ and in space with an equal spatial size $1/20$ in the form of a checkerboard at each time level, just chosen as the finest finite element grid for the comparison purpose. Table 10 illustrates the change trend of DNN's discrete energies in time with 1 second as the increment, which shows a definite energy dissipation process from the initial time (0s) all the way to the terminal time (10s), completely different

TABLE 9. CPU time (second) cost by the FEM and the DNN approach for KG equations.

(# of Mesh Nodes) × # of Time Levels ≡ # of Sample Points	FEM	DNN
$(10 \times 10) \times 5$	3	93
$(20 \times 20) \times 10$	8	596
$(40 \times 40) \times 20$	54	29344
$(80 \times 80) \times 40$	471	204734

from discrete energies' change trend of the FEM shown in Table 6 that is basically conserved all the time.

TABLE 10. Discrete energies' change trend of the DNN approach for the coupled KG system.

t	n	E_h^n	$\lvert E_h^n - E_h^0 \rvert$
t=0s	0	1.43E+01	0.00E+00
t=1s	20	5.92E+00	8.42E+00
t=2s	40	8.20E-01	1.35E+01
t=3s	60	2.55E-02	1.43E+01
t=4s	80	2.18E-03	1.43E+01
t=5s	100	5.12E-04	1.43E+01
t=6s	120	1.23E-04	1.43E+01
t=7s	140	1.75E-05	1.43E+01
t=8s	160	1.69E-05	1.43E+01
t=9s	180	1.90E-05	1.43E+01
t=10s	200	1.70E-05	1.43E+01

Therefore, again, although the DNN's approach for solving the coupled KG equations is less sophisticated in contrast to the subtle development of an energy-preserving FEM for the same KG equations, the energy conservation property is lost from DNN's solutions. In that sense, the DNN's approach just likes a direct finite element discretization for the KG equations as described in Section 3.2. Even so, the DNN's approach still does not hold a convergence pattern. But in a scenario that the energy conservation is not a big concern, the DNN's approach can always deliver a fast solution in a short time of development with ease and an acceptable accuracy for nonlinear hyperbolic/wave equations without confronting challenges of grid generation for complex domains in high dimension, as commented in Remark 5.1.

6. Conclusions

For the purpose of a comparative study, two numerical approaches, finite element method (FEM) and deep neural network approach (DNN) are studied for two types of coupled nonlinear hyperbolic/wave system, KdV equations and KG equations. Both approaches have their own advantages and disadvantages when solving two nonlinear hyperbolic systems, respectively. Overall, a sophisticatedly developed FEM can produce a stable, efficient and optimal convergence for both hyperbolic problems that are defined in low dimension, and can even preserve the total energy for the coupled KG equations. Such a subtle development of FEM for

complex nonlinear hyperbolic problems, even they are modeled by low-dimensional PDEs, is usually difficult and time-consuming, let alone high-dimensional PDEs to which traditional numerical methods (such as the FEM) fail in defining corresponding discretization forms in the first place. As a contrast, the uniformly designed DNN's approach can only produce a universal approximation to both hyperbolic problems without holding a distinct convergence order, whereas its development does not require high expertise on theories and implementations of numerical PDEs. Moreover, the DNN's approach belongs to the category of mesh-free methods that circumvents the meshing procedure that especially remains as a challenging task for solving high-dimensional problems with complex geometrical domains, greatly reduces difficulties of the methodology development and computational costs that traditional numerical methods (e.g., FEMs) usually have to confront and sometimes are even unable to deal with, making the DNN's approach to possibly handle real-time simulations/predictions using practical high-dimensional data with ease.

Acknowledgments

X. Zhu was partially supported by National Science Foundation of China (NSFC) (No. 51809026) and China Scholarship Council (CSC) (No. 201908530069). M. He was partially supported by Natural Science Foundation of Zhejiang Province, China (No. LY21A010011 and No. LQ19A010009). P. Sun was partially supported by a grant from the Simons Foundation (MPS-706640, PS).

References

[1] T Alagesan, Y Chung, and K Nakkeeran. Soliton solutions of coupled nonlinear Klein-Gordon equations. Chaos, Solitons & Fractals, 21(4):879–882, 2004.

[2] A. Aly, G. Guadagni, and J. B. Dugan. Derivative-free optimization of neural networks using local search. In 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON), pages 0293–0299, 2019.

[3] D. N. Arnold and R. Winther. A superconvergent finite element method for the Korteweg-de Vries equation. Mathematics of Computation, 38(157):23–36, 1982.

[4] J. Bellazzini, V. Benci, C. Bonanno, and A. M. Micheletti. Solitons for the nonlinear Klein-Gordon equation. Advanced Nonlinear Studies, 10(2):481–499, 2010.

[5] M. Boiti, JJ-P Leon, M. Manna, and F. Pempinelli. On the spectral transform of a Korteweg-de Vries equation in two spatial dimensions. Inverse problems, 2(3):271–279, 1986.

[6] P. Caudrey, J. Eilbeck, and J. Gibbon. The sine-Gordon equation as a model classical field theory. Il Nuovo Cimento B (1971-1996), 25(2):497–512, 1975.

[7] Jen-Tzung Chien. Chapter 4 - independent component analysis. In J.-T. Chien, editor, Source Separation and Machine Learning, pages 99–160. Academic Press, 2019.

[8] R. Côte and C. Muñoz. Multi-solitons for nonlinear Klein-Gordon equations. In Forum of Mathematics, Sigma, volume 2. Cambridge University Press, 2014.

[9] D. G. Crighton. Applications of KdV, pages 39–67. Springer Netherlands, 1995.

[10] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear approximation and (Deep) ReLU networks. arXiv:1905.02199 [cs], 2019.

[11] D. Deng and D. Liang. The energy-preserving finite difference methods and their analyses for system of nonlinear wave equations in two dimensions. Applied Numerical Mathematics, 151:172–198, 2020.

[12] D. Deng and Q. Wu. The studies of the linearly modified energy-preserving finite difference methods applied to solve two-dimensional nonlinear coupled wave equations. Numerical Algorithms, 88(4):1875–1914, 2021.

[13] T. P. Dinh, S. Wang, and A. Yassine. Training multi-layered neural network with a trust-region based algorithm. RAIRO, 24:523–553, 1990.

[14] M. Dissanayake and N. Phan-Thien. Neural-network-based approximations for solving partial differential equations. communications in Numerical Methods in Engineering, 10(3):195–201, 1994.

[15] R.K. Dodd, H.C. Morris, J.C. Eilbeck, and J.D. Gibbon. Soliton and nonlinear wave equations. London and New York, 1982.
[16] G. dos Reis, S. Engelhardt, and G. Smith. Simulation of McKean Vlasov SDEs with super linear growth. arXiv:1808.05530 [math.PR], 2021.
[17] C. Duan, Y. Jiao, Y. Lai, X. Lu, and Z. Yang. Convergence rate analysis for deep Ritz method. arXiv:2103.13330 [math.NA], 2021.
[18] R. M. Dudley. The speed of mean Glivenko-Cantelli convergence. The Annals of Mathematical Statistics, 40:40–50, 1969.
[19] R. Dutta, U. Koley, and N.H. Risebro. Convergence of a higher order scheme for the Korteweg-De Vries equation. SIAM Journal on Numerical Analysis, 53(4):1963–1983, 2015.
[20] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat., 5:349–380, 2017.
[21] W. E and B. Yu. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 6:1–12, 2018.
[22] F. E.-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer. An introductory review of deep learning for prediction models with big data. Front. Artif. Intell., 3, Article 4:1–23, 2020.
[23] S.M. El-Sayed. The decomposition method for studying the Klein-Gordon equation. Chaos, Solitons & Fractals, 18(5):1025–1030, 2003.
[24] P.G. Estévez and S. Leble. A wave equation in 2+1: Painlevé analysis and solutions. Inverse Problems, 11(4):925–937, 1995.
[25] N. Fournier and A. Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. Probability Theory and Related Fields, 162:707–738, 2015.
[26] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT press, 2016.
[27] J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. In Proceedings of the National Academy of Sciences, volume 115, pages 8505–8510, 2018.
[28] M. He and P. Sun. Energy-preserving finite element methods for a class of nonlinear wave equations. Applied Numerical Mathematics, 157:446–469, 2020.
[29] Q. Hong, J.W. Siegel, and Jinchao Xu. A priori analysis of stable neural network solutions to numerical PDEs. arXiv:2104.02903 [math.NA], 2021.
[30] P. Huang. Convergence of a full discrete finite element method for the Korteweg-de Vries equation. Portugaliae Mathematica, 77(1):31–43, 2020.
[31] V. Imaikin, A. Komech, and P.A. Markowich. Scattering of solitons of the Klein-Gordon equation coupled to a classical particle. Journal of Mathematical Physics, 44(3):1202–1217, 2003.
[32] A.D. Jagtap, E. Kharazmi, and G.E. Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Comput. Methods Appl. Mech. Engrg., 365:113028, 2020.
[33] A. Karczewska, P. Rozmej, M. Szczeciński, and B. Boguniewicz. Finite element method for extended KdV equations. arXiv preprint arXiv:1604.04105, 2016.
[34] D. Kaya and S.M. El-Sayed. A numerical solution of the Klein-Gordon equation and convergence of the decomposition method. Applied mathematics and computation, 156(2):341–353, 2004.
[35] K.R. Khusnutdinova, Y. Stepanyants, and M.R. Tranter. Soliton solutions to the fifth-order Korteweg-de Vries equation and their applications to surface and internal water waves. Phys. Fluids, 30:022104, 2018.
[36] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2015.
[37] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG], 2017.
[38] Y. Liao and P. Ming. Deep Nitsche method: Deep Ritz method with essential boundary conditions. Commun. Comput. Phys., 29:1365–1384, 2021.
[39] S. Lou. Generalized dromion solutions of the (2+ 1)-dimensional KdV equation. Journal of Physics A: Mathematical and General, 28(24):7227, 1995.
[40] S.-Y. Lou and X.-B. Hu. Infinitely many lax pairs and symmetry constraints of the KP equation. Journal of Mathematical Physics, 38(12):6401–6427, 1997.
[41] J. Lu, Y. Lu, and M. Wang. A priori generalization analysis of the deep Ritz method for solving high dimensional elliptic equations. arXiv:2101.01708 [math.NA], 2021.

[42] J.C. Luke. Interaction of solitons with the electromagnetic field in classical nonlinear field models. Complexity in Biological and Physical Systems: Bifurcations, Solitons and Fractals, page 113, 2018.
[43] S. Mandt, M. D. Hoffman, and D. M. Blei. Stochastic gradient descent as approximate bayesian inference. Journal of Machine Learning Research, 18:1–35, 2017.
[44] X. Meng, Z. Li, D. Zhang, and G.E. Karniadakis. PPINN: Parareal physics-informed neural network for time-dependent PDEs. Comput. Methods Appl. Mech. Engrg., 370:113250, 2020.
[45] P. A. Papadopoulos. Solitary waves and nonlinear Klein-Gordon equations. PhD thesis, University of New Hampshire, 1988.
[46] Yan-Ze Peng. Exact periodic and solitary waves and their interactions for the (2+ 1)-dimensional KdV equation. Physics Letters A, 351(1-2):41–47, 2006.
[47] A. Pinkus. Approximation theory of the MLP model in neural networks. Acta Numerica, 8:143–195, 1999.
[48] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378:686–707, 2019.
[49] M. Raissi, A. Yazdani, and G.E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science, 367(6481):1026–1030, 2020.
[50] V. Ranganathan and A. Lewandowski. ZORB: A derivative-free backpropagation algorithm for neural networks. arXiv:2011.08895 [cs.LG], 2020.
[51] Y.-J. Ren. Korteweg–de Vries equation (KdV), different analytical methods for solving the. In Mathematics of complexity and dynamical systems. 2012.
[52] M. Rezapour and T. Asaki. Neural-trust-region algorithm for unconstrained optimization. arXiv:2004.09058 [math.OC], 2020.
[53] H. Robbins and S. Monro. A stochastic approximation method. Annals of Mathematical Statistics, 22:400–407, 1951.
[54] S. Ruder. An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs.LG], 2017.
[55] R. Sassaman, M. Edwards, F. Majid, and A. Biswas. 1-soliton solution of the coupled nonlinear Klein-Gordon equations. Studies in Mathematical Sciences, 1(1):30–37, 2010.
[56] I. Segal. Nonlinear partial differential equations in quantum field theory. In Proc. Symp. Appl. Math. AMS, volume 17, pages 210–226, 1965.
[57] H. Sheng and C. Yang. PFNN: a penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. Journal of Computational Physics, 428:110085, 2021.
[58] J.W. Siegel and J. Xu. Approximation rates for neural networks with general activation functions. Neural Networks, 128:313–321, 2020.
[59] J. Sirignano and K. Spiliopoulos. DGM: a deep learning algorithm for solving partial differential equations. Journal of Computational Physics, 375:1339–1364, 2018.
[60] M. Song, B.S. Ahmed, and A. Biswas. Topological soliton solution and bifurcation analysis of the Klein-Gordon-Zakharov equation in-Dimensions with power law nonlinearity. Journal of Applied Mathematics, 2013, 2013.
[61] M. Song, Z. Liu, E. Zerrad, and A. Biswas. Singular solitons and bifurcation analysis of quadratic nonlinear Klein-Gordon equation. Applied Mathematics & Information Sciences, 7(4):1333, 2013.
[62] M. Tajiri. On soliton solutions of the nonlinear coupled Klein-Gordon equation. Journal of the Physical Society of Japan, 52(11):3722–3726, 1983.
[63] J. Weed and F. Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. Bernoulli, 25:2620–2648, 2019.
[64] J. Wu and X. Geng. Inverse scattering transform and soliton classification of the coupled modified Korteweg-de Vries equation. Commun. Nonlinear Sci. Numer. Simul., 53:83–93, 2017.
[65] K. Xu and E. Darve. Trust region method for coupled systems of pde solvers and deep neural networks. arXiv:2105.07552 [math.NA], 2021.
[66] K.D. Yamada. Derivative-free neural network for optimizing the scoring functions associated with dynamic programming of pairwise-profile alignment. Algorithms Mol Biol, 13:5, 2018.
[67] J. Yan and C.-W. Shu. A local discontinuous galerkin method for KdV type equations. SIAM Journal on Numerical Analysis, 40(2):769–791, 2002.

[68] Y. Zang, G. Bao, X. Ye, and H. Zhou. Weak adversarial networks for high-dimensional partial differential equations. Journal of Computational Physics, 411:109409, 2020.
[69] S. Zhang. A generalized auxiliary equation method and its application to (2+1)-dimensional Korteweg-de Vries equations. Computers & Mathematics with Applications, 54(7-8):1028–1038, 2007.
[70] D.X. Zhou. Deep distributed convolutional neural networks: Universality. Analysis and Applications, 16:895–919, 2018.

School of Mathematics and Computer Sciences, Dali University, 2 Hongsheng Street, Dali, Yunnan 671003, China
E-mail: zxw4688@126.com

School of Sciences, Hangzhou Dianzi University, Hangzhou, Zhejiang, China
E-mail: hemingyan@hdu.edu.cn

Department of Mathematical Sciences, University of Nevada, Las Vegas, 4505 Maryland Parkway, Las Vegas, Nevada 89154, USA
E-mail: pengtao.sun@unlv.edu