Numerical solution of Generalized Burger-Huxley & Huxley's equation using Deep Galerkin neural network method

Harender Kumar[a], Neha Yadav[a*], Atulya Nagar[b]

[a,a*] *Department of Mathematics & Scientific Computing*
National Institute of Technology Hamirpur
Hamirpur-177001, H.P.
India

[b] *School of Mathematics,*
Computer Science and Engineering,
Liverpool Hope University, United Kingdom

Abstract

In this paper, a deep learning algorithm based on Deep Galerkin method (DGM) is presented for the approximate solution of the generalized Burgers-Huxley equation (gBHE), and generalized Huxley's equation (gHE). In this method, a deep neural network (DNN) is used for approximating the solution without generating mesh grid, which satisfies the differential operator, boundary and initial conditions. DNN is trained on randomly selected batches of time and space points, thus helping to avoid forming a mesh. Adam optimizer is used for optimizing the parameters of the DNN. Further, the convergence of the cost function and convergence of the neural network to the exact solution is demonstrated. This method shows very encouraging results which have been compared with recent methods such as: A fourth order improved numerical scheme(FDS4), Adomain-decomposition method (ADM), Modified cubic B-spline differential quadrature method (MCB- DQM), Variational iteration method(VIM), and others.

Keywords: Deep learning, Deep neural network, GRU network, Adam optimizer, Generalized Burger-Huxley & Huxley's equation

*corresponding author
Email address: nyadav@nith.ac.in (Neha Yadav[a])

1. Introduction

Nonlinear partial differential equations (NPDEs) are used to model the majority of physical phenomenon that arises in numerous sectors of science and engineering. The gBHE is one of the well-known NPDE. It describes the interaction between the reaction mechanism, the convective effect, and diffusion transport . It came into existence due to the joint efforts of Bateman [1, 2] and Burger [3] for the Burger equation, and Hodgkin and Huxley [4] for the Huxley equation. H. Bateman first proposed the Burger equation in 1915, and Johannes M. Burgers explored it in 1948. It is the most straightforward paradigm for comprehending the physical features of phenomenon such as hydrodynamic turbulence, vorticity transportation, heat conduction, wave processes in thermoelastic medium, elasticity, mathematical modeling of turbulent fluid, gas dynamics, sound and shock wave theory, dispersion in porous media, and so on. A. Hodgkin and A. Huxley in 1952, proposed a model to explain the ionic mechanisms underlying the initiation and propagation of action potentials in the squid giant axon, and they got the Nobel Prize in Physiology in 1963 for this model. Satsuma et al.[5] discovered applications of gBHE in biology, combustion, chemistry, nonlinear acoustics, mathematics and engineering, metallurgy, in 1986.
Let $\mathrm{D} \subset \mathrm{R}^d$ be a bounded set and $\partial\mathrm{D}$ denotes its boundary, and assume $\mathrm{D}_T = \mathrm{D} \times (0, T]$ and $\partial\mathrm{D}_T = \partial\mathrm{D} \times (0, T]$. The most general form of the gBHE [6] is defined as follows:

$$\mathcal{L}[\Psi(x,t)] = \Psi_t - \Psi_{xx} + p\,\Psi^s\,\Psi_x + q\,\Psi\,\left(\Psi^s - 1\right)\left(\Psi^s - r\right) = 0, \quad (x,t) \in \mathrm{D}_T \tag{1}$$

with the initial and boundary conditions

$$\begin{cases} \Psi(x,0) = \left(\frac{r}{2} + \frac{r}{2}\tan h\,[B_1 x]\right)^{1/s} = \Psi_0(x)(\text{say}), & x \in \mathrm{D} \\ \Psi(0,t) = \left(\frac{r}{2} + \frac{r}{2}\tanh\,[-B_1 B_2 t]\right)^{1/s} = f_1(x,t)(\text{say}), & (x,t) \in \partial\mathrm{D}_T \\ \Psi(1,t) = \left(\frac{r}{2} + \frac{r}{2}\tanh\,[B_1\,(1 - B_2 t)]\right)^{1/s} = f_2(x,t)(\text{say}), & (x,t) \in \partial\mathrm{D}_T \end{cases} \tag{2}$$

The exact solution of gBHE i.e., Eq. (1) - Eq. (2) is given by Eq.(3) :

$$\Psi(x,t) = \left(\frac{r}{2} + \frac{r}{2}\tan h\,[B_1\,(x - B_2 t)]\right)^{1/s} \tag{3}$$

where

$$\begin{cases} B_1 = \frac{-p\,s + s\,\sqrt{p^2+4q(1+s)}}{4(1+s)} \\ \& \\ B_2 = \frac{r\,p}{1+s} - \frac{(1+s-r)\left(-p+\sqrt{p^2+4q(1+s)}\right)}{2(1+s)} \end{cases} \tag{4}$$

where, $p > 0$ represents advection coefficient, $q \geqslant 0$ represents reaction coefficient, $r \in (0, 1)$ and $s > 0$ are real coefficients, while Ψ_{xx} is the diffusive term, $\Psi^s\Psi_x$ is the advection term, and $\Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)$ is the reaction term.

At $s = 1, p = 0$, Eq. (1) becomes the Huxley equation which describe wall motion in liquid crystals and nerve pulse propagation in nerve fibres [7]:

$$\Psi_t - \Psi_{xx} + q\Psi(\Psi - 1)(\Psi - r) = 0 \tag{5}$$

At $s = 1, q = 0$, Eq. (1) becomes the Burger's equation which describe the far field of wave propagation in non-linear dissipative systems [2].

$$\Psi_t - \Psi_{xx} + p\Psi\Psi_x = 0 \tag{6}$$

Non-linear diffusion equations such as Eq. (5) and Eq. (6) are well-known in non-linear physics. Eq. (1) becomes the modified Burger's equation when $q = 0$, and becomes the Burgers-Huxley equation (BHE) at $s = 1$ and $p \neq 0, q \neq 0$ as:

$$\Psi_t - \Psi_{xx} + p\Psi\Psi_x + q\Psi(\Psi - 1)(\Psi - r) = 0 \tag{7}$$

and finally at $p = 0$, Eq. (1) becomes the Fitzhugh-Nagoma equation [8, 4, 9] respectively.

Many researchers have made several attempts in recent years to obtain analytical and numerical solutions for gBHE. By using nonlinear transformation, Wang et al. [10] finds the kink wave and solitary solutions of the gBHE. Ismail et al. [6] and Hashim et al. [11, 12] used the Adomian decomposition method (ADM), Wazwaz [13] used the tanh-coth method, Bataineh et al. [14] used the homotopy analysis method, and Khami and Molabahrami [15] used the homotopy analysis method to find the solitary wave solution of BHE. Batiha et al [16] used the variational iteration method (VIM), Efimova and Kudryashov [17] used Hope-Cole transformation, and Gao and Zhao [18] used He's Exp-function method to find the traveling wave solutions of the gBHE, Griffiths and Schiesser [19] presented a traveling wave analysis for the BHE.

For obtaining the approximation solution of gBHE over the variety of the domain, several types of approaches have been devised such as a fourth-order finite difference scheme (FDS4) [20], Hybrid B-Spline Collocation Method [21], Chebyshev spectral collocation with the domain decomposition [22], high order finite difference schemes [23], domain decomposition algorithm based on Chebyshev polynomials (DDAC) [24], differential quadrature method (DQM)[25], optimal Homotopy asymptotic method (OHAM) [26], Homotopy analysis method [15], B-spline collocation method [27]. In higher dimensions, many of these methods, particularly grid-based methods such as FDS4 [20], Chebyshev spectral collocation with the domain decomposition [22], DDAC [24], and others are plagued by concerns of instability and computing expense. Other efficient numerical methods are also developed in recent literature for various type of differential equation problems [28, 29, 30, 31, 32, 33].

Several machine learning-based methods have been proposed in recent years to address the issue of large dimensionality faced by mesh-based methods [34, 35, 36, 37]. Recently, Sirignano and Spiliopoulos [38] presented a mesh-free deep learning algorithm called 'Deep-Galerkin-Method(DGM)' to get approximate solutions of high-dimensional PDEs. The Galerkin technique is a widely used numerical approach for finding a reduced-form solution to a PDE by combining basis functions in a linear combination. DGM is similar to the Galerkin method, with few significant differences based on machine learning approaches. In DGM, compare to the Galerkin method, linear combination of basis functions is replaced by a DNN. In this method, there is no need to generate a mesh since the random sampling technique is used for generating spatial points. At randomly sampled spatial points, the stochastic gradient descent (SGD) technique is used to train DNN so that it satisfies the differential operator, initial and boundary conditions. Galerkin's method and machine learning come together naturally in DGM. Because of its simple and uncomplicated implementation, DGM approach has gotten a lot of attention.

Keeping in view the application/importance of gBHE and advantages of DGM algorithms the goal of this study is to obtain the approximate solution of the gBHE using DGM and a different type of architecture that is comparable to the Gated Recurrent Unit(GRU) network[39], without using Monte Carlo Method. GRU is an advanced version of the standard recurrent neural network (RNN) or it may be considered as a refined version of the Long-Short-Term-Memory(LSTM) network[40], this seeks to tackle the vanishing gradient problem that comes with standard RNN. The main difference between GRU and LSTM is that while LSTM has three gates: input, output, and forget, GRU only has two gates: reset and update. GRU is simpler than LSTM since it contains fewer gates. Unlike an LSTM, a GRU does not have an output gate or any internal memory, thus uses fewer training parameters, less memory, and operates more quickly than LSTM. In the proposed method given nonlinear PDE is converted into a machine learning problem using a cost function based on the L2-norm error function, and an approximating function given by a DNN is employed to approximate the unknown solution.The suggested method is easy to use and implement, and it provides an approximate solution for any value in the solution domain. The suggested method's efficiency and reliability are demonstrated by successful solution of gBHE and Huxley's equations. Convergence analysis of cost function and convergence of neural network to the gBHE solution is also discussed.

Further in this paper, we describe the Methodology in Section 2., and implementation details for the algorithm in Section 2.1. In Section 3., we describe the Convergence of the method, and Numerical results and Discussion are described in Section 4. Finally, we analyze our findings in the Conclusions part in Section 5.

2. Methodology

In this section, we present the methodology of DGM to approximate the solution of gBHE. Recall the form of gBHE from Eq(1, 2), i.e.,

$$\begin{cases} \mathcal{L}[\Psi(x,t)] = 0, \quad (x,t) \in \mathrm{D}_T \\ \Psi(x,0) = \Psi_0(x), \quad x \in \mathrm{D} \\ \Psi(x,t) = f(x,t),\ (x,t) \in \partial\mathrm{D}_T \end{cases} \tag{8}$$

where Ψ be a function of space (x) and time (t) defined on the region D_T, and $x \in \mathrm{D} \subset \mathcal{R}^d$. By using DGM, our aim is to approximates $\Psi(x,t)$ with approximating function $\widehat{\Psi}(x,t;\theta)$ given by a DNN, where $\theta \in R^k$ are the parameter of DNN. Firstly, we construct a cost function as follows:

$$C(\widehat{\Psi}) = \underbrace{\|\mathcal{L}[\hat{\Psi}(x,t;\theta)]\|^2_{\mathrm{D}_T,\rho_1}}_{differential\ operator} + \underbrace{\left\|\widehat{\Psi}(x,t;\theta) - f(x,t)\right\|^2_{\partial\mathrm{D}_T,\rho_2}}_{boundary\ condition} + + \underbrace{\left\|\widehat{\Psi}(x,0;\theta) - \Psi_0(x)\right\|^2_{\mathrm{D},\rho_3}}_{initial\ condition} \tag{9}$$

Here, $\|g(w)\|^2_{\omega,\rho} = \int_\omega |g(w)|^2 \rho(w) dw$ i.e., we use L^2-norm function to calculate the error in above Eq.(9) with $\rho(r)$ is a density defined on ω. Now, we have to find a set of parameter 'θ' of the functions $\widehat{\Psi}(x,t;\theta)$ s.t. it minimizes the cost function $C(\widehat{\Psi})$. If the optimized value of $C(\widehat{\Psi})$ is very small then the approximating function $\widehat{\Psi}(x,t;\theta)$ will close to satisfied the PDE's differential operator, boundary and initial condition. If we directly integrate on D_T to determine the cost function's minimum, the calculation grows exponentially as the dimension d grows. To avoid the problem described above, we use the DGM instead of typical numerical techniques and this avoids the necessity of forming a mesh. The DGM is delivered in the following format and consists of the following five steps:

Step 1. Initialize the parameter set θ_0 and the learning rate α_n.

Step 2. Generate random points $R_n = \{(X_n, T_n), (Z_n, \Gamma_n), W_n\}$ from the domain's interior and time/spatial boundaries according to their probability densities(ρ), i.e. generate (X_n, T_n) from D_T according to ρ_1, (Z_n, Γ_n) from $\partial\mathrm{D}_T$ according to ρ_2, and W_n from D according to ρ_3.

Step 3. Calculate the squared error $E(R_n; \theta_n)$ at the randomly sampled points R_n .

$$E(R_n, \theta_n) = \Big(\mathcal{L}[\widehat{\Psi}(X_n, T_n; \theta_n)]\Big)^2 + \Big(\widehat{\Psi}(Z_n, \Gamma_n; \theta_n) - f(Z_n, \Gamma_n)\Big)^2 + \Big(\widehat{\Psi}(W_n, 0; \theta_n) - \Psi_0(W_n)\Big)^2$$

Step 4. Take a descent step at the random point R_n

$$\theta_{n+1} = \theta_n - \alpha_n \nabla_\theta \; E(R_n; \theta_n)$$

Step 5. Repeat until the convergence criterion is satisfied.

$$\lim_{n\to\infty} \nabla_\theta E(R_n; \theta_n) = 0$$

Here, learning rate $\alpha_n \in (0, 1)$ decrease an $n \to \infty$ and $\nabla_\theta E(R_n; \theta_n)|_{\theta_n}$ is the unbiased estimation of $\nabla_\theta C(R_n; \theta_n)$ i.e., $\mathbb{E}\left[\nabla_\theta E(R_n; \theta_n)|_{\theta_n}\right] = \nabla_\theta C(R_n; \theta_n)$ In fact,

$$\begin{aligned}
&\mathbb{E}\left[\nabla_\theta E(R_n; \theta_n)|_{\theta_n}\right] \\
=&\mathbb{E}\left[\nabla_\theta \left(\Big(\mathcal{L}[\widehat{\Psi}(X_n, T_n; \theta_n)]\Big)^2 + \Big(\widehat{\Psi}(Z_n, \Gamma_n; \theta_n) - f(Z_n, \Gamma_n)\Big)^2 + \Big(\widehat{\Psi}(W_n, 0; \theta_n) - \Psi_0(W_n)\Big)^2\right)\right] \\
=&\nabla_\theta \Bigg[\int_{\mathrm{D}_T} \Big(\mathcal{L}[\widehat{\Psi}(X_n, T_n; \theta_n)]\Big)^2 \; \rho_1(X_n, T_n) \; d(X_n, T_n) + \int_{\partial\mathrm{D}_T} \Big(\widehat{\Psi}(Z_n, \Gamma_n; \theta_n) - f(Z_n, \Gamma_n)\Big)^2 \; \rho_2(Z_n, \Gamma_n) \; d(Z_n, \Gamma_n) \\
&+ \int_{\mathrm{D}} \Big(\widehat{\Psi}(W_n, 0; \theta_n) - \Psi_0(W_n)\Big)^2 \; \rho_3(W_n) \; d(W_n)\Bigg] \\
=&\nabla_\theta \Bigg[\int_{\mathrm{D}_T} \Big|\mathcal{L}[\widehat{\Psi}(X_n, T_n; \theta_n)]\Big|^2 \; \rho_1(X_n, T_n) \; d(X_n, T_n) + \int_{\partial\mathrm{D}_T} \Big|\widehat{\Psi}(Z_n, \Gamma_n; \theta_n) - f(Z_n, \Gamma_n)\Big|^2 \; \rho_2(Z_n, \Gamma_n) \; d(Z_n, \Gamma_n) \\
&+ \int_{\mathrm{D}} \Big|\widehat{\Psi}(W_n, 0; \theta_n) - \Psi_0(W_n)\Big|^2 \; \rho_3(W_n) \; d(W_n)\Bigg] \\
=&\nabla_\theta \left[\|\mathcal{L}[\widehat{\Psi}(X_n, T_n; \theta_n)]\|^2_{\mathrm{D}_T, \rho_1} + \Big\|\widehat{\Psi}(Z_n, \Gamma_n; \theta_n) - f(Z_n, \Gamma_n)\Big\|^2_{\partial\mathrm{D}_T, \rho_2} + \Big\|\widehat{\Psi}(W_n, 0; \theta_n) - \Psi_0(W_n)\Big\|^2_{\mathrm{D}, \rho_3}\right] \\
=&\nabla_\theta \, C(R_n; \theta_n) \, .
\end{aligned}$$

Thus, we can use $\nabla_\theta E(R_n;\theta_n)$ instead of $\nabla_\theta C(R_n;\theta_n)$. In order to describe the DGM more vividly, we represent following Figure 1. Further in the next Section 2.1, we present the implementation of DGM algorithm.

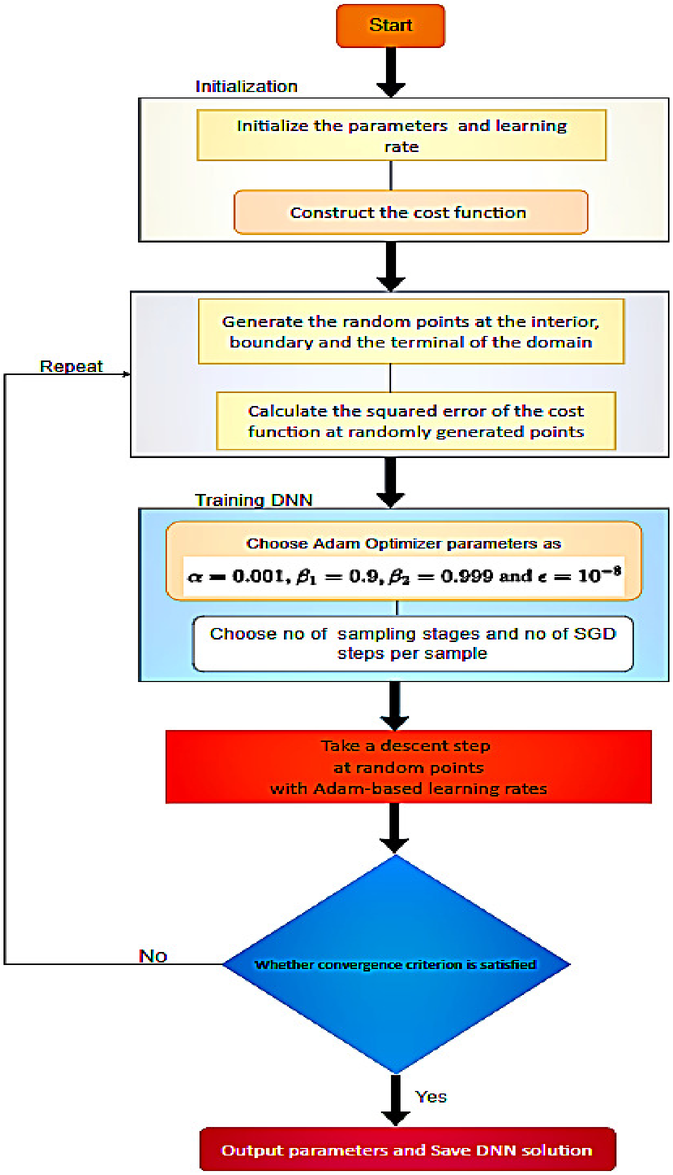
Start
Initialization
Initialize the parameters and learning rate
Construct the cost function
Repeat
Generate the random points at the interior, boundary and the terminal of the domain
Calculate the squared error of the cost function at randomly generated points
Training DNN
Choose Adam Optimizer parameters as
$\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ and $\epsilon = 10^{-8}$
Choose no of sampling stages and no of SGD steps per sample
Take a descent step at random points with Adam-based learning rates
No
Whether convergence criterion is satisfied
Yes
Output parameters and Save DNN solution

Figure 1: Entire process of DGM

2.1. Description of the algorithm's implementation

The implementation of the algorithm is described in this section, including the DGM's network design, hyper-parameters, and computing strategy. Sirignano and Spiliopoulos [38] used an architecture which is comparable to LSTM network [40]. While, here we use an architecture (see; Fig. 2) which is similar to GRU network [39] . It is made up of three layers: an input layer, a hidden layer, and an output layer, all of which are referred to as the DGM layer, however it may easily be expanded to include more hidden layers.

$$\begin{aligned}
H^1 &= \sigma\left(w^1 \vec{x} + b^1\right), \\
I^m &= \sigma\left(u^{i,m}\,\vec{x} + w^{i,m}\,H^m + b^{i,m}\right), \quad m = 1, \ldots, M \\
J^m &= \sigma\left(u^{j,m}\,\vec{x} + w^{j,m}\,H^m + b^{j,m}\right), \quad m = 1, \ldots, M \\
K^m &= \sigma\left(u^{k,m}\,\vec{x} + w^{k,m}\left(H^m \odot J^m\right) + b^{k,m}\right), \quad m = 1, \ldots, M, \\
H^{m+1} &= \left(1 - I^m\right) \odot K^m + I^m \odot H^m, \quad m = 1, \ldots, M, \\
\widehat{\Psi}(x, t; \theta) &= w H^{L+1} + b,
\end{aligned} \tag{10}$$

where $\vec{x} = (x, t)$, $M + 1$ represents the number of hidden layers in the network, and $\odot$ represents Hadamard (element-wise multiplication), and $\theta = \{w^1, b^1, (u^{i,m}, w^{i,m}, b^{i,m})_{m=1}^{M}, (u^{j,m}, w^{j,m}, b^{j,m})_{m=1}^{M}, \left(u^{k,m}, w^{k,m}, b^{k,m}\right)_{m=1}^{M}, w, b\}$ are the DNN's parameters, N represents number of units in each layer, σ represents element-wise nonlinearity i.e., $\sigma(k) = \left(\phi\left(k_1\right), \phi\left(k_2\right), \ldots, \phi\left(k_N\right)\right)$, ϕ represents non-linear activation function such as sigmoidal, relu, tanh and others. Dimensions of DNN's parameters (θ) are given as: $w^1 \in \mathbb{R}^{N\times(d+1)}, b^1 \in \mathbb{R}^N, u^{i,m} \in \mathbb{R}^{N\times(d+1)}, w^{i,m} \in \mathbb{R}^{N\times N}$, $b^{i,m} \in \mathbb{R}^N, u^{j,m} \in \mathbb{R}^{N\times(d+1)}, w^{j,m} \in \mathbb{R}^{N\times N}, b^{j,m} \in \mathbb{R}^N, u^{k,m} \in \mathbb{R}^{N\times(d+1)}, w^{k,m} \in \mathbb{R}^{N\times N}, b^{k,m} \in \mathbb{R}^N, w \in \mathbb{R}^{1\times N}$, and $b \in \mathbb{R}$.
For our work, we choose the hyperparameters, $M = 2$ (i.e., three hidden layers), $N = 50$ (i.e., nodes per layer), and $\phi(y) = tanh(y)/relu(y)$. Parameters of the DNN (Eq.(10)) were initialized by using the Xavier initialization [41], and Adam optimizer[42] is used for updating the parameters of DNN (Eq.(10)).

3. Convergence Analysis

The cost function $C(\widehat{\Psi})$ can measure how well $\widehat{\Psi}(x, t; \theta)$ satisfies the differential operator, boundary and initial condition. The approximation capabilities of neural network architectures have recently been investigated by many authors. Especially, in [43], they have ascertained that the standard multilayer feed-forward networks with activation function (ϕ) can approximate any continuous function define on arbitrary compact subset of D, whenever activation function (ϕ) is continuous, bounded

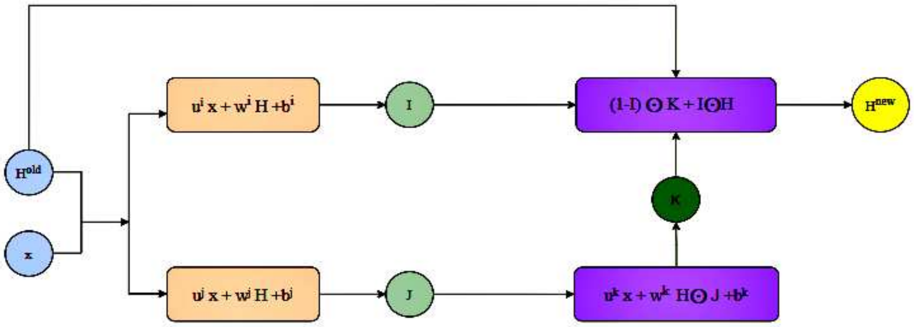

Figure 2: A single DGM layer's operations

and non-constant. Further, convergence analysis of cost function & neural network to the solution of gBHE is performed in a similar context [44, 45]. To show this, first we will prove that there exists $\widehat{\Psi^n} \in \mathfrak{C}^n$ such that $C\left(\widehat{\Psi^n}\right) \to 0$, as $n \to \infty$ i.e., convergence of the cost function $C(\widehat{\Psi})$, and then discuss the convergence of the neural network to the solution of gBHE i.e., $\widehat{\Psi^n} \to \Psi$, as $n \to \infty$. where, $\widehat{\Psi^n}$ is a neural network with n hidden units, and $\mathfrak{C}^n$ is a class of neural networks with n hidden units. From theorem 3 of [26], we can define $\mathfrak{C}^n$ as follows:

$$\mathfrak{C}^n(\phi) = \left\{ \Xi(x,t) : \mathrm{R}^{1+d} \mapsto \mathrm{R} : \Xi(x,t) = \sum_{i=1}^{n} \mathcal{B}_i \, \phi \left(\mathrm{w}_{1,i} \, t + \sum_{j=1}^{d} \mathrm{w}_{j,i} \, x_j + c_j \right) \right\} \tag{11}$$

where
- ϕ : the activation function of sigmoid type
- c_j : the threshold of the j_{th} hidden unit;
- $\mathrm{w}_{j,i}$: the connection weight of the i_{th} hidden unit and the j_{th} input unit;
- $\mathcal{B}_i$: the connection weight of the i_{th} hidden unit and the output unit.

where $\theta = (\mathrm{b}_1, \cdots, \mathrm{b}_n, \mathrm{w}_{1,1}, \cdots, \mathrm{w}_{d,n}, c_1, \cdots, c_n) \in \mathrm{R}^{2n+n(1+d)}$ are the parameter space's elements.

3.1. Convergence of the cost function $C(\widehat{\Psi})$

In this subsection, a theorem is presented to show the existence of neural networks $\widehat{\Psi}$, that makes the cost function $C(\widehat{\Psi})$ arbitrary small, based on the results of [28] and few assumptions. First, we define the assumptions $\boldsymbol{A_1} - \boldsymbol{A_3}$:

$\boldsymbol{A_1}$. Let $\mathrm{D} \subset \mathrm{R}^d$ be a bounded set and $\partial\mathrm{D}$ is its smooth boundary, and we set $\mathrm{D}_T = \mathrm{D} \times (0,T]$ and $\partial\mathrm{D}_T = \partial\mathrm{D} \times (0,T]$, and let parameters ρ_1, ρ_2 and ρ_3 be the measures of the D_T, D and $\partial\mathrm{D}$, respectively.

$\boldsymbol{A_2}$. Let $\mathfrak{C}^n(\phi)$ given by Eq. (11) is bounded and non-constant, where $\phi \in \mathcal{C}^2(\mathrm{D})$ is a common function of hidden units. Moreover $\mathfrak{C}(\phi) = \{\mathfrak{C}^n(\phi)\}_{n=1}^{\infty}$ is the set implemented by $\mathfrak{C}^n(\phi)$, where n is an arbitrarily large number of multiple hidden layer units.

$\boldsymbol{A_3}$ Assume that the following non-linear terms $\widehat{\Psi}_{xx}$ and $p\Psi^s\Psi_x + q\Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)$ be locally Lipschitz in (Ψ, Ψ_x) with Lipschitz constant which has at most polynomial growth on Ψ and Ψ_x uniformly with respect to x, t. i.e.,

$$|\widehat{\Psi}_{xx} - \Psi_x| \leq \left(|\widehat{\Psi}_x|^{b_1/2} + |\Psi_x|^{b_2/2}\right)|\widehat{\Psi}_x - \Psi_x| \tag{12}$$

$$\left|p\widehat{\Psi}^s\widehat{\Psi}_x - p\Psi^s\Psi_x\right| \leq \left(\left|\widehat{\Psi}^s\right|^{b_1/2} + \left|\widehat{\Psi}_x\right|^{b_2/2} + |\Psi^s|^{b_3/2} + |\Psi_x|^{b_4/2}\right)\left(\left|\widehat{\Psi}^s - \Psi^s\right| + \left|\widehat{\Psi}_x - \Psi_x\right|\right) \tag{13}$$

$$\left|\widehat{\Psi}\left(\widehat{\Psi}^s - 1\right)\left(\widehat{\Psi}^s - r\right) - \Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)\right| \leq \left(\left|\widehat{\Psi}\right|^{b_1/2} + \left|\widehat{\Psi}^s\right|^{b_2/2} + |\Psi|^{b_3/2} + |\Psi^s|^{b_4/2}\right)\left(|\widehat{\Psi} - \Psi| + \left|\widehat{\Psi}^s - \Psi^s\right|\right) \tag{14}$$

where $0 \leq b_1, b_2, b_3, b_4 < \infty$ are some constants.

Theorem 1. *Under the assumptions of* $(\boldsymbol{A_1}) - (\boldsymbol{A_3})$, *there exists a positive constant* $\mathcal{J} > 0$ *and a function* $\widehat{\Psi} \in \mathfrak{C}(\phi)$ *such that*

$$C(\widehat{\Psi}) \leq \mathcal{J}\,\epsilon$$

for $\forall \epsilon > 0$, *where* $\mathcal{J}$ *may depend on* ρ_1, ρ_2, ρ_3 *and* D_T.

Proof: From Theorem 3 of [43] we can conclude that there exists a function $\widehat{\Psi} \in \mathfrak{C}(\phi)$ which is uniformly 2-dense on compact subset of $\mathcal{C}^2\left(\mathrm{R}^{1+d}\right)$. This means that for $\widehat{\Psi} \in \mathcal{C}^{1,2}\left(\mathrm{R}^d \times [0, T]\right)$ and $\epsilon > 0$, there is $\widehat{\Psi} \in \mathfrak{C}(\phi)$ such that

$$\sup_{(x,t)\in\mathrm{D}_T}\left|\partial_t\Psi(x,t) - \partial_t\widehat{\Psi}(x,t;\theta)\right| + \max_{|a|\leq 2}\sup_{(x,t)\in\bar{\mathrm{D}}_T}\left|\partial_x^{(a)}\Psi(x,t) - \partial_x^{(a)}\widehat{\Psi}(x,t;\theta)\right| < \epsilon \tag{15}$$

Under assumption $\boldsymbol{A_3}$ and applying the Hölder inequality with indices e_1 and e_2 satisfying $\frac{1}{e_1} + \frac{1}{e_2} = 1$, we have first non-linear term as:

$$
\begin{aligned}
&\int_{\mathrm{D}_T} |\widehat{\Psi}_{xx} - \Psi_{xx}|^2 d\rho_1(x,t)\\
&\qquad \leq \int_{\mathrm{D}_T} \left(|\widehat{\Psi}_x|^{b_1} + |\Psi_x|^{b_2}\right)\left(|\widehat{\Psi}_x - \Psi_x|^2\right) d\rho_1(x,t)\\
&\qquad \leq \left[\int_{\mathrm{D}_T} \left(|\widehat{\Psi}_x|^{b_1} + |\Psi_x|^{b_2}\right)^{e_1} d\rho_1(x,t)\right]^{1/e_1} \left(\int_{\mathrm{D}_T} \left(|\widehat{\Psi}_x - \Psi_x|\right)^{2e_2} d\rho_1(x,t)v\right)^{1/e_2}\\
&\qquad \leq \left[\int_{\mathrm{D}_T} \left(|(\widehat{\Psi}_x - \Psi_x)|^{b_1} + |\Psi_x|^{b_1 \vee b_2}\right)^{e_1} d\rho_1(x,t)\right]^{1/e_1} \left(\int_{\mathrm{D}} \left(|\widehat{\Psi}_x - \Psi_x|\right)^{2e_2} d\rho_1(x,t)\right)^{1/e_2}\\
&\qquad \leq \mathcal{J}\left(\epsilon^{b_1} + \sup|\Psi_x|^{b_1 \vee b_2}\right)\epsilon^2
\end{aligned}
\tag{16}
$$

Similarly, second non-linear term as:

$$
\begin{aligned}
&\int_{\mathrm{D}_T} \left|p\widehat{\Psi}^s\widehat{\Psi}_x - p\Psi^s\Psi_x\right|^2 d\rho_1(x,t)\\
&\leq \int_{\mathrm{D}_T} \left(\left|\widehat{\Psi}^s\right|^{b_1} + \left|\widehat{\Psi}_x\right|^{b_2} + |\Psi^s|^{b_3} + |\Psi_x|^{b_4}\right) \times \left(\left|\widehat{\Psi}^s - \Psi^s\right|^2 + \left|\widehat{\Psi}_x - \Psi_x\right|^2\right) d\rho_1(x,t)\\
&\leq \left(\int_{\mathrm{D}_T} \left(\left|\widehat{\Psi}^s\right|^{b_1} + \left|\widehat{\Psi}_x\right|^{b_2} + |\Psi^s|^{b_3} + |\Psi_x|^{b_4}\right)^{e_1} d\rho_1(x,t)\right)^{1/e_1}\\
&\qquad\qquad\qquad\qquad \times \left(\int_{\mathrm{D}_T} \left(\left|\widehat{\Psi}^s - \Psi^s\right|^2 + \left|\widehat{\Psi}_x - \Psi_x\right|^2\right)^{e_2} d\rho_1\left(x,t\right)\right)^{1/e_2}\\
&\leq \mathcal{J}\left(\int_{\mathrm{D}} \left(\left|\widehat{\Psi}^s - \Psi^s\right|^{b_1} + \left|\widehat{\Psi}_x - \Psi_x\right|^{b_2} + |\Psi^s|^{b_1 \vee s_3} + |\Psi_x|^{b_2 \vee b_4}\right)^{e_1} d\rho_1(x,t)\right)^{1/e_1}\\
&\qquad\qquad\qquad\qquad \times \left(\int_{\mathrm{D}_T} \left(\left|\widehat{\Psi}^s - \Psi^s\right|^2 + \left|\widehat{\Psi}_x - \Psi_x\right|^2\right)^{e_2} d\rho_1\left(x,t\right)\right)^{1/e_2}\\
&\leq \mathcal{J}\left(\epsilon^{b_1} + \epsilon^{b_2} + \sup_{\mathrm{D}_T}|\Psi^s|^{b_1 \vee b_3} + \sup_{\mathrm{D}_T}|\Psi_x|^{b_2 \vee b_4}\right)\epsilon^2
\end{aligned}
\tag{17}
$$

and third non-linear term is:

$$
\begin{aligned}
&\int_{\mathrm{D}_T} \left|\widehat{\Psi}\left(\widehat{\Psi}^s - 1\right)\left(\widehat{\Psi}^s - r\right) - \Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)\right|^2 d\rho_1(x,t) \\
&\leq \int_{\mathrm{D}_T} \left(|\widehat{\Psi}|^{b_1} + \left|\widehat{\Psi}^s\right|^{b_2} + |\Psi|^{b_3} + |\Psi^s|^{b_4}\right) \times \left(|\widehat{\Psi} - \Psi|^2 + \left|\widehat{\Psi}^s - \Psi^s\right|^2\right) d\rho_1(x,t) \\
&\leq \left(\int_{\mathrm{D}_T} \left(|\widehat{\Psi}|^{b_1} + \left|\widehat{\Psi}^s\right|^{b_2} + |\Psi|^{b_3} + |\Psi^s|^{b_4}\right)^{e_1} d\rho_1(x,t)\right)^{1/e_1} \times \left(\int_{\mathrm{D}_T} \left(|\widehat{\Psi} - \Psi|^2 + \left|\widehat{\Psi}^s - \Psi^s\right|^2\right)^{e_2} d\rho_1(x,t)\right)^{1/e_2} \\
&\leq \mathcal{J}\left(\int_{\mathrm{D}_T} \left(|\widehat{\Psi} - \Psi|^{b_1} + \left|\widehat{\Psi}^s - \Psi^s\right|^{b_2} + |\Psi|^{b_1 \vee b_3} + |\Psi^s|^{b_2 \vee b_4}\right)^{e_1} d\rho_1(x,t)\right)^{1/e_1} \\
&\qquad\qquad\qquad\qquad \times \left(\int_{\mathrm{D}_T} \left(|\widehat{\Psi} - \Psi|^2 + \left|\widehat{\Psi}^s - \Psi^s\right|^2\right)^{e_2} d\rho_1(x,t)\right)^{1/e_2} \\
&\leq \mathcal{J}\left(\varepsilon^{b_1} + \varepsilon^{b_2} + \sup_{\mathrm{D}_T} |\Psi|^{b_1 \vee b_3} + \sup_{\mathrm{D}_T} |\Psi^s|^{b_2 \vee b_4}\right)\varepsilon^2
\end{aligned}
\tag{18}
$$

where $b_1 \vee b_3 = \max\{b_1, b_3\}$, and $\mathcal{J}$ may vary from line to line.

From Equations (15, 16, 17 and 18) we can conclude that the cost function $C(\widehat{\Psi}) \leq P\epsilon^2$. i.e.,

$$
\begin{aligned}
C(\widehat{\Psi}) &= \|\mathcal{L}[\widehat{\Psi}(x,t;\theta)]\|^2_{\mathrm{D}_T,\rho_1} + \left\|\widehat{\Psi}(x,t;\theta) - f(x,t)\right\|^2_{\partial\mathrm{D}_T,\rho_2} + \left\|\widehat{\Psi}(x,0;\theta) - \Psi_0(x)\right\|^2_{\mathrm{D}_T,\rho_3} \\
&= \|\mathcal{L}[\widehat{\Psi}(x,t;\theta)] - \mathcal{L}[\Psi(x,t)]\|^2_{\mathrm{D}_T,\rho_1} + \left\|\widehat{\Psi}(x,t,\theta) - f(x,t)\right\|^2_{\partial\mathrm{D}_T,\rho_2} + \left\|\widehat{\Psi}(x,0;\theta) - \Psi_0(x)\right\|^2_{\mathrm{D}_T,\rho_3} \\
&= \int_{\mathrm{D}_T} \left|\widehat{\Psi}_t - \Psi_t\right|^2 d\rho_1(x,t) + \int_{\mathrm{D}_T} \left|\widehat{\Psi}_{xx} - \Psi_{xx}\right|^2 d\rho_1(x,t) + \int_{\mathrm{D}_T} \left|p\widehat{\Psi}^s\widehat{\Psi}_x - p\Psi^s u_x\right|^2 d\rho_1(x,t) \\
&+ \int_{\mathrm{D}_T} \left|\widehat{\Psi}\left(\widehat{\Psi}^s - 1\right)\left(\widehat{\Psi} - r\right) - \Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)\right|^2 d\rho_1(x,t) + \int_{\partial\mathrm{D}_T} \left|\widehat{\Psi}(0,t,\theta) - f_1(x,t)\right|^2 d\rho_2(x,t) \\
&+ \int_{\partial\mathrm{D}_T} \left|\widehat{\Psi}(1,t,\theta) - f_2(x,t)\right|^2 d\rho_2(x,t) + \int_{\mathrm{D}} \left|\widehat{\Psi}(0,x,\theta) - \Psi_0(x)\right|^2 d\rho_3(x,t) \\
&\leq \mathcal{J}\epsilon^2
\end{aligned}
\tag{19}
$$

for an appropriate constant $\mathcal{J} < \infty$. After re-scaling ϵ, the final step completes Theorem 1's proof.

3.2. Convergence of the neural network to the gBHE solution

The convergence of the cost function $C(\widehat{\Psi})$ was examined in the previous subsection. We don't go into detail about non-homogeneous problems in this part because non-homogeneous problems can be handled using the homogeneous method (e.g.: Section 4 of Chapter V in [46] or Chapter 8 of [47]). In this subsection we will show how the neural network $\widehat{\Psi}$ converges to the exact solution $\Psi(x,t)$ as $n \to \infty$ of the gBHE with homogeneous boundary condition.

$$\begin{cases} \mathcal{L}[\Psi(x,t)] = 0, \quad (x,t) \in \mathrm{D}_T \\ \Psi(x,0) = \Psi_0(x), \quad x \in \mathrm{D} \\ \Psi(0,t) = \Psi(1,t) = 0 \quad \text{on} \quad \partial\mathrm{D}_T \end{cases} \tag{20}$$

Now our cost function become as Eq (21)

$$C(\widehat{\Psi}) = \left\| \mathcal{L}[\widehat{\Psi}(x,t;\theta)] \right\|^2_{\mathrm{D}_T,\rho_1} + \left\| \widehat{\Psi}(x,t;\theta) \right\|^2_{\partial\mathrm{D}_T,\rho_2} + \left\| \widehat{\Psi}(x,0,\theta) - \Psi_0(x) \right\|^2_{\mathrm{D},\rho_3} \tag{21}$$

Remember that the above norms are $L^2(X)$ norms in the relevant spaces $X = \mathrm{D}_T, \partial\mathrm{D}_T$ and D respectively. We can use the following result from Theorem 1:

$$C\left(\widehat{\Psi}^n\right) \to 0 \text{ as } n \to \infty$$

In addition, every neural network $\widehat{\Psi}^n$ with a source term $h^n(x,t)$, satisfies a PDE as follows:

$$\begin{cases} \mathcal{L}\left[\widehat{\Psi}^n(x,t)\right] = h^n(x,t), \text{ for } (x,t) \in \mathrm{D}_T \\ \widehat{\Psi}^n(x,0) = \Psi_0^n(x), \text{ for } x \in \mathrm{D} \\ \widehat{\Psi}^n(x,t) = f^n(x,t), \text{ for } (0,t) \in \partial\mathrm{D}_T \end{cases} \tag{22}$$

for some h^n, f^n, and Ψ_0^n such that

$$\|h^n\|^2_{2,\mathrm{D}_T} + \|f^n\|^2_{2,\partial\mathrm{D}_T} + \|\Psi_0^n - \Psi_0\|^2_{2,\mathrm{D}} \to 0 \text{ as } n \to \infty. \tag{23}$$

further, some assumptions are made for this subsection:

A

$\boldsymbol{A_4}$ - Let $\lambda(x,t)$ be a positive function for all $(x,t) \in \mathrm{D}_T$ such that

$$|p\Psi^s\Psi_x + q\Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)| \leq \lambda(x,t)\|\Psi_x\|,$$

with $\lambda \in L^{d+2+\eta}\left(\mathrm{D}_T\right)$ for some $\eta > 0$.
$\boldsymbol{A_5}$- Ψ_{xx} and $p\Psi^s\Psi_x + q\Psi\left(\Psi^s - 1\right)\left(\Psi^s - r\right)$ are Lipschitz continuous in $(x, t, \Psi, \Psi_x) \in \mathrm{D}_T \times \mathrm{R} \times \mathrm{R}^d$ uniformly on compacts of the form $\left\{(x,t) \in \bar{\mathrm{D}}_T, |\Psi| \leq K, |\Psi_x| \leq K\right\}$.
$\boldsymbol{A_6}$- $\Psi_0(x) \in \mathcal{C}^{0,2+\xi}(\bar{\mathrm{D}})$ for some $\xi > 0$ with itself and its first derivative bounded in $\bar{\mathrm{D}}$.
$\boldsymbol{A_7}$- D is a bounded, open subset of R^d with boundary $\partial\mathrm{D} \in \mathcal{C}^2$.
$\boldsymbol{A_8}$- For every $n \in \mathrm{N}, \widehat{\Psi}^n \in \mathcal{C}^{1,2}\left(\bar{\mathrm{D}}_T\right)$, and $\left(\widehat{\Psi}^n\right)_{n\in\mathrm{N}} \in L^2\left(\mathrm{D}_T\right)$.

Theorem 2. Assume that the Eq. (20) has a unique bounded solution in $\mathcal{C}^{0,\mu,\mu/2}\left(\bar{\mathrm{D}}_T\right) \cap L^2\left(0, T; W_0^{1,2}(\mathrm{D})\right) \cap W_0^{(1,2),2}\left(\mathrm{D}'_T\right)$ for any $\mu > 0$ and any subdomain D'_T of D_T, under the criteria $\boldsymbol{A_4}$ to $\boldsymbol{A_8}$ and (23). Furthermore, If the sequence $\left\{\widehat{\Psi}^n(x,t)\right\}_{n\in\mathbb{N}}$ is uniformly bounded and equicontinuous, then $\widehat{\Psi}^n$ converges strongly to Ψ in $L^2\left(\mathrm{D}_T\right)$, and $\widehat{\Psi}^n$ uniformly converges to Ψ in D_T.

Proof. By using Theorems 6.3-6.5 of [47] and Theorem 2.1 of [48], we can obtain the existence, regularity, uniqueness of Eq.(20), and boundedness can be established from Theorem 2.1 in [48] and chapter V.2 in [47] of Eq.(20). Consider equations (22) under the situation $f^n(x,t) = 0$ and the solution to this problem denoted by $\widehat{\Phi^n}(x,t)$. We need to state that $\widehat{\Phi^n}(x,t)$ is uniformly bounded and satisfies the following equations:

$$\begin{aligned} \mathcal{L}\left[\widehat{\Phi^n}(x,t)\right] &= h^n(x,t), \text{ for } (x,t) \in \mathrm{D}_T \\ \widehat{\Phi^n}(0,x) &= \Psi_0^n(x), \text{ for } x \in \mathrm{D} \\ \widehat{\Phi^n}(x,t) &= 0, \text{ for } (x,t) \in \partial\mathrm{D}_T \end{aligned} \tag{24}$$

From Lemma 4.1 of [49] & under the assumptions $\boldsymbol{A_4}$ to $\boldsymbol{A_8}$, we might assume that $\left\{\widehat{\Phi^n}\right\}_{n\in\mathrm{N}}$ is uniformly bounded in $L^\infty\left(0, T; L^2(\mathrm{D})\right) \cap L^2\left(0, T; W_0^{1,2}(\mathrm{D})\right)$, and there exist a weakly sub-sequence, $\left\{\widehat{\Phi^n}\right\}_{n\in\mathrm{N}}$ (To keep things simple, we'll keep using the same notation for the sub-sequence.), which converges to Ψ in $L^\infty\left(0, T; L^2(\mathrm{D})\right)$ and weakly in $L^2\left(0, T; W_0^{1,2}(\mathrm{D})\right)$ and to some w weakly in $L^2(\mathrm{D})$ for every fixed $t \in (0, T]$.
Due to compactness of the embedding $W_0^{1,2}(\mathrm{D}) \subset L^2(\mathrm{D})$ and from Corollary 4 of [48], $\left\{\widehat{\Phi^n}\right\}_{n\in\mathrm{N}}$ converges strongly to Ψ in $L^2(\mathrm{D_T})$. Now, up to sub-sequences, $\left\{\widehat{\Phi^n}\right\}_{n\in\mathrm{N}}$ converges almost everywhere to Ψ in D_T. Further, from theorem 3.3 of [4], we can conclude that $\nabla\widehat{\Phi^n} \to \nabla\Psi$ almost everywhere in $\mathrm{D_T}$. Finally, we conclude that $\left\{\widehat{\Phi^n}\right\}_{n\in\mathbb{N}}$ strongly converges to Ψ in $L^2(0, T; W_0^{1,2}(\mathrm{D}))$.

Using assumption $\boldsymbol{A_4}$ and applying the Hölder inequality with indices e_1 and e_2 satisfying $\frac{1}{e_1}+\frac{1}{e_2}=1$, and put $\beta=1+\frac{d}{d+4}\in(1,2)$, we have

$$\begin{aligned}\int_{\mathrm{D}_T}\left|p\,\widehat{\Phi^n}^s\widehat{\Phi^n_x}+q\,\widehat{\Phi^n}\left(\widehat{\Phi^n}^s-1\right)\left(\widehat{\Phi^n}^s-r\right)\right|^\beta dx\,dt &\le \int_{\mathrm{D}_T}|\lambda(x,t)|^\beta\left|\widehat{\Phi^n_x}(x,t)\right|^\beta dx\,dt\\ &\le \left(\int_{\mathrm{D}_T}|\lambda(x,t)|^{e_1\beta}dx\,dt\right)^{1/e_1}\left(\int_{\mathrm{D}_T}\left|\widehat{v^n_x}(x,t)\right|^{e_2\beta}dx\,dt\right)^{1/e_2}.\end{aligned}$$

Let $e_2=2/\beta>1$, this implies that $e_1=\frac{2}{2-\beta}$, thus we get $e_1\beta=d+2$.
Moreover, by using assumptions $\boldsymbol{A_4}$ for a measurable set $M\subset\mathrm{D_T}$ (with a constant K) we get

$$\int_M\left|p\widehat{\Phi^n}^s\widehat{\Phi^n_x}+q\widehat{\Phi^n}\left(\widehat{\Phi^n}^s-1\right)\left(\widehat{\Phi^n}^s-r\right)\right|^\beta dx\,dt\le K\left(\int_M|\lambda(x,t)|^{d+2}dx\,dt\right)^{(2-\beta)/2}\le K\,|M|^{\frac{n}{d+2+\eta}} \tag{25}$$

Now according to Vitali's theorem we have

$$p\widehat{\Phi^n}^s\widehat{\Phi^n_x}+q\widehat{\Phi^n}\left(\widehat{\Phi^n}^s-1\right)\left(\widehat{\Phi^n}^s-r\right)\to p\Psi^s\Psi_x+q\Psi\left(\Psi^s-1\right)\left(\Psi^s-r\right)\text{ strongly in }L^\beta\left(\mathrm{D}_T\right)$$

as $n\to\infty$. Due to compactness of the embedding $L^q(\mathrm{D})\subset L^2(\mathrm{D})$, we can conclude that:

$$p\widehat{\Phi^n}^s\widehat{\Phi^n_x}+q\widehat{\Phi^n}\left(\widehat{\Phi^n}^s-1\right)\left(\widehat{\Phi^n}^s-r\right)\to p\Psi^s\Psi_x+q\Psi\left(\Psi^s-1\right)\left(\Psi^s-r\right)\text{ strongly in }L^2\left(\mathrm{D}_T\right).$$

For all $t_1\in(0,T]$ the weak formulation of the PDE (24) with $f^n=0$ given as follows (notice that Ψ_0^n strongly converges to Ψ_0 in $L^2(\mathrm{D})$).

$$\begin{aligned}&\int_{\mathrm{D}_{t_1}}\left[-\widehat{\Phi^n}\partial_t\phi+\left\langle\nabla\widehat{\Phi^n},\nabla\phi\right\rangle+\left(p\,\widehat{\Phi^n}^s\widehat{\Phi^n_x}+q\widehat{\Phi^n}\left(\widehat{\Phi^n}^s-1\right)\left(\widehat{\Phi^n}^s-r\right)-h^n\right)\phi\right](x,t)dxdt\\ &\qquad+\int_{\mathrm{D}}\widehat{\Phi^n}\left(x,t_1\right)\phi\left(x,t_1\right)dx-\int_{\mathrm{D}}\Psi_0^n(x)\phi(x,0)dx=0\end{aligned}$$

for every $\phi\in C_0^\infty\left(\mathrm{D}_T\right)$.
We may deduce from the convergence results that the limit point Ψ satisfies for every $t_1\in(0,T]$ the following Eq. (26) represent the weak formulation of Eq. (20).

$$\begin{aligned}\int_{\mathrm{D}_{t_1}} & \left[-\Psi \partial_t \phi+\langle\nabla \Psi, \nabla \phi\rangle+\left(p \Psi^s \Psi_x+q \Psi\left(\Psi^s-1\right)\left(\Psi^s-r\right)\right) \phi\right](x, t) d x d t \\ & +\int_{\mathrm{D}} \Psi\left(t_1, x\right) \phi\left(0, t_1\right) d x-\int_{\mathrm{D}} \Psi_0(x) \phi(x, 0) d x=0\end{aligned} \tag{26}$$

It is still necessary to examine the convergence of $\widehat{\Psi}^n - \widehat{\Phi}^n$ to zero, where $\widehat{\Psi}^n$ is the neural network approximation satisfying (24) and $\widehat{\Phi}^n$ is the neural network approximation satisfying (24) with $f^n = 0$. Since we assumed that $\left\{\widehat{\Psi}^n\right\}_n$ is uniformly bounded in $L^2\left(\mathrm{D}_T\right)$, therefore up to a sub-sequence, $\widehat{\Psi}^n$ weakly converges in $L^2\left(\mathrm{D}_T\right)$. Furthermore in addition, $\widehat{\Psi}^n(x,t)$ in Eq. (24) converges strongly to zero in $L^2\left(\mathrm{D}_T\right)$. Furthermore, $\widehat{\Psi}^n$ defined by f^n on the boundary $\partial\mathrm{D}_T$, will converge to 0 at least along a sub-sequence on the boundary (see for example Lemma 2.1 in Chapter II of [46].) As may be shown from the proof of Theorems 6.3-6.4-6.5 in Chapter V.6 in [46], and by using smoothness and uniqueness, $\widehat{\Psi}^n$ will same as $\widehat{\Phi}^n(x,t)$ (i.e., the solution to the PDE (24)) almost everywhere when $f^n = 0$.

Set $F_n = \left|\widehat{\Psi}^n - \widehat{\Phi}^n\right|^2$. Thus, function sequence $\{F_n\}$ is uniformly bounded in the $L^2(\mathrm{D}_T)$ because of the boundeness of $\widehat{\Psi}^n$ and $\widehat{\Phi}^n$ above. $\{F_n\}$ is integrable on area $\bar{\mathrm{D}}_T$ naturally. Moreover, $\{F_n\}$ converges to 0 almost everywhere. Furthermore, using the definition of F_n together with the uniform boundedness as well as equicontinuity of $\widehat{\Psi}^n$ and $\widehat{\Phi}^n$, $\forall\, \epsilon_0 > 0,\ \exists\quad \mathrm{c} > 0,$ such that $|(x,t)-(y,t)| < \mathrm{c},$ and $\{(x,t),(y,t)\} \in \bar{\mathrm{D}}_T$ and $n \geq 1$, there is

$$\begin{aligned}
& \Big|F_n(x,t) - F_n(y,t)\Big| \\
= & \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Phi}^n(x,t)\right|^2 - \left|\widehat{\Psi}^n(y,t) - \widehat{\Phi}^n(y,t)\right|^2\,\right| \\
= & \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Phi}^n(x,t)\right| + \left|\widehat{\Psi}^n(y,t) - \widehat{\Phi}^n(y,t)\right|\,\right| \times \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Phi}^n(x,t)\right| - \left|\widehat{\Psi}^n(y,t) - \widehat{\Phi}^n(y,t)\right|\,\right| \\
\leq & \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Phi}^n(x,t)\right| + \left|\widehat{\Psi}^n(y,t) - \widehat{\Phi}^n(y,t)\right|\,\right| \times \left|\,\left|\Big(\widehat{\Psi}^n(x,t) - \widehat{\Psi}^n(y,t)\Big) + \Big(\widehat{\Phi}^n(y,t) - \widehat{\Phi}^n(x,t)\Big)\right|\,\right| \\
\leq & \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Phi}^n(x,t)\right| + \left|\widehat{\Psi}^n(y,t) - \widehat{\Phi}^n(y,t)\right|\,\right| \times \left|\,\left|\widehat{\Psi}^n(x,t) - \widehat{\Psi}^n(y,t)\right| + \left|\widehat{\Phi}^n(y,t) - \widehat{\Phi}^n(x,t)\right|\,\right| \\
\leq & \mathcal{J}\,\epsilon_0,
\end{aligned}$$

which implies that function sequence $\{F_n(x,t)\}$ is equicontinuous. Thus, using Vitali's theorem, we infer that

$$\lim_{n\to\infty} \left\|\widehat{\Psi}^n - \widehat{\Phi}^n\right\|^2_{2,\mathrm{D}_T} = 0,$$

which shows that $\widehat{\Psi}^n - \widehat{\Phi}^n$ converges to 0 strongly in $L^2(\mathrm{D}_T)$. Using a triangle inequality, we can gain the following inequality

$$\begin{aligned}
&\lim_{n\to\infty}\left\|\widehat{\Psi}^n-\Psi\right\|_{2,\mathrm{D}_T}\\
=&\lim_{n\to\infty}\left\|\widehat{\Psi}^n-\widehat{\Phi}^n+\widehat{\Phi}^n-\Psi\right\|_{2,\mathrm{D}_T}\\
\leq&\lim_{n\to\infty}\left\|\widehat{\Psi}^n-\widehat{\Phi}^n\right\|_{2,\mathrm{D}_T}+\lim_{n\to\infty}\left\|\widehat{\Phi}^n-\Psi\right\|_{2,\mathrm{D}_T}\\
=&0.
\end{aligned}$$

This result shows that $\widehat{\Psi}^n$ converges strongly to Ψ in $L^2(\mathrm{D}_T)$. Furthermore, based on the equicontinuity and uniform boundedness of $\widehat{\Psi}^n$, by Arzelà-Ascoli theorem, we obtain that $\widehat{\Psi}^n$ uniformly converges to Ψ in D_T.

4. Numerical results and Discussion

In the following section, we assess and test the suggested method's performance on the gBHE and gHE, as well as demonstrate its efficacy. According to the DGM method, we have to generate a sample of random points from the domain of the given equation which includes interior point, boundary points, and terminal/initial points. For our work, we generate one thousand random points from the interior of the domain, and from the boundary and the terminal of the domain we generate one hundred random points for each. For training purposes, we set training parameters as given in Table 1. Parameters of the DNN in Eq. (10) were initialized by using the xavier initialization [41], and Adam optimizer [42] is used for updating the parameters of DNN in Eq. (10). We run the algorithm in a loop of fifty times and recorded the standard deviation(STD) between the fitted values for each time of the loop. Four examples based on various parameter values are chosen to test the efficiency of the method. For each example average CPU time is recorded as 60 seconds for a single loop and for a loop of 50 times the average CPU time is recorded as 16 minutes. For simulations, Google Colab has been used in this work. The obtained numerical solution of gBHE and gHE for different parameter combinations at some selected node points is compared with existing numerical methods in literature and discussed in Example 1-4.

Example 1. The obtained results of the gBHE Eq. (1) using the DGM method are compared by ADM [6, 12] and VIM [16] method at real coefficients r=0.001 and $s = 1$, advection coefficient $p = 1$, reaction coefficient $q = 1$, at times $t = 0.05, t = 0.1$ and $t = 1$ and presented in Table 2. The absolute errors at some selected node points are compared with the errors in ADM and VIM methods and it can be observed that the proposed method provides comparable solution to the exist-

Table 1: Training Parameters

Parameters	Numbers
Sampling stages	50
No. of SGD steps	10
No. of hidden layers	3
No. of nodes per layers	50
Activation functions	sigmoid/relu

ing ADM and VIM methods. The graph of absolute error in the DGM solution is presented by Fig. 3.

Table 2: Absolute errors when $r = 0.001, p = q = s = 1$

x	t	STD	Proposed	ADM[6]	ADM [12]	VIM [16]
0.1	0.05	0.000848	1.912E-07	1.937E-07	1.874E − 08	1.874E − 08
	0.1	0.000738	3.560E-07	3.874E-07	3.748E − 08	3.748E − 08
	1	0.000680	2.581E-07	3.875E-06	3.748E − 07	3.748E − 07
0.5	0.05	0.000980	2.295E-07	1.937E-07	1.874E − 08	1.874E − 08
	0.1	0.000905	2.901E-07	3.875E-07	3.748E − 08	1.375E − 08
	1	0.000807	2.466E-08	3.875E-06	3.748E − 07	3.748E − 07
0.9	0.05	0.001073	1.276E-07	1.937E-07	1.874E − 08	1.874E − 08
	0.1	0.001036	3.357E-07	3.875E-07	3.748E − 08	3.748E − 08
	1	0.000862	2.611E-07	3.876E-06	3.748E − 07	3.748E − 07

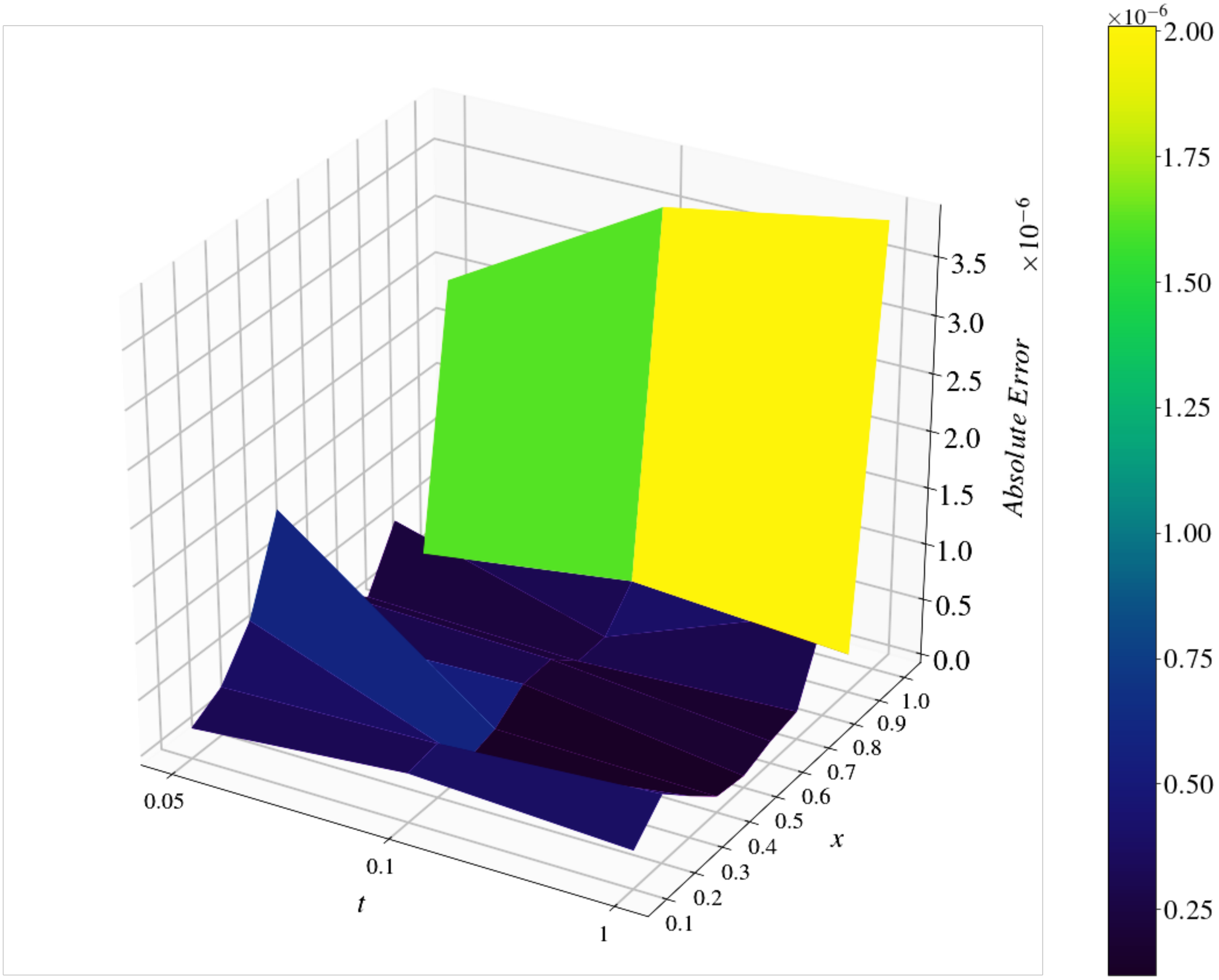

Figure 3: Absolute error for $p = q = 1,\ s = 1$ & $r = 0.001$

Example 2. The obtained results of the gBHE Eq. (1) using proposed method are compared by ADM [12] and VIM [16] methods at real coefficients $r = 0.01$ and $s = 2, 4$, advection coefficient $(p) = 1$, reaction coefficient $(q) = 1$, at times $t = 0.1, t = 0.2, t = 0.3, t = 0.4$ and $t = 0.5$ and reported in Table 3. The absolute errors at some selected node points are compared with the errors in ADM and VIM methods and it can be observed that the proposed method provides better solution than ADM and VIM methods. The graph of absolute error in the DGM solution is presented by Fig. 4 & Fig. 5

Table 3: Absolute errors when $r = 0.01, p = q = 1, s = 2, 4$

		$s = 2$				$s = 4$			
t	x	STD	Proposed	ADM [12]	VIM [16]	STD	Proposed	ADM [12]	VIM [16]
0.1	0.1	0.000592	1.0E-05	5.516E − 05	5.516E − 05	0.000624	2.2E-05	2.178E − 04	2.177E − 04
	0.2	0.000459	2.7E-06	1.103E − 04	1.103E − 04	0.059261	2.5E-05	4.357E − 04	4.353E − 04
	0.3	0.000508	1.6E-05	1.655E − 04	1.655E − 04	0.000430	2.0E-06	6.537E − 04	6.528E − 04
	0.4	0.001494	6.0E-06	2.207E − 04	2.206E − 04	0.000511	2.0E-06	8.718E − 04	8.703E − 04
	0.5	0.000773	5.0E-06	2.760E − 04	2.757E − 04	0.001312	2.4E-05	1.090E − 03	1.088E − 03
0.3	0.1	0.000589	7.0E-06	5.514E − 05	5.513E − 05	0.000754	2.6E-05	2.176E − 04	2.175E − 04
	0.2	0.000464	3.1E-05	1.103E − 04	1.103E − 04	0.051273	4.5E-05	4.352E − 04	4.348E − 04
	0.3	0.000547	4.2E-05	1.655E − 04	1.654E − 04	0.000481	2.1E-05	6.530E − 04	6.521E − 04
	0.4	0.001611	1.6E-05	2.206E − 04	2.205E − 04	0.000592	1.9E-05	8.709E − 04	8.693E − 04
	0.5	0.000737	2.9E-05	2.758E − 04	2.756E − 04	0.001376	1.5E-04	1.089E − 03	1.086E − 03
0.5	0.1	0.000617	9.0E-06	5.511E − 05	5.511E − 05	0.000888	6.6E-05	2.173E − 04	2.172E − 04
	0.2	0.000494	1.6E-05	1.102E − 04	1.102E − 04	0.037515	2.5E-04	4.348E − 04	4.344E − 04
	0.3	0.000667	3.0E-06	1.654E − 04	1.653E − 04	0.000571	7.4E-05	6.523E − 04	6.514E − 04
	0.4	0.001718	3.0E-06	2.205E − 04	2.204E − 04	0.000669	3.8E-05	8.699E − 04	8.684E − 04
	0.5	0.000782	5.9E-05	2.757E − 04	2.755E − 04	0.001464	6.7E-05	1.088E − 03	1.085E − 03

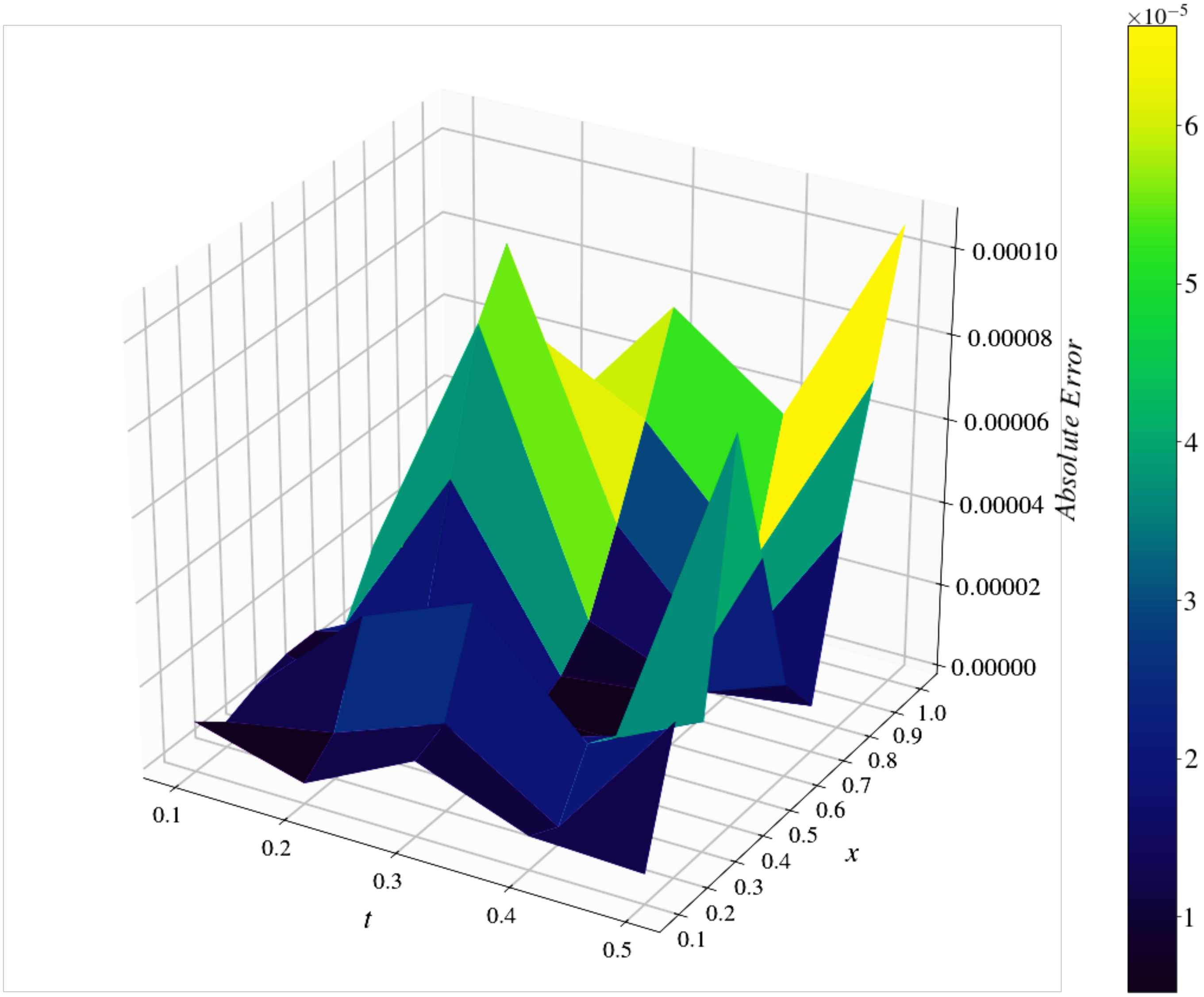

Figure 4: Absolute error for $p = q = 1,\ s = 2$ & $r = 0.01$

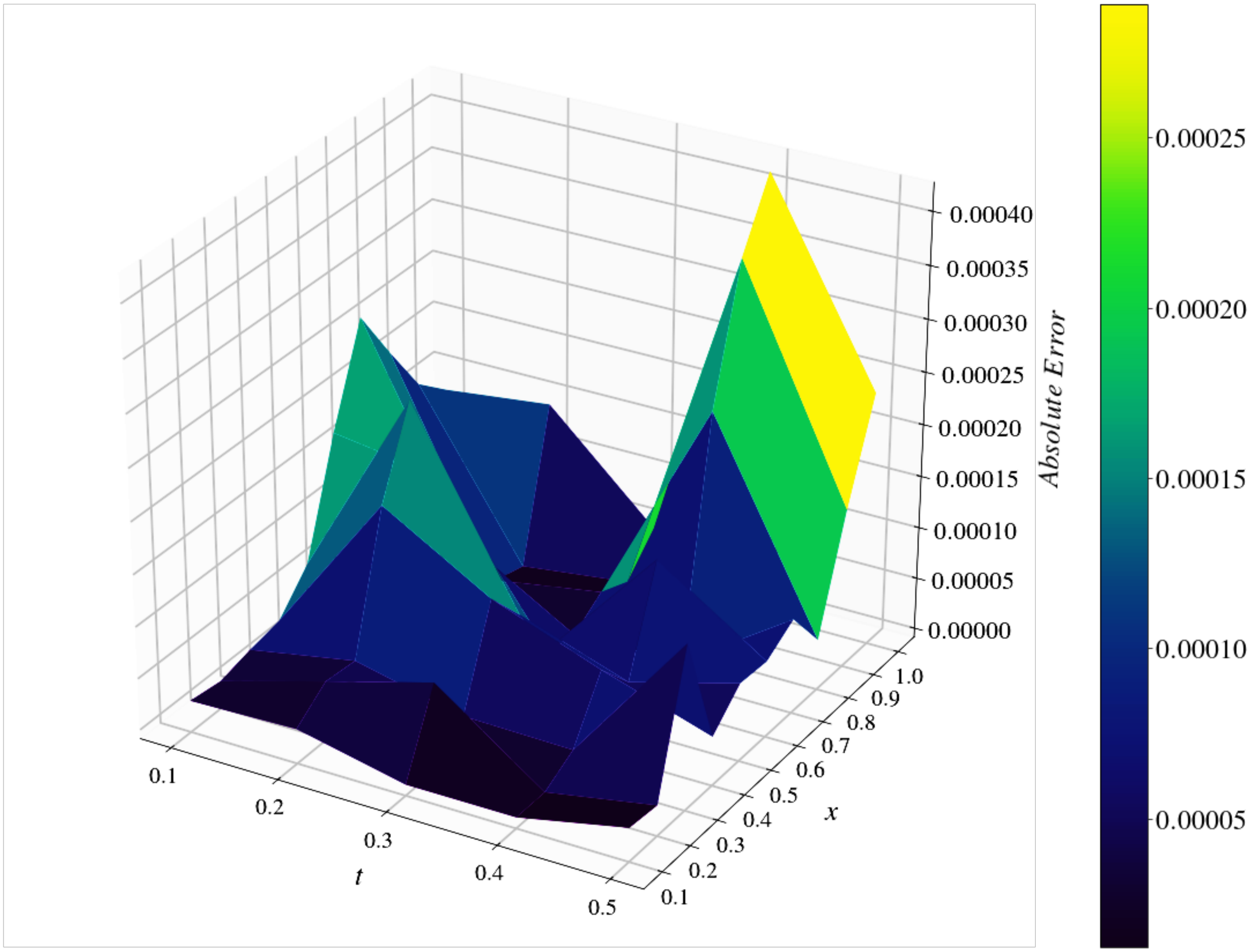

Figure 5: Absolute error for $p = q = 1,\ s = 4$ & $r = 0.01$

Example 3. The obtained results of the gHE Eq. (5) using proposed method are compared by ADM[6], FDS4[20] and OHAM [26] method at real coefficients $r = 0.001$ and $s = 1$, advection coefficient $(p) = 0$, reaction coefficient $(q) = 1$, at times $t = 0.05, t = 0.1$ and $t = 1$ and reported in Table 4. The absolute errors at some selected node points are compared with the errors in ADM, FDS4 and OHAM methods and it can be observed from Table 4 that the proposed method provides comparable solution to the other reported methods.The graph of absolute error in the DGM solution is presented by Fig. 6.

Table 4: Absolute errors when $r = 0.001, p = 0, q = s = 1$

x	t	STD	Proposed	ADM[6]	FDS4[20]	OHAM [26]
0.1	0.05	0.000627	1.179E-07	1.875E-07	2.499E − 08	2.499E − 08
	0.1	0.000735	2.875E-07	3.749E-07	4.998E − 08	4.998E − 08
	1	0.001205	3.046E-07	3.750E-06	4.998E − 07	4.998E − 07
0.5	0.05	0.000771	1.185E-07	1.875E-07	2.499E − 08	2.499E − 08
	0.1	0.000795	3.041E-08	3.750E-07	4.998E − 08	4.998E − 08
	1	0.001238	1.023E-07	3.750E-06	4.998E − 07	4.998E − 07
0.9	0.05	0.000892	1.463E-07	1.875E-07	2.499E − 08	2.499E − 08
	0.1	0.000791	4.055E-07	3.750E-07	4.998E − 08	4.998E − 08
	1	0.001328	3.167E-07	3.751E-06	4.998E − 07	4.998E − 07

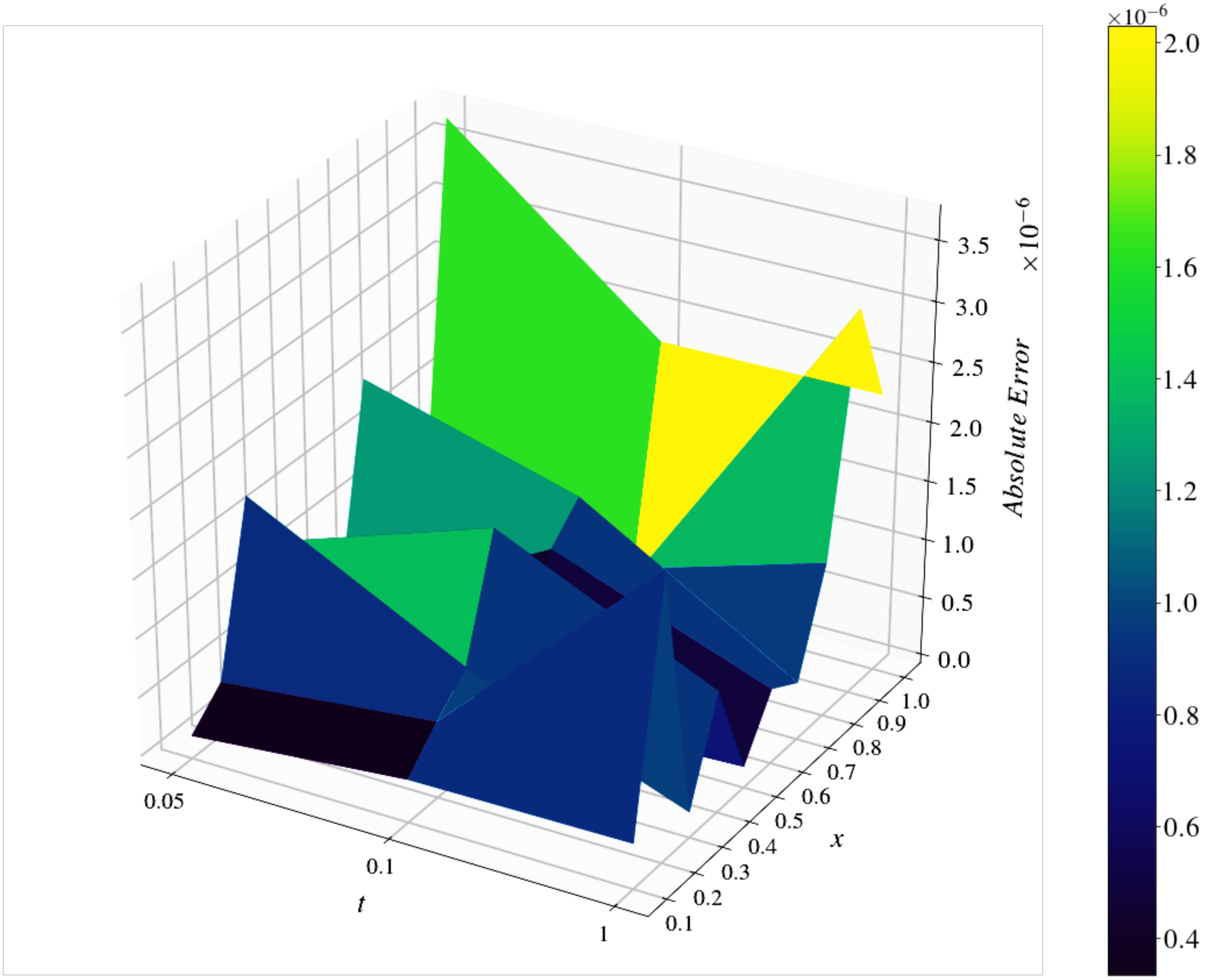

Figure 6: Absolute error for $p = 0,\ q = 1,\ s = 1$ & $r = 0.001$

Example 4. The obtained results of the gHE Eq. (5) using proposed method are compared by ADM[6], FDS4[20] and MCB-DQM [50] method at real coefficients $r = 0.001$ and $s = 2, 3$, advection coefficient $(p) = 0$, reaction coefficient $(q) = 1$, at times $t = 0.05, t = 0.1$ and $t = 1$ and reported in Table 5. The absolute errors at some selected node points are compared with the errors in ADM, FDS4 and MCB-DQM methods and it can be observed from Table 5 that the proposed method provides better solution than FDS4 and comparable solution to ADM and MCB-DQM. The graph of absolute error in the DGM solution is presented by Fig. 7 and Fig. 8.

Table 5: Absolute errors when $r = 0.001$, $p = 0$, $q = 1$, $s = 2, 3$

		$s = 2$					$s = 3$				
t	x	STD	Proposed	ADM[6]	FDS4[20]	MCB-DQM [50]	STD	Proposed	ADM[6]	FDS4[20]	MCB-DQM [50]
0.1	0.05	0.000560	5.420E-08	5.589E-07	1.118E − 06	4.492E − 07	0.000430	2.882E-07	1.984E-06	3.967E − 06	1.595E − 06
	0.1	0.000228	1.511E-07	1.118E-06	2.235E − 06	6.615E − 07	0.000770	1.855E-07	3.968E-06	7.935E − 06	2.348E − 06
	1	0.001205	3.046E-07	3.750E-06	4.998E − 07	4.998E − 07	0.000547	6.081E-07	3.966E-05	7.935E − 05	3.522E − 06
0.5	0.05	0.000754	2.345E-07	5.588E-07	1.118E − 06	1.031E − 06	0.000537	2.664E-07	1.984E-06	3.967E − 06	3.658E − 06
	0.1	0.000270	3.485E-07	1.118E-06	2.235E − 06	1.711E − 06	0.000955	2.236E-06	3.967E-06	7.933E − 06	6.072E − 06
	1	0.002580	3.485E-06	1.007E-05	2.235E − 05	2.779E − 06	0.000604	3.925E-07	3.966E-05	7.933E − 05	9.861E − 06
0.9	0.05	0.000927	2.216E-07	5.588E-07	1.117E − 06	4.492E − 07	0.000636	4.401E-07	1.983E-06	3.966E − 06	1.594E − 06
	0.1	0.000325	1.201E-07	1.118E-06	2.237E − 06	6.614E − 06	0.001092	7.322E-07	3.967E-06	7.931E − 06	2.348E − 06
	1	0.007759	3.275E-07	1.117E-06	2.234E − 05	9.926E − 07	0.000621	7.693E-06	3.965E-05	7.931E − 05	3.522E − 06

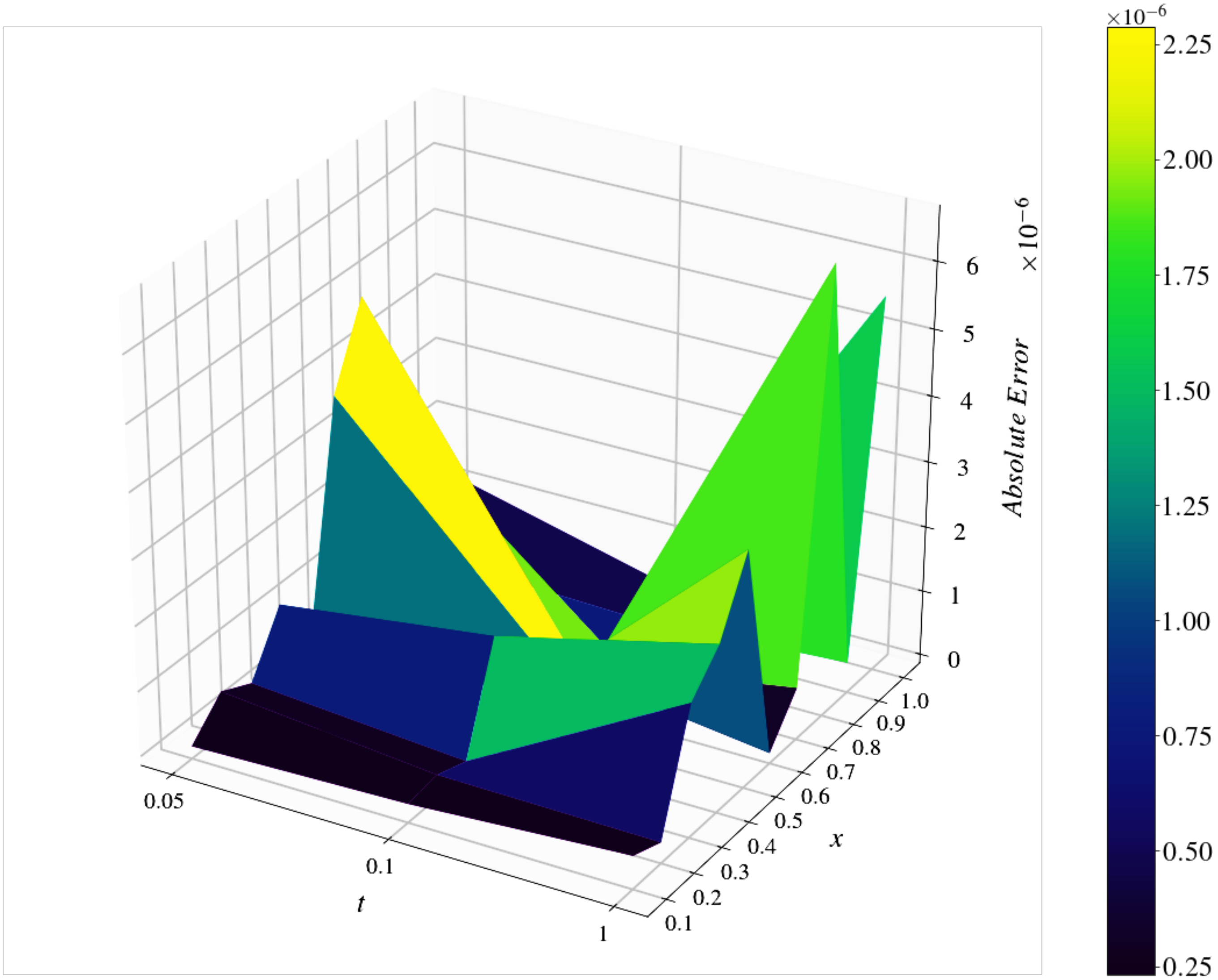

Figure 7: Absolute error for $p = 0$, $q = 1$, $s = 2$ & $r = 0.001$

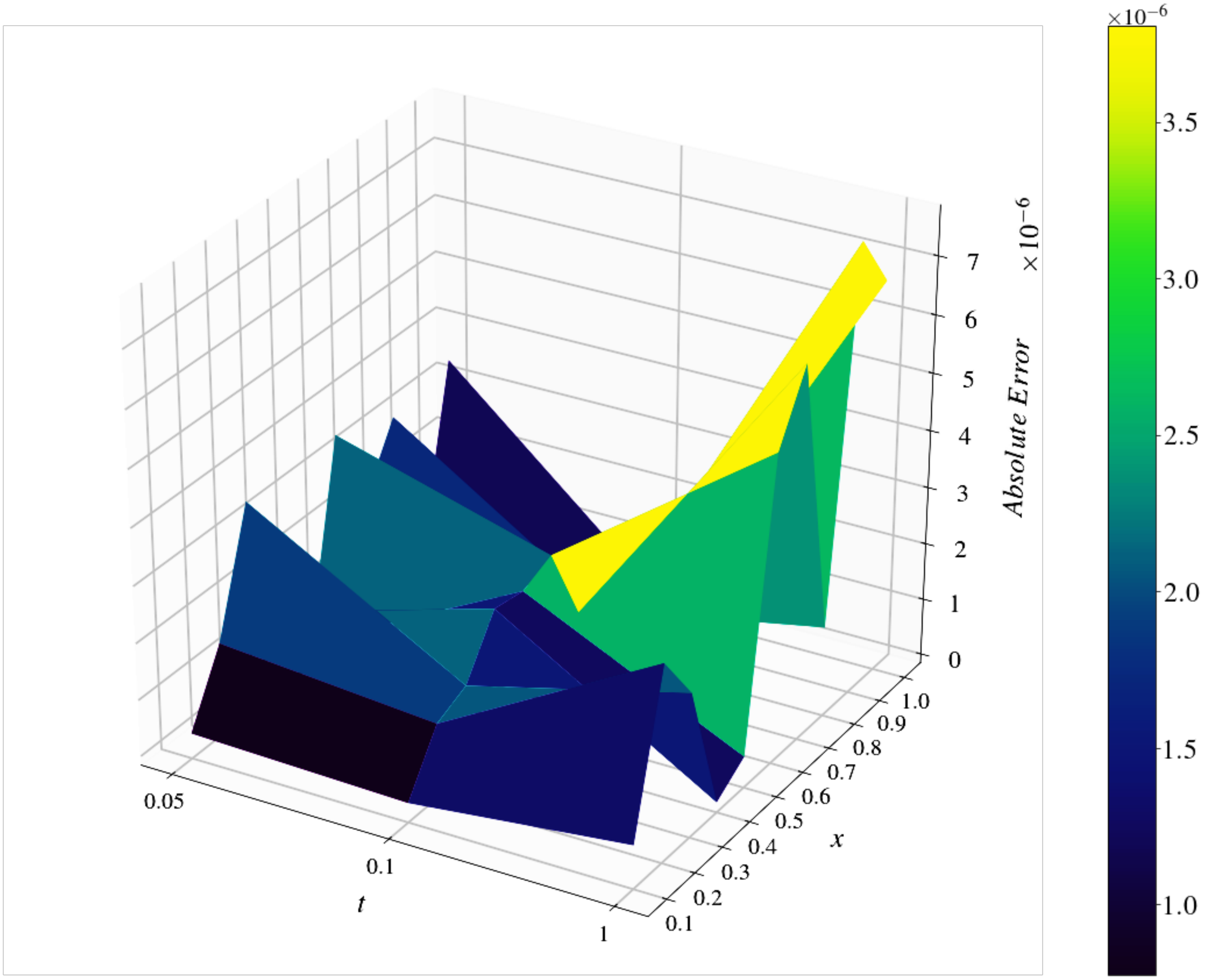

Figure 8: Absolute error for $p = 0$, $q = 1$, $s = 3$ & $r = 0.001$

It is clear from the findings of four instances in Table 2 - 5 that the DGM algorithm offers a solution that is comparable to other conventional numerical approaches, such as ADM, VIM, FDS4, OHAM, and MCB-DQM in terms of accuracy. In addition to accuracy, the DGM approach does not require domain discretization, whereas all of the aforementioned methods depend on the mesh. While the finite difference method substitutes derivatives by finite-difference formulas and the ADM method expresses the unknown function in terms of infinite series with nonlinear term is decomposed by an infinite series of polynomials, the DGM does not call for the linearization of nonlinear terms or dimensionality reduction. In OHAM, a series of equations are derived in terms of embedding parameter, non-zero auxiliary function, and unknown function, and the Taylor series is used to approximate the solution, which is not an easy task, in contrast, in DGM algorithm a cost function is constructed in terms of L^2- norm function. In MCB-DQM algorithm also given equation is reduced to a system of first order ordinary differential scheme and approximation methods are used to find the solution. In DGM algorithm DNN are used instead of basis function and cost function is constructed in terms of L^2- norm, which makes the algorithm simpler, efficient and generalized. It does not require any calculus effort unlike traditional numerical methods.

5. Conclusion

In this article, DGM algorithm is applied to provide the approximate solution of the gBHE and gHE. An architecture similar to GRU network architecture is used in implementation of the algorithm which is more faster and advantageous over other DNN architectures. This algorithm does not require linearization of nonlinear PDEs as well as dimension reduction, and it does not create meshes, which is an important feature because meshes become infeasible in higher dimensions. This method also removes the difficulty of construction of trial solution. The proposed method has been successfully implemented for obtaining the approximate solutions of the gBHE and gHE , which is fully demonstrated by the numerical results in terms of absolute error along with 3-D surface plot graph of absolute error. Further, convergence analysis of the cost function as well as neural network to the gBHE solution is also discussed. Numerical simulation has been done to compare the obtained DNN solution with the exact solution as well as the solution obtained by traditional methods such as ADM, VIM, OHAM, FDS4 and MCB-DQB. Numerical results shows the efficacy of the proposed method, which is reliable and encouraging.

References

[1] H. Bateman, Some recent researches on the motion of fluids, Monthly Weather Review 43 (4) (1915) 163–170.

[2] G. B. Whitham, Linear and nonlinear waves, John Wiley & Sons 42 (2011).

[3] J. M. Burgers, A mathematical model illustrating the theory of turbulence, Advances in applied mechanics 1 (1948) 171–199.

[4] A. L. Hodgkin, A. F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, The Journal of physiology 117 (4) (1952) 500–544.

[5] J. Satsuma, M. Ablowitz, B. Fuchssteiner, M. Kruskal, Topics in soliton theory and exactly solvable nonlinear equations, World Scientific (1987) 255–262.

[6] H. N. Ismail, K. Raslan, A. A. Abd Rabboh, Adomian decomposition method for burger's–huxley and burger's–fisher equations, Applied mathematics and computation 159 (1) (2004) 291–301.

[7] X. Wang, Nerve propagation and wall in liquid crystals, Physics Letters A 112 (8) (1985) 402–406.

[8] A. G. Bratsos, A fourth-order numerical scheme for solving the modified burgers equation, Computers & Mathematics with Applications 60 (5) (2010) 1393–1400.

[9] R. FitzHugh, Mathematical models of excitation and propagation in nerve, Biological engineering (1969) 1–85.

[10] X. Wang, Z. Zhu, Y. Lu, Solitary wave solutions of the generalised burgers-huxley equation, Journal of Physics A: Mathematical and General 23 (3) (1990) 271.

[11] I. Hashim, M. Noorani, B. Batiha, A note on the adomian decomposition method for the generalized huxley equation, Applied Mathematics and Computation 181 (2) (2006) 1439–1445.

[12] I. Hashim, M. S. M. Noorani, M. S. Al-Hadidi, Solving the generalized burgers–huxley equation using the adomian decomposition method, Mathematical and Computer Modelling 43 (2006) 1404–1411.

[13] A.-M. Wazwaz, Analytic study on burgers, fisher, huxley equations and combined forms of these equations, Applied Mathematics and Computation 195 (2) (2008) 754–761.

[14] A. S. Bataineh, M. Noorani, I. Hashim, Analytical treatment of generalized burgers-huxley equation by homotopy analysis method, Bulletin of the Malaysian Mathematical Sciences Society 32 (2) (2009) 233–243.

[15] A. Molabahrami, F. Khani, The homotopy analysis method to solve the burgers–huxley equation, Nonlinear Analysis: Real World Applications 10 (2) (2009) 589–600.

[16] B. Batiha, M. Noorani, I. Hashim, Application of variational iteration method to the generalized burgers–huxley equation, Chaos, Solitons & Fractals 36 (3) (2008) 660–663.

[17] O. Y. Yefimova, N. Kudryashov, Exact solutions of the burgers-huxley equation, Journal of Applied Mathematics and Mechanics 3 (68) (2004) 413–420.

[18] H. Gao, R.-X. Zhao, New exact solutions to the generalized burgers–huxley equation, Applied Mathematics and Computation 217 (4) (2010) 1598–1603.

[19] G. Griffiths, W. E. Schiesser, Traveling wave analysis of partial differential equations: numerical and analytical methods with matlab and maple, Academic Press (2010).

[20] A. G. Bratsos, A fourth order improved numerical scheme for the generalized burgers—huxley equation, American Journal of Computational Mathematics 1 (03) (2011) 152–158.

[21] I. Wasim, M. Abbas, M. Amin, Hybrid B-spline collocation method for solving the generalized Burgers-Fisher and Burgers-Huxley equations, Mathematical Problems in Engineering 2018 (2018).

[22] M. Javidi, A modified chebyshev pseudospectral dd algorithm for the gbh equation, Computers & Mathematics with Applications 62 (9) (2011) 3366–3377.

[23] M. Sari, G. Gürarslan, A. Zeytinoğlu, High-order finite difference schemes for numerical solutions of the generalized burgers–huxley equation, Numerical Methods for Partial Differential Equations 27 (5) (2011) 1313–1326.

[24] M. Javidi, A. Golbabai, A new domain decomposition algorithm for generalized burger's–huxley equation based on chebyshev polynomials and preconditioning, Chaos, Solitons & Fractals 39 (2) (2009) 849–857.

[25] R. Mittal, R. Jiwari, Numerical study of burger–huxley equation by differential quadrature method, Journal of Applied Mathematics and Mechanics 5 (2009) 1–9.

[26] R. Nawaz, H. Ullah, S. Islam, M. Idrees, Application of optimal homotopy asymptotic method to burger equations, Journal of Applied Mathematics 2013 (2013) 8.

[27] R. Mohammadi, B-spline collocation algorithm for numerical solution of the generalized burger's-huxley equation, Numerical Methods for Partial Differential Equations 29 (4) (2013) 1173–1191.

[28] I. Wasim, M. Abbas, M.K. Iqbal, A new extended B-spline approximation technique for second order singular boundary value problems arising in physiology, Journal of Mathematics and Computer Science 19 (4) (2019) 258–267.

[29] M.K. Iqbal, M. Abbas, N. Khalid, New cubic B-spline approximation for solving non-linear singular boundary value problems arising in physiology, Communications in Mathematics and Applications 9 (3) (2018) 377–392.

[30] M.K. Iqbal, M. Abbas,I. Wasim, New cubic B-spline approximation for solving third order Emden–Flower type equations, Applied Mathematics and Computation 331 (2018) 319–333.

[31] M.K. Iqbal, M. Abbas, B.Zafar, New quartic B-spline approximation for numerical solution of third order singular boundary value problems, Punjab University Journal of Mathematics 51 (5) (2020) 43–59.

[32] M. K. Iqbal, M. Abbas, B. Zafar, New quartic B-spline approximations for numerical solution of fourth order singular boundary value problems, Punjab University Journal of Mathematics 52 (3) (2020) 47–63.

[33] M.K. Iqbal, M.W. Iftikhar, M.S. Iqbal, M. Abbas, Numerical treatment of fourth-order singular boundary value problems using new quartic B-spline approximation technique, International Journal of advanced and applied sciences 7 (6) (2020) 48–56.

[34] Y. Bai, T. Chaolu, S. Bilige, Solving Huxley equation using an improved PINN method, Nonlinear Dynamics 105 (4) (2021) 3439–3450.

[35] Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, J. Kurths, Solving Fokker-Planck equation using deep learning, Chaos: An Interdisciplinary Journal of Nonlinear Science 30 (1) (2020) 013–133.

[36] H. Zhang, Y. Xu, Y. Li, J. Kurths, Statistical solution to SDEs with α-stable Lévy noise via deep neural network, International Journal of Dynamics and Control 8 (4) (2020) 1129–1140.

[37] H. Zhang, Y. Xu, Q. Liu, X. Wang, Y. Li, Solving Fokker–Planck equations using deep KD-tree with a small amount of data, Nonlinear Dynamics 108 (2022) 4029–4043.

[38] J. Sirignano, K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, Journal of computational physics 375 (2018) 1339–1364.

[39] K. Cho, B. V. Merriënboer, D. Bahdanau, Y. Bengio, On the Properties of Neural Machine Translation: Encoder–Decoder Approaches, Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (2014) 103–111.

[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[41] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings (2010) 249–256.

[42] D.P. Kingma, B. Jimmy, Adam: A Method for Stochastic Optimization, In Proceedings of the ICLR: International Conference for Learning Representations (2015).

[43] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural networks 4 (2) (1991) 251–257.

[44] J. Li, J. Yue, W. Zhang, W. Duan, The deep learning galerkin method for the general stokes equations, arXiv preprint arXiv:2009.11701 (2020).

[45] J. Li, W. Zhang, J. Yue, A deep learning galerkin method for the second-order linear elliptic equations., International Journal of Numerical Analysis & Modeling 18 (4) (2021) 427–441.

[46] O. A. Ladyzhenskaia, V. A. Solonnikov, N. N. Ural'tseva, Linear and quasi-linear equations of parabolic type, American Mathematical Society 23 (1988).

[47] D. Gilbarg, N. S. Trudinger, Elliptic partial differential equations of second order, Springer 224 (2015).

[48] J. Gaines, Numerical experiments with s(p) de's, London Mathematical Society Lecture Note Series (1995) 55–71.

[49] M. M. Porzio, Existence of solutions for some “noncoercive” parabolic equations, Discrete & Continuous Dynamical Systems 5 (3) (1999) 553–568.

[50] B. K. Singh, G. Arora, M. K. Singh, A numerical scheme for the generalized burgers–huxley equation, Journal of the Egyptian Mathematical Society 24 (4) (2016) 629–637.

Credit Author Statement

CRediT authorship contribution statement

Harender Kumar: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Neha Yadav:** Supervision, Validation. **Atulya Nagar:** Formal analysis, Writing – review & editing.