Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **16**, No. 4, pp. 833-859

DOI: 10.4208/aamm.OA-2022-0278
August 2024

Cell-Average Based Neural Network Method for Hunter-Saxton Equations

Chunjie Zhang[1], Changxin Qiu[1,*], Xiaofang Zhou[1]
and Xiaoming He[2]

[1] *School of Mathematics and Statistics, Ningbo University, Ningbo, Zhejiang 315211, China*

[2] *Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO 65409, USA*

Received 1 November 2022; Accepted (in revised version) 3 July 2023

Abstract. In this paper, we develop a cell-average based neural network (CANN) method for solving the Hunter-Saxton equation with its zero-viscosity and zero-dispersion limits. Motivated from the finite volume schemes, the cell-average based neural network method is constructed based on the finite volume integrals of the original PDEs. Supervised training is designed to learn the solution average difference between two neighboring time steps. The training data set is generated by the cell average based on a single initial value of the given PDE. The training process employs multiple time levels of cell averages to maintain stability and control temporal accumulation errors. After being well trained based on appropriate meshes, this method can be utilized like a regular explicit finite volume method to evolve the solution under large time steps. Furthermore, it can be applied to solve different type of initial value problems without retraining the neural network. In order to validate the capability and robustness of the CANN method, we also utilize it to deal with the corrupted learning data which is generated from the Gaussian white noise. Several numerical examples of different types of Hunter-Saxton equations are presented to demonstrate the effectiveness, accuracy, capability, and robustness of the proposed method.

AMS subject classifications: 35E15, 65M15, 68T07

Key words: Finite volume scheme, cell-average based neural network, Hunter-Saxton equation, corruption data.

1 Introduction

The Hunter-Saxton (HS) equation [1,2] is a nonlinear wave equation which has been used to describe waves in a massive director field that propagate in nematic liquid crystals

[*] Corresponding author.

Emails: 2111071049@nbu.edu.cn (C. Zhang), qiuchangxin@nbu.edu.cn (C. Qiu), 2111071051@nbu.edu.cn (X. Zhou), hex@mst.edu (X. He)

when certain molecules move and cause interference. The HS equation under consideration here is given as

$$u_{xxt}+2u_xu_{xx}+uu_{xxx}=0. \tag{1.1}$$

If we consider the viscosity and dispersion [3,4], we have the corresponding regularization with viscosity:

$$u_{xxt}+2u_xu_{xx}+uu_{xxx}-\varepsilon_1 u_{xxxx}=0, \tag{1.2}$$

and the corresponding regularization with dispersion:

$$u_{xxt}+2u_xu_{xx}+uu_{xxx}-\varepsilon_2 u_{xxxxx}=0, \tag{1.3}$$

where $\varepsilon_1 \geq 0$ and ε_2 are small constants. In the past two decades, this equation has attracted extensive attention because of its rich mathematical structure and properties. Therefore it is not surprising that many different numerical methods have been proposed and analyzed for the HS equation, including the finite difference method [5], local discontinuous Galerkin (LDG) method [6, 7], collocation method [8–10], collocation finite element method [11], quasilinearization method [12, 13] and others [14–16]. However, compared with the extensively studied classical numerical methods, the HS equation is still in great need of efforts for developing and analyzing the stable, accurate, and efficient numerical neural network methods, especially for peakon solutions.

Recently, there has been increasing interest in developing neural networks to solve partial differential equations (see, e.g., [17–30]). Neural networks generate a wide range of functions by combining linear transformations and activation functions. One of the most notable characteristics of neural networks is that they do not require a hand-crafted geometric mesh or point cloud, as do the traditional, well-studied finite difference, finite volume, and finite element methods. According to their basic technique and core goal, the neural network methods for solving PDEs can be roughly divided into two categories.

One group is to design neural network methods and apply the neural network methods to solve many types of partial differential equations. These neural network methods take x and t as the network input vectors, and have the advantages of automatic differentiation and mesh free, including the early work [31], the popular physics-informed neural network (PINN) methods [26,32–34], and many others [22,35–38]. Moreover, PIELM [39] is proposed to improve the speed of PINN in a larger domain for practical problems. In [40], PINNs are used to directly encode the control equations into the deep neural network through automatic differentiation to overcome the limitations of incompressible laminar flow and turbulence. In [41–43], weak formulations are applied in the loss function, instead of the PDEs explicitly enforced on collocation points. We further mention the works of [44] and [45], for which method of lines are explored with Fourier basis and Residual networks applied to evolve the dynamical system.

The second group is to apply neural network to improve the existing numerical methods. In [46], a multilayer perceptron (MLP) is constructed to identify troubled-cells. In [47], the authors use (deep) Reinforcement learning to learn the new solvers for conservation laws. In [48], an MLP network is designed to estimate the artificial viscosity in