Numerical solution of the Sine-Gordon equation by novel physics-informed neural networks and two different finite difference methods

Svetislav Savović[1] *, Miloš Ivanović[1], Branko Drljača[2], Ana Simović[1]

[1] Faculty of Science, University of Kragujevac, R. Domanovića 12, 34000 Kragujevac, Serbia

[2] University of Priština in Kosovska Mitrovica, Faculty of Sciences and Mathematics, L. Ribara 29, 38220 Kosovska Mitrovica, Serbia

Abstract

This study employs a novel physics-informed neural networks (PINN) approach, standard explicit finite difference method (EFDM) and Chen-Charpentier et al.'s finite difference method (CCFDM) to tackle the one-dimensional Sine-Gordon equation (SGE). Two test problems with known analytical solutions are investigated to demonstrate the effectiveness of these techniques. While the three employed approaches demonstrate strong agreement, our analysis reveals that the EFDM results are in the best agreement with the analytical solutions. Given the consistent agreement between the numerical results from the EFDM, CCFDM, PINN approach and the analytical solutions, all three methods are recommended as competitive options. The solution techniques employed in this study can be a valuable asset for present and future model developers engaged in various nonlinear physical wave phenomena, such as propagation of solitons in optical fibers.

Keywords: Sine-Gordon equation; nonlinear wave phenomena; solitons; finite difference method; physics-informed neural networks

* Corresponding author (S. Savović). E-mail address: savovic@kg.ac.rs

1. Introduction

Partial differential equations (PDEs) offer a robust framework for characterizing a broad spectrum of phenomena in engineering, mathematics, physics, biology, and chemistry. They are particularly adept at describing processes such as heat transfer, fluid dynamics, wave propagation in electronic circuits, relativistic field theory, mechanical transmission lines, and chemical reactions. A prominent example of a nonlinear hyperbolic PDE is the SGE, which traces its origins to the nineteenth century and initially emerged in studies of surfaces exhibiting constant negative curvature [1-2]. This equation finds widespread use in simulating and elucidating various physical phenomena spanning multiple scientific domains. In condensed matter physics, it serves to investigate phenomena like solitons (commonly referred to as "kink" and "antikink") and topological defects [3]. In nonlinear optics, the SGE models the propagation of optical pulses in nonlinear media, particularly within optical fibers [4]. Additionally, in superconductivity research, the SGE elucidates the behavior of Josephson junctions, crucial components in superconducting devices [5]. Moreover, the equation has applications in surface science, where it delineates the dynamics of atoms and molecules on surfaces, including the propagation of surface waves [6]. Furthermore, in biophysics, the SGE aids in modeling phenomena such as nerve impulse propagation and protein dynamics [7]. These instances illustrate merely a fraction of the extensive applications of the SGE across diverse scientific and engineering challenges.

The features of the SGE have attracted considerable research interest, leading to investigations utilizing a variety of analytical and numerical techniques due to its broad applicability. Analytical methods such as the tan method, rational Exp-function method, sech method, extended tan method, and sine–cosine method have been applied to solve the double SGE [8]. The Daftardar-Gejji and Jafari method has been utilized to derive an approximate analytical solution for the SGE [9].

Deresse [10] successfully integrated the double Sumudu transform with an iterative approach to approximate the solution for the one-dimensional coupled SGE. Azari et al. [11] employed a local radial basis function based on finite difference for numerical solutions of the SGE. Significant efforts have been devoted to developing reliable numerical techniques for handling the SGE in recent decades. Hong et al. [12] introduced novel classes of fully discrete energy-preserving numerical algorithms for the SGE under Neumann boundary conditions. Moghaderi et al. [13] proposed a multigrid compact FDM for solving the one-dimensional nonlinear SGE. Babu and Asharaf [14] utilized a differential quadrature technique based on a modified set of cubic B-splines to numerically solve both one and two-dimensional SGEs, as well as their coupled form. Shiralizadeh et al. [15] implemented the numerical method of the rational radial basis function to solve perturbed and unperturbed SGEs with Dirichlet or Neumann boundary conditions, particularly suited for cases with steep fronts or sharp gradients. Additionally, the authors investigated the two-dimensional stochastic time fractional SGE using the clique polynomial approach [16]. Novkoski et al. [17] utilized a procedure for computing the direct scattering transform of the periodic SGE. These recent advancements underscore the growing interest in addressing the challenges posed by the SGE, with researchers employing diverse numerical and analytical techniques to explore its solutions and properties.

This paper aims to compare the numerical solutions of the SGE obtained through a novel deep learning-based approach called PINN as well as the standard EFDM and CCFDM [18]. The EFDM and CCFDM rely on a numerical technique known as FDM to approximate the derivatives of a function at discrete points within the domain. To compute function values, a grid of discrete points is established across the domain, and the approximate derivatives are calculated by evaluating the differences between function values at neighboring locations. The accuracy of FDM depends on

factors such as the grid step size and the order of approximation used for derivative calculations. On the other hand, PINN leverages machine learning to solve PDEs. Within PINN, a neural network is trained to grasp the underlying physics of a system and approximate the solution to a PDE. This enables PINNs to encapsulate the essential physics of the problem and deliver accurate solutions across the entire domain. The neural network is trained to minimize the residual of the PDE, measuring the discrepancy between the expected and actual solutions. A key advantage of PINN lies in its capacity to handle complex boundary conditions and geometries, which may pose challenges for traditional numerical techniques. However, it may entail significant computational costs and necessitate substantial amounts of training data. Additionally, the choice of hyperparameters, such as the number of layers and neurons in the neural network, can impact the performance of PINN.

Since PINN is a new method for solving the PDEs, there is a lack of evidence whether PINN can provide a high accuracy of the solutions of different PFEs, when compared to more conventional numerical techniques like the finite difference method. In our previous work, we demonstrated that EFDM outperformed the numerical solutions computed using PINN [18] in solving the nonlinear parabolic differential equation of Burgers' type. In order to better examine the effectiveness and precision of the PINN in addressing various kinds of nonlinear PDEs, in our best knowledge for the first time, we compare in this work the accuracy of the EFDM, CCFDM and PINN approach for solving the SGE for two test problems with different initial and boundary conditions. This is in contrast to other works which reported numerical results obtained using only conventional numerical techniques (not PINN), for example, two classes of structure-preserving algorithms for Sine-Gordon equation, which are based on the projection approach and the supplementary variable method [12].

The numerical results reported in this study can be a valuable resource for explanation a various nonlinear optical phenomena, including the propagation of solitons in optical fibers.

2. The Sine-Gordon equation

We considered the SGE which is a nonlinear PDE of hyperbolic type given by:

$$\frac{\partial^2 u(x,t)}{\partial t^2} = D\frac{\partial^2 u(x,t)}{\partial x^2} - \sin(u(x,t)) + F(x,t), \quad a \le x \le b, \;\; 0 \le t \le T\,, \tag{1}$$

with boundary conditions:

$$u(x=a,t) = g_a(t), \quad u(x=b,t) = g_b(t), \quad 0 \le t \le T \tag{2}$$

and initial condition:

$$u(x,t=0) = h(x), \quad a \le x \le b \tag{3}$$

where *t* and *x* are the time and space variables, respectively.

It is worth noting that equation (1) for $F(x,t)=0$, $\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=a,b} = 0$, $h(x) = 4\arctan\left(\exp\left(\frac{x}{\sqrt{1-c^2}}\right)\right)$, and $\frac{\partial u(x,t=0)}{\partial x} = -2\frac{c}{\sqrt{1-c^2}}\operatorname{sech}\left(\frac{x}{\sqrt{1-c^2}}\right)$, describes kink solitons moving with velocity *c*, while for $F(x,t)=0$, $\left.\frac{\partial u(x,t)}{\partial x}\right|_{x=a,b} = 0$, $h(x) = 4\arctan\left(\exp\left(-\frac{x}{\sqrt{1-c^2}}\right)\right)$, and $\frac{\partial u(x,t=0)}{\partial x} = -2\frac{c}{\sqrt{1-c^2}}\operatorname{sech}\left(\frac{x}{\sqrt{1-c^2}}\right)$, describes antikink solitons moving with velocity *c*. This special case of SGE is of high importance in examining nonlinear phenomena in optical fibers.

3. Finite difference methods

3.1 Explicit finite difference method

Using the EFDM, where the central FD schemes are used to represent derivative terms $\partial^2 u(x,t)/\partial x^2 = \left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right)/(\Delta x)^2$ and $\partial^2 u(x,t)/\partial t^2 = \left(u_i^{j+1} - 2u_i^j + u_i^{j-1}\right)/$

$(\Delta t)^2$, and assuming $D = \pi^{-2}$, and $F(x,t) = \sin[\cos(\pi x)\cos(t)\,]$, equation (1) is written in the following form:

$$\frac{u_i^{j+1}-2u_i^j+u_i^{j-1}}{(\Delta t)^2} = D\,\frac{u_{i+1}^j-2u_i^j+u_{i-1}^j}{(\Delta x)^2} - \sin\left(u_i^j\right) + F(x_i,t_j) \tag{4}$$

where $u_i^j \equiv u(x_i,t_j)$, indexes *i* and *j* refer to the discrete step lengths Δx and Δt for the coordinate *x* and time *t*. The grid dimensions in *x* and *t* directions are $K = 1/\Delta x$ and $M = T/\Delta t$, respectively. Using the FD scheme, the initial condition (2) and boundary conditions (3) are given as:

$$u_i^1 = h(x_i), \quad a \le x_i \le b, \;\; i = 1,2,\ldots,K \;\; (t=0) \tag{5}$$

$$u_1^j = g_a(t_j), \quad u_K^j = g_b(t_j), \quad j = 1,2,\ldots,M \;\; (x = a \text{ and } x = b) \tag{6}$$

Equation (4) represents a formula for u_i^{j+1} at the $(i, j+1)^{\text{th}}$ mesh point in terms of the known values along the j^{th} time row. The truncation error for scheme (4) is $\epsilon_T = O(\Delta t^2+\Delta x^2)$, which can be reduced using small enough values of Δt and Δx until the precision attained is within the error tolerance.

3.2. Chen-Charpentier finite difference method

Using the CCFDM, derivative terms are represented as $\partial^2 u(x,t)/\partial x^2 = \left(u_{i+1}^j - 2u_i^{j+1} + u_{i-1}^j\right)/(\Delta x)^2$ [19], and $\partial^2 u(x,t)/\partial t^2 = \left(u_i^{j+1} - 2u_i^j + u_i^{j-1}\right)/(\Delta t)^2$, equation (1) is written in the following form:

$$\frac{u_i^{j+1}-2u_i^j+u_i^{j-1}}{(\Delta t)^2} = D\,\frac{u_{i+1}^j-2u_i^{j+1}+u_{i-1}^j}{(\Delta x)^2} - \sin\left(u_i^j\right) + F(x_i,t_j) \tag{7}$$

where $u_i^j \equiv u(x_i,t_j)$, indexes *i* and *j* refer to the discrete step lengths Δx and Δt for the coordinate *x* and time *t*. The grid dimensions in *x* and *t* directions are $K = 1/\Delta x$ and $M = T/\Delta t$, respectively. Using the CCFD scheme (7), the initial condition is given in equation (5) and boundary conditions are given in equation (6). For scheme (7), the truncation error is $\epsilon_T = O(\Delta t^2+\Delta x^2)$, which can be decreased by choosing a small enough values of Δt and Δx.

4. Physics-informed neural networks

4.1 The basic concept of the physics-informed neural networks is solving PDEs

The PINN offers a machine-learning approach for approximate solutions to PDEs. The general form of PDEs, along with their associated initial and boundary conditions, is as follows:

$$\begin{aligned} &\frac{\partial u(x,t)}{\partial t} + N\left[u(x,t)\right] = 0, \quad x \in \Omega, \ \ t \in [0,T] \\ &u(x,t=0) = h(x), \ \ x \in \Omega \\ &u(x,t) = g(x,t), \ \ x \in \Omega_g, \ \ t \in [0,T] \end{aligned} \tag{8}$$

Here, N is a differential operator, $x \in \Omega \subseteq R^d$ and $t \in R$ represent spatial and temporal dimensions respectively, $\Omega \subseteq R^d$ is a computational domain, $\Omega_g \subseteq \Omega$ is a computational domain of the exposed boundary conditions, $u(x,t)$ is the solution of the PDEs with initial condition $h(x)$ and boundary conditions $g(x,t)$.

The approximator network and the residual network are the two subnets that make up PINN in its original construction [20]. The approximator network receives input (x,t), undergoes the training process, and provides an approximate solution $\widehat{u}(x,t)$ as an output. The approximator network utilized for training employs a grid of points, known as collocation points, sampled either randomly or regularly from the simulation domain. To update the weights and biases of the approximator network during training, a composite loss function of the following structure was minimized:

$$L = L_r + L_0 + L_b \tag{9}$$

where:

$$L_r = \frac{1}{N_r}\sum_{i=1}^{N_r}\left|u(x^i,t^i) + N[u(x^i,t^i)]\right|^2$$

$$L_0 = \frac{1}{N_0}\sum_{i=1}^{N_0}\left|u(x^i,t^i)-h^i)\right|^2 \tag{10}$$

$$L_b = \frac{1}{Nb}\sum_{i=1}^{N_b}\left|u(x^i,t^i)-g^i)\right|^2$$

Here, L_r, L_0, and L_b represent residuals of governing equations, initial and boundary conditions, respectively. N_r, N_0, and N are the numbers of mentioned collocation points of the computational domain, initial and boundary conditions, respectively. The residual network, an integral but non-trainable component within the PINN model, computes these residuals. To calculate the residual L_r in PINN, derivatives of the outputs concerning the inputs *x* and *t* are required. Automated differentiation is employed for this purpose, as it enables the aggregation of derivatives from individual constituent operations to compute the derivative of the entire composition. This methodology distinguishes PINNs from earlier endeavors in the early 1990s, which relied on manually deriving back-propagation rules. This approach stands as a pivotal facilitator for the development of PINNs. Presently, most deep learning frameworks, including TensorFlow and PyTorch, possess robust automatic differentiation capabilities, eliminating the need for laborious derivations or numerical discretization when computing derivatives across all spatial and temporal orders.

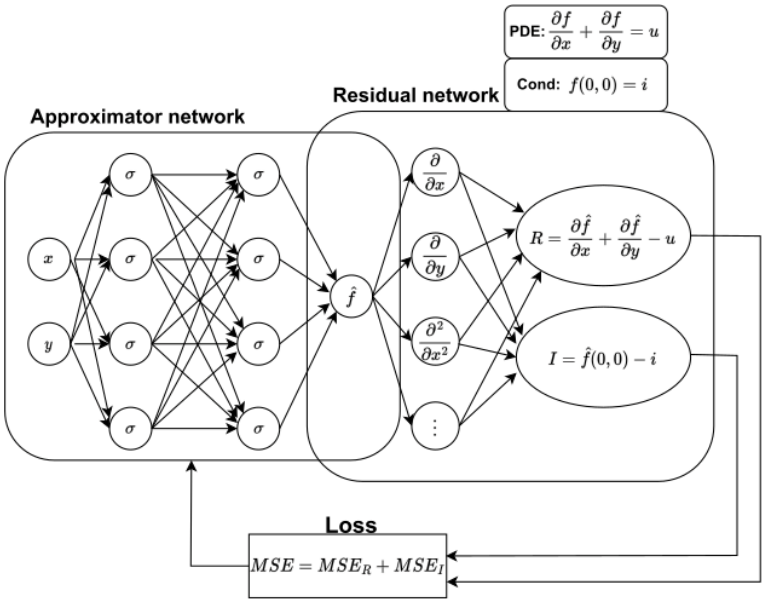


Figure 1. The architecture of a PINN and its standard training procedure were developed to tackle a basic partial differential equation, with PDE and Cond denoting the governing equations, and *R* and *I* representing the residuals. Following training, the approximator network furnishes an estimated solution. The residual network, an intrinsic but non-trainable element of PINN, is adept at computing derivatives of the approximator network outputs with respect to inputs and generating the composite loss function, symbolized by MSE.

A schematic of the PINN is demonstrated in Figure 1 in which a simple partial differential equation $\partial f/\partial x+\partial f/\partial y=u$ is used as an example. The approximator network is used to approximate the solution $u(x,t)$ which then goes to the residual network to calculate the residual loss L_r, boundary condition loss L_b, and initial condition loss L_0. The weights and biases of the approximator network are trained using a composite loss function consisting of residuals L_r, L_0, and L_b through gradient-descent technique based on the back-propagation.

4.2 Implementation of PINN in solving the Sine-Gordon equation

For the development of the PINN model to solve the SGE, we utilized the DeepXDE library [21]. Our PINN architecture comprises two inputs (x,t) and consists of 3 layers, each containing 40 neurons with *sigmoid* activation. We initially determined the hyperparameter values using our evolutionary optimization framework proposed in [22], followed by additional manual fine-tuning. In a previous study [18], we employed the tanh activation function to solve the Burgers equation. However, in this study, the sigmoid activation function proved to deliver the best performance and accuracy. The set of collocation points is divided into three subsets. The largest subset, comprising 8000 collocation points, corresponds to the general problem domain. The second and third subsets, containing 400 and 800 collocation points respectively, are utilized to enforce boundary and initial conditions. We experimented with varying weights for the initial and boundary conditions as well as different numbers of interior collocation points. When the boundary condition weights were too low, oscillations appeared near the domain edges. Additionally, models trained with 4000 and 2000 interior collocation points exhibited increased relative error rates, rising from 2% (for 8000) to 9% and 11%, respectively. Further increasing the number of collocation points, however, showed no significant improvement in accuracy. These conditions remain consistent across all test cases. The PINN training process involves two stages. Initially, we employ the Adam algorithm to optimize weights and biases for 60000 epochs in the first phase, using a learning rate of 10^{-3}. Subsequently, in the second phase, following completion of a "global" search, the Limited Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm is employed to further approach the optimal solution according to [23]. The entire training procedure, executed on the nVidia Tesla T4 GPU accelerator, takes approximately 248 seconds. It is conceivable that employing different hyper-parameters, such as alternative activation functions, training methods, and PINN topologies, may yield improved solutions in practice. However, we opted for hyper-parameter values commonly used in the SGE literature, acknowledging that identifying optimal hyper-parameters is a demanding and time-intensive task beyond the scope of our study.

This paper compares the accuracy of numerical results obtained for two test problems of the SGE using EFD, CCFD and PINN methods against analytical solutions documented in the literature.

5. Numerical results and discussion

To illustrate the accuracy of the EFDM, CCFDM and PINN approach, several numerical computations are carried out for two test problems.

Test problem 1: Consider the SGE (1) with the initial condition:

$$u(x, t = 0) = \cos(\pi x), \quad 0 < x < 1 \tag{11}$$

and boundary conditions:

$$u(x = 0, t) = \cos(t), \quad 0 \leq t \leq 2 \tag{12}$$

$$u(x = 1, t) = -\cos(t), \quad 0 \leq t \leq 2$$

Assuming $D = \pi^{-2}$ and $F(x, t) = \sin[\cos(\pi x)\cos(t)\,]$ in SGE (1), its EFD solution can be written as:

$$\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\Delta t)^2} = \frac{1}{\pi^2}\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} - \sin\left(u_i^j\right) + \sin\left[\cos(\pi x_i)\cos(t_j)\,\right] \tag{13}$$

$$u_i^{j+1} = ru_{i-1}^j + 2(1-r)u_i^j + ru_{i+1}^j - u_i^{j-1} - (\Delta t)^2 \sin\left(u_i^j\right) + (\Delta t)^2 \sin\left[\cos(\pi x_i)\cos(t_j)\,\right]$$

where $= (\Delta t)^2/(\pi \Delta x)^2$.

CCFD solution of SGE (1) can be written as:

$$\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{(\Delta t)^2} = \frac{1}{\pi^2}\frac{u_{i+1}^j - 2u_i^{j+1} + u_{i-1}^j}{(\Delta x)^2} - \sin\left(u_i^j\right) + \sin\left[\cos(\pi x_i)\cos(t_j)\,\right] \tag{14}$$

$$u_i^{j+1} = \{ru_{i-1}^j + 2u_i^j + ru_{i+1}^j - u_i^{j-1} - (\Delta t)^2 \sin\left(u_i^j\right) + (\Delta t)^2 \sin\left[\cos(\pi x_i)\cos\left(t_j\right)\right]\}/(1+2r)$$

where $= (\Delta t)^2/(\pi \Delta x)^2$. The analytical solution of the problem is given as [24]:

$$u(x, t) = \cos(\pi x)\cos(t) \tag{15}$$

Equations (13) and (14) represent the EFD and CCFD solutions of this test problem, respectively. The initial condition (11) in terms of finite differences becomes:

$$u_i^1 = \cos(\pi x_i), \quad 0 < x_i < 1, \quad i = 1,2,\ldots,K \quad (t = 0) \tag{16}$$

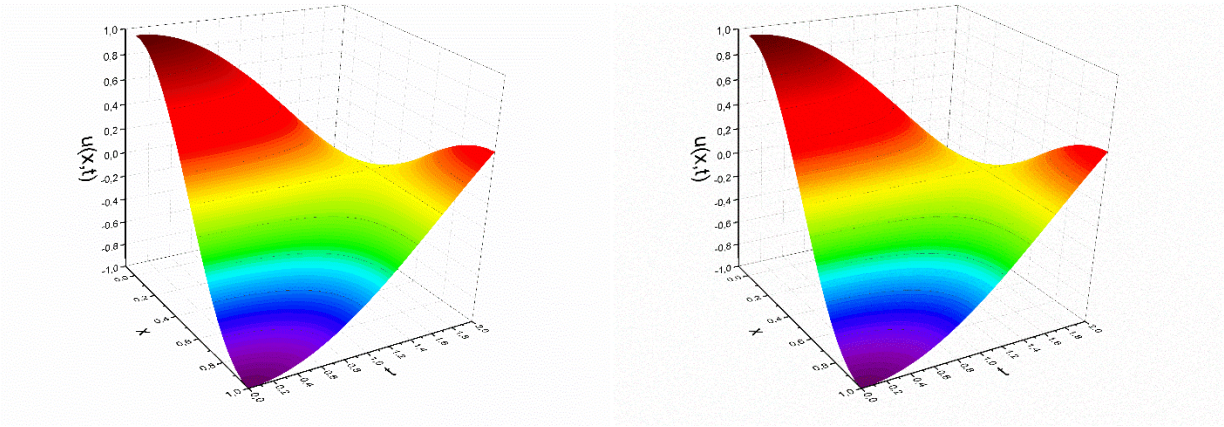
and boundary conditions (12) are given as:

$$u_1^j = \cos(t_j), \quad u_K^j = -\cos(t_j), \quad j = 1,2,\dots,M \quad (x = 0 \text{ and } x = 1) \tag{17}$$

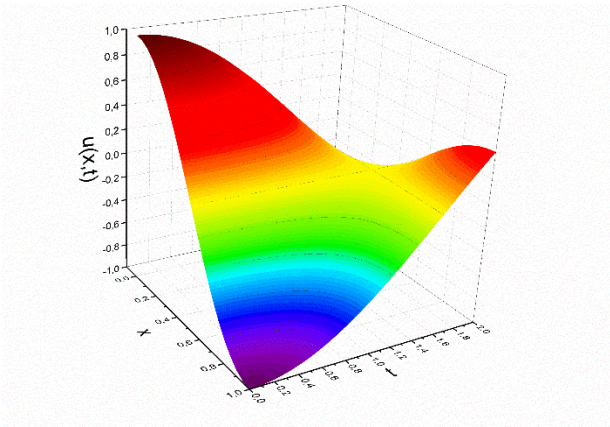
Figure 2 shows our numerical solution of the SGE (1) obtained using EFD scheme (step lengths are Δx=0.05 and Δt=0.0001), CCFD scheme (step lengths are Δx=0.05 and Δt=0.0001) and PINN in 3-dimensions. Since Figure 2 does not permit a precise comparison of the three numerical methods, we calculated the root mean square error, which is represented by:

$$Error = \sqrt{\frac{1}{N}\sum_{i=1}^{K}\left(u_i^{method} - u_i^{analit}\right)^2} \tag{18}$$

where K is the total number of observed points along x axis. Equation (18) was selected as the error function to assess the method's accuracy. A lower error value indicates that the method provides a better distribution of $u(x,t)$ over a given time interval. Table 1 (for T=0.1), Table 2 (for T=1.0), and Table 3 (for T=2.0) show the comparison of the numerical solutions of Test Problem 1 produced using the EFDM, CCFDM and PINN approach with the analytical solutions. The accuracy and computational time of the EFDM, CCFDM and PINN approach in solving Test problem 1 at different times T is shown in Tables 4 and 5. Since PINN training does not carry out classical time stepping scheme, all times in Table 5 are identical and equal to training time. The EFDM offers the best match with the analytical solution and the shortest computational time, it should be highlighted.



(a) (b)



(c)

Figure 2. (a) EFD, (b) CCFD and (c) PINN solutions of Test problem 1 in 3-dim at different times.

Table 1. Comparison of numerical and analytical solutions of the Test problem 1 for T=0.1.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	0.946216	0.948641	0.944360	0.946258
0.2	0.805308	0.804981	0.802940	0.804935
0.3	0.584988	0.584852	0.584184	0.584819

0.4	0.307206	0.307475	0.309658	0.307458
0.5	0.000000	0.000000	0.000000	0.000000
0.6	-0.307206	-0.307475	-0.302658	-0.307458
0.7	-0.584988	-0.584852	-0.582750	-0.584820
0.8	-0.805308	-0.804981	-0.805884	-0.804935
0.9	-0.946217	-0.948641	-0.947661	-0.946258
1.0	-0.994954	-0.994994	-0.995012	-0.994954

Table 2. Comparison of numerical and analytical solutions of the Test problem 1 for T=1.0.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	0.516640	0.522306	0.514519	0.513443
0.2	0.443820	0.446852	0.440810	0.436761
0.3	0.323080	0.330305	0.325625	0.317325
0.4	0.170390	0.169535	0.178679	0.166828
0.5	0.000000	0.000000	0.000000	0.000000
0.6	-0.170390	-0.169535	-0.152560	-0.166828
0.7	-0.323080	-0.330305	-0.30311	-0.317325
0.8	-0.443820	-0.446852	-0.42431	-0.436761
0.9	-0.516640	-0.522306	-0.50597	-0.513443
1.0	-0.539870	-0.540302	-0.54016	-0.539866

Table 3. Comparison of numerical and analytical solutions of the Test problem 1 for T=2.0.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	-0.395283	-0.399106	-0.389806	-0.396165
0.2	-0.335927	-0.337419	-0.328859	-0.336998
0.3	-0.241973	-0.246279	-0.237416	-0.244843
0.4	-0.127501	-0.128838	-0.122575	-0.128722
0.5	0.000000	0.000000	0.000000	0.000000
0.6	0.127500	0.128838	0.131383	0.128722
0.7	0.241976	0.246279	0.244583	0.244844
0.8	0.335927	0.337419	0.333709	0.336998
0.9	0.395283	0.399106	0.392060	0.396165
1.0	0.416553	0.416146	0.418024	0.416553

The accuracy of the EFDM, CCFDM and PINN approach in solving Test problem 1 at different times is shown in Table 4. The EFDM offers the best match with the analytical solution, it should be highlighted.

Table 4. The accuracy of EFDM, CCFDM and PINN approach of the Test problem 1 at different times T.

T	Error (EFDM)	Error (CCFDM)	Error (PINN)
0.1	2.07×10^{-4}	3.94×10^{-3}	4.51×10^{-3}
1.0	4.65×10^{-4}	8.42×10^{-3}	9.98×10^{-3}

2.0	1.52×10^{-3}	3.66×10^{-3}	4.76×10^{-3}

Table 5. Computational time of EFDM, CCFDM and PINN approach of the Test problem 1 at different times *T*.

T	Computational time (s) EFDM	Computational time (s) CCFDM	Computational time (s) PINN
0.1	1.5	2.0	248.0
1.0	20.5	21.5	248.0
2.0	38.5	41.0	248.0

Test problem 2: Consider the SGE (1) with the initial condition:

$$u(x, t = 0) = \sin(\pi x), \quad 0< x < 1 \tag{19}$$

and boundary conditions:

$$u(x = 0, t) = 0, \quad 0\leq t \leq 2 \tag{20}$$

$$u(x = 1, t) = 0, \quad 0\leq t \leq 2$$

Assuming $D = 1$ and $F(x, t) = 0$ in SGE (1), its EFD solution of SGE (1) can be written as:

$$u_i^{j+1} = su_{i-1}^j + 2(1 - s)u_i^j + su_{i+1}^j - u_i^{j-1} - (\Delta t)^2 \sin\left(u_i^j\right) \tag{21}$$

where $s = (\Delta t)^2/(\Delta x)^2$.

CCFD solution of SGE (1) can be written as:

$$u_i^{j+1} = \{su_{i-1}^j + 2u_i^j + su_{i+1}^j - u_i^{j-1} - (\Delta t)^2 \sin\left(u_i^j\right)\}/(1 + 2s) \tag{22}$$

The analytical solution of the problem is given as [25]:

$$u(x, t) = \frac{1}{2}\left[\sin\left(\pi(x + t)\right) + \sin\left(\pi(x - t)\right)\right] \tag{23}$$

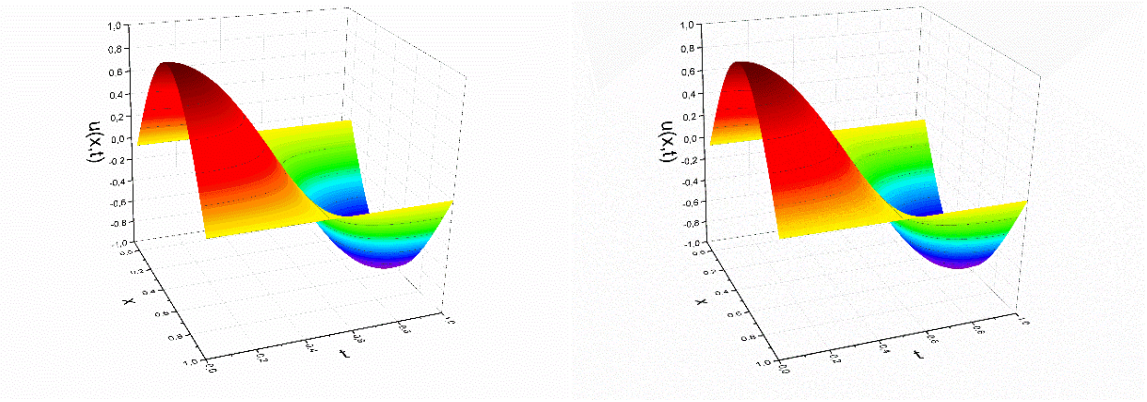
Equations (21) and (22) represent the EFD and CCFD solutions of this test problem, the initial condition (19) in terms of finite differences becomes:

$$u_i^1 = \sin(\pi x_i), \quad 0 < x_i < 1, \quad i = 1,2,\dots,K \quad (t = 0) \tag{24}$$

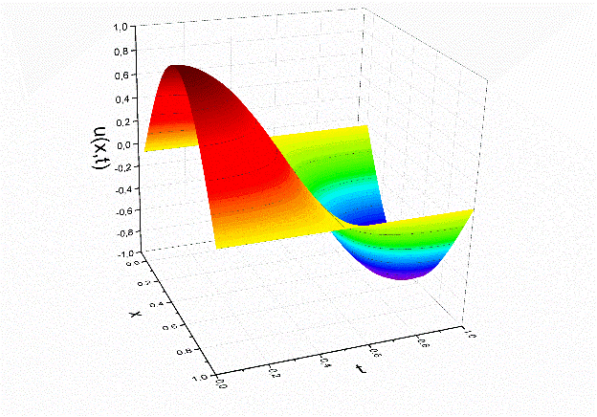
and boundary conditions (20) are given as:

$$u_1^j = 0, \quad u_K^j = 0, \quad j = 1,2,\dots,M \quad (x = 0 \text{ and } x = 1) \tag{25}$$

Figure 3 depicts our numerical solution of the SGE (1) obtained using EFD scheme (step lengths are Δx=0.05 and Δt=0.0001), CCFD scheme (step lengths are Δx=0.05 and Δt=0.0001) and PINN approach in 3-dimensions.



(a) (b)



(c)

Figure 3. (a) EFD, (b) CCFD and (c) PINN solutions of Test problem 2 in 3-dim at different times.

The comparison of the numerical solutions of the Test problem 2 obtained using the EFDM, CCFDM and PINN approach with analytical solution is displayed in Table 6 (for T=0.01), Table 7 (for T=0.1) and Table 8 (for T=1.0). The accuracy and computational times of the EFDM, CCFDM and PINN approach in solving Test problem 2 at different times T is shown in Tables 9 and 10. The EFDM offers the best match with the analytical solution, it should be highlighted.

Table 6. Comparison of numerical and analytical solutions of the Test problem 2 for T=0.01.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	0.308860	0. 308851	0.311214	0.308858
0.2	0.587472	0. 587471	0.590860	0.587484
0.3	0.808704	0. 808586	0.813018	0.808602
0.4	0.950430	0. 950552	0.955571	0.950568
0.5	0.999374	0. 999470	1.003996	0.999487
0.6	0.950430	0. 950552	0.953803	0.950568
0.7	0.808704	0. 808586	0.810964	0.808601
0.8	0.587472	0. 587471	0.590098	0.587483
0.9	0.308860	0. 308851	0.311204	0.308858
1.0	0.000000	0.000000	0.000000	0.000000

Table 7. Comparison of numerical and analytical solutions of the Test problem 2 for T=0.1.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	0.292114	0.292499	0.309108	0.293742
0.2	0.556125	0.556473	0.589712	0.558731
0.3	0.765498	0. 766098	0.811483	0.769027
0.4	0.900292	0.900768	0.951017	0.904046
0.5	0.946313	0.947189	0.994443	0.950570
0.6	0.900291	0.900768	0.939432	0.904046
0.7	0.765497	0.766098	0.794477	0.769027
0.8	0.556124	0.556473	0.576097	0.558731
0.9	0.292114	0.292499	0.305305	0.293742
1.0	0.000000	0.000000	0.000000	0.000000

Table 8. Comparison of numerical and analytical solutions of the Test problem 2 for T=1.0.

x	EFDM solution	CCFDM solution	PINN solution	Analytical solution
0.1	-0.305795	-0.294496	-0.311997	-0.309017
0.2	-0.582285	-0.560186	-0.590231	-0.587784
0.3	-0.801375	-0.771066	-0.812557	-0.809016
0.4	-0.942320	-0.906476	-0.957621	-0.951055
0.5	-0.990102	-0.953139	-1.010339	-0.999999
0.6	-0.942334	-0.906476	-0.963884	-0.951055

0.7	-0.801407	-0.771066	-0.821299	-0.809016
0.8	-0.582327	-0.560186	-0.596018	-0.587784
0.9	-0.305873	-0.294496	-0.310942	-0.309016
1.0	0.000000	0.0000000	0.000000	0.000000

Table 9. The accuracy of EFDM, CCFDM and PINN approach of the Test problem 2 at different times T.

T	Error (EFDM)	Error (CCFDM)	Error (PINN)
0.01	8.80×10^{-5}	9.87×10^{-5}	3.32×10^{-3}
0.1	3.09×10^{-3}	3.19×10^{-3}	9.98×10^{-3}
1.0	6.82×10^{-3}	6.99×10^{-3}	7.45×10^{-3}

Table 10. Computational time of EFDM, CCFDM and PINN approach of the Test problem 2 at different times T.

T	Computational time (s) EFDM	Computational time (s) CCFDM	Computational time (s) PINN
0.01	0.2	0.3	248.0
0.1	1.1	1.8	248.0
1.0	19.0	21.0	248.0

Finally, to the best of our knowledge, we compare the accuracy of the EFD, CCFDM and PINN approach in solving the SGE for the first time in this study. We found that the EFDM, with appropriately small step lengths Δx and Δt, demonstrates the best accuracy compared to the

numerical solutions generated using the CCFDM and PINN approach. The solution methodologies employed in this study can be a valuable resource for current and future model developers tackling a range of nonlinear physical wave phenomena, including the propagation of solitons in optical fibers.

5. Conclusion

We compared our numerical results for solving the nonlinear hyperbolic partial differential SGE, obtained using EFDM, CCFDM and a novel PINN approach, with the analytical solutions reported in the literature. We demonstrated that while all three employed methods show good agreement with the analytical solutions, while the EFDM exhibits superior accuracy compared to the numerical solutions generated using the CCFDM and PINN approach. Since the numerical results from all three methods closely match the analytical solutions, these approaches are competitive and worthy of endorsement. The solution techniques used in this study can be applied to developing numerical models for other nonlinear PDEs in the future.

Abbreviations:

CCFDM: Chen-Charpentier finite difference method

EFDM: Explicit finite difference method

L-BFGS: Limited memory Broyden-Fletcher-Goldfarb-Shanno

PDE: Partial differential equation

PINN: Physics informed neural networks

SGE: Sine-Gordon equation

Author Contributions: Conceptualization, S.S., M.I., B.D. and A.S.; methodology, software, S.S., M.I., B.D. and A.S.; writing—original draft preparation, B.D., S.S. and M.I.; writing—review and editing, S.S., M.I., B.D. and A.S.; supervision, S.S.; project administration, funding acquisition, M.I., S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Serbian Ministry of Science, Technological Development and Innovations (Agreement No. 451-03-65/2024-03/200122).

Disclosures: The authors declare no conflicts of interest.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

References:

[1] E. Bour, "Teorie de la deformation des surfaces," de l'Ecole imperiale polytechnique, 22, 1–148, 1861.

[2] J. Rubinstein, "Sine-Gordon equation," J. Math. Phys. 11, 258–266, 1970.

[3] L. M. Alonso, "Soliton classical dynamics in the sine-gordon equation in terms of the massive thirring model," Phys. Rev. D 30, 2595–2601, 1984.

[4] G. Agrawal, "Nonlinear Fiber Optics," Academic, Cambridge, MA, USA, 5th edition, 2013.

[5] M. Tinkham and V. Emery, "Introduction to Superconductivity," University of California, California, Irvine, 1996.

[6] V. G. Bykov, "Sine-Gordon equation and its application to tectonic stress transfer," J. Seismology 18, 497–510, 2014.

[7] A. Scott, "Nonlinear Science," Oxford University Press, Oxford, UK, 1999.

[8] H. Rezazadeh, A. Zabihi, A.G. Davodi, R. Ansari, H. Ahmad, and S.-W. Yao, “New optical solitons of double Sine-Gordon equation using exact solutions methods,” Results Phys. 49, 106452, 2023.

[9] B. Batiha, “New solution of the Sine-Gordon equation by the Daftardar-Gejji and Jafari method,” Symmetry 14, 57, 2022.

[10] A. T. Deresse, “Double sumudu transform iterative method for one-dimensional nonlinear coupled Sine-Gordon equation,” Advances Math. Phys. 2022, 6977692, 2022.

[11] Y. Azari, G. Garmanjani, and H. Rabiei, “Numerical solution of nonlinear Sine-Gordon equation with local RBF-based finite difference collocation method,” The 44th Annual Iranian Mathematics Conference 27-30 August 2013, Ferdowsi University of Mashhad, Iran, PP. 331–334.

[12] Q. Hong, Y. Wang, and Y. Gong, “Efficient energy-preserving numerical approximations for the sine-Gordon equation with Neumann boundary conditions,” arXiv:2004.10970v2 [math.NA] 11 Aug 2020.

[13] H. Moghaderi and M. Dehghan, “A multigrid compact finite difference method for solving the one-dimensional nonlinear sine-Gordon equation,” Math. Methods Appl. Sciences 38, 3901–3922, 2014.

[14] A. Babu and N. Asharaf, “Numerical solution of nonlinear sine-gordon equation using modifed cubic b-spline-based differential quadrature method,” Comput. Methods Differ. Equ. 11, 369–386, 2023.

[15] M. Shiralizadeh, A. Alipanah, and M. Mohammadi, “Numerical solution of one-dimensional sine-gordon equation using rational radial basis functions,” J. Math. Model. 10, 387–405, 2022.

[16] Z. Eidinejad, R. Saadati, J. Vahidi, and C. Li, "Numerical solutions of 2d stochastic time-fractional sine–Gordon equation in the Caputo sense," Int. J. Numer. Model.: Electronic Networks, Devices and Fields, 36, e3121, 2023.

[17] F. Novkoski, E. Falcon, and C. T. Pham, "A numerical direct scattering method for the periodic sine-Gordon equation," Eur. Phys. J. Plus 138, 1146, 2023.

[18] S. Savović, M. Ivanović, and R. Min, "A Comparative study of the explicit finite difference method and physics-informed neural networks for solving the Burgers' equation," Axioms 12, 982, 2023.

[19] B.M. Chen-Charpentier, and H. V. Kojouharov, "An unconditionally positivity preserving scheme for advection–diffusion reaction equations," Math. Comput. Model. 57, 2177–2185, 2013.

[20] 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," J. Computat. Phys. 378, 2019, 686–707, 2019.

[21] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," CoRR abs/1907.04502, 2019.

[22] A. Kaplarević-Mališić, B. Andrijević, F. Bojović, S. Nikolić, L. Krstić, B. Stojanović, and M. Ivanović, "Identifying optimal architectures of physics-informed neural networks by evolutionary strategy," Appl. Soft Computing, 146, 110646, 2023.

[23] S. Markidis, "The old and the new: can physics-informed deep-learning replace traditional linear solvers?", Front. big Data 4, 669097, 2021.

[24] A. T. Deresse, Y. O. Mussa, and A. K. Gizaw, "Analytical solution of two-dimensional sine-gordon equation," Advances Math. Phys. 2021, 6610021, 2021.

[25] K. R. Raslan, A. A. Soliman, Khalid K. Ali, M. Gaber, and S. R. Almhdy, “Numerical solution for the Sine-Gordon equation using the finite difference method and the non-standard finite difference method,” Appl. Math. Inf. Sci. 17, 253–260, 2023.