applied sciences

MDPI

Article

Numerical Solutions of the Nonlinear Dispersive Shallow Water Wave Equations Based on the Space–Time Coupled Generalized Finite Difference Scheme

Po-Wei Li *, Shenghan Hu and Mengyao Zhang

School of Mathematics and Statistics, Qingdao University, Qingdao 266071, China; shenghan2001@163.com (S.H.); zmengyao1016@163.com (M.Z.)
* Correspondence: wade0701@hotmail.com

Abstract: This study applies the space–time generalized finite difference scheme to solve nonlinear dispersive shallow water waves described by the modified Camassa–Holm equation, the modified Degasperis–Procesi equation, the Fornberg–Whitham equation, and its modified form. The proposed meshless numerical scheme combines the space–time generalized finite difference method, the two-step Newton's method, and the time-marching method. The space–time approach treats the temporal derivative as a spatial derivative. This enables the discretization of all partial derivatives using a spatial discretization method and efficiently handles mixed derivatives with the proposed mesh-less numerical scheme. The space–time generalized finite difference method is derived from Taylor series expansion and the moving least-squares method. The numerical discretization process only involves functional data and weighting coefficients on the central and neighboring nodes. This results in a sparse matrix system of nonlinear algebraic equations that can be efficiently solved using the two-step Newton's method. Additionally, the time-marching method is employed to advance the space–time domain along the time axis. Several numerical examples are presented to validate the effectiveness of the proposed space–time generalized finite difference scheme.

Keywords: nonlinear shallow water wave; meshless methods; space–time generalized finite difference method; Degasperis–Procesi equation; Fornberg–Whitham equation

Citation: Li, P.-W.; Hu, S.; Zhang, M. Numerical Solutions of the Nonlinear Dispersive Shallow Water Wave Equations Based on the Space–Time Coupled Generalized Finite Difference Scheme. *Appl. Sci.* **2023**, *13*, 8504. https://doi.org/10.3390/app13148504

Academic Editor: Cesare Biserni

Received: 10 June 2023
Revised: 12 July 2023
Accepted: 21 July 2023
Published: 23 July 2023

1. Introduction

Dispersive shallow water waves are wave propagation phenomena that have received extensive attention in engineering due to their numerous applications in physics, engineering, mechanics, and mathematics. In the past few decades, several partial differential equations (PDEs), including the Camassa−Holm equation [1,2], the Degasperis−Procesi equation [2–4], the Fornberg−Whitham equation [3,5–12], and the Schrödinger equation [13–17], have been presented to describe the traveling wave phenomenon and water waves of finite depths. These PDEs and their modified forms mathematically represent transient nonlinear PDEs that involve third-order time–space mixed partial derivatives or complex terms. Therefore, obtaining the numerical solutions of those PDEs is a significant issue for analyzing the traveling wave phenomenon. In previous studies, two types of methods were mainly used to obtain solutions for such equations: the approximate method, the spectral method [18,19], and numerical simulation.

Due to the complexity of these equations, finding exact solutions is generally challenging. Therefore, numerical methods play a crucial role in solving these equations. Traditional numerical approaches, relying on temporal and spatial discretization, face difficulties when dealing with high-order mixed partial derivatives. As a result, researchers have extensively employed approximate methods to derive numerical solutions, which offer convenient and effective alternatives to classical numerical treatments. For example, Wazwaz [2] used the extended tanh method to establish the new solitary wave solutions

of the modified Degasperis−Procesi and Camassa−Holm equations. Abidi and Omrani [10] applied the homotopy analysis method (HAM) to acquire the HAM results of the Fornberg−Whitham equation and compared them with Adomian's decomposition method (ADM). Ahmad et al. [5] utilized the modified variational iteration algorithm (MVIA) to obtain the approximate numerical solutions of the Fornberg−Whitham equation and its modified form. In addition to the above-mentioned methods, there are other approximate methods such as the method of phase portrait analysis [9], He's variational iteration method (VIM) [8], and the reproducing kernel Hilbert space method (RKHSM) [7]. Recently, due to the evolution of computer technology and numerical methods, novel numerical schemes have also been applied to solve such equations. These include the Godunov method [6], the cubic B-spline quasi-interpolation method [4], the hybrid radial basis function method [3], the linearized implicit pseudo-spectral method [18], and the Gegenbauer spectral method [19].

In the field of numerical discretization methods for PDEs, two main categories can be distinguished: mesh-based methods and meshless/free methods. Mesh-based methods include well-known techniques such as finite difference (FDM), finite volume, and finite element methods. These methods have been extensively developed and modified to solve engineering problems and analyze complex physical phenomena that are challenging to investigate through practical experiments. Despite its widespread application, mesh generation remains a challenging task. To overcome this issue, the concept of using numerical discretization methods without the need for a mesh has emerged as an important research direction. Meshless methods offer a solution by avoiding the requirement for mesh generation, making it easier to handle computational domains with complex shapes. In meshless methods, numerical discretization involves constructing interpolation functions on arbitrary distribution nodes to approximate the governing equation. Over the years, meshless methods have gained advantages such as simplified numerical procedures, easy implementation, flexibility, and the ability to construct hybrid numerical schemes tailored to specific research needs. At present, there are several meshless numerical methods, for example, the radial basis functions collocation method (RBFCM) [3], the virtual boundary meshless Galerkin method [20], the moving particle semi-implicit method [21], the radial point interpolation meshless method [22], the singular boundary method [23], the localized scheme based on boundary-type method [24–26], the generalized finite difference method (GFDM) [27–37], etc.

The GFDM is a localized domain-type meshless collocation method based on the multivariate Taylor series expansion theory and the moving least-squares method. In 1980, the idea of FDM without mesh was proposed by Liszka and Orkisz [27]. After Benito et al. [28] proposed the explicit numerical process in 2001, numerical applications related to the GFDM have been proposed in the past two decades. By utilizing the GFDM, the approximate expression of the partial derivative terms can be represented by a linear combination of the weight coefficients and function values in a local support domain. This support domain only contains the goal node and its nearby nodes. Once the nodes are distributed in the computational domain, the governing equation can be discretized into a system of algebraic equations. In addition, the resultant matrix system is a sparse matrix due to the localized property of the GFDM. The above characteristics make the GFDM capable of applying to complex problems, such as theoretical analysis [29], shallow water equation [30,31], porous media flow [32], wave propagation [33], elliptic interface problems [34], extended Fisher–Kolmogorov equation [35], stream function formulation [36], and elastic wave [37]. In this research, the space–time GFDM (ST-GFDM) was applied as the foundation of the proposed meshless numerical scheme, and it is an extended meshless method from the GFDM.

The space–time (ST) coupled approach is a numerical technique that treats the temporal derivative as one of the spatial derivatives. Prior to the use of the ST-coupled approach, time-dependent problems were typically solved by combining spatial discretization methods with temporal discretization methods. However, this approach often led to challenges

in defining the accuracy and stability of the numerical scheme. The ST-coupled approach transforms a time-dependent problem of dimension (n) into a steady-state problem of dimension (n + 1). As a result, all the partial derivatives in the governing equation can be discretized using numerical methods designed for spatial discretization. This simplification enables a clearer determination of the properties of the numerical scheme, such as accuracy and stability. Recently, the ST coupled approach has been widely combined with meshless methods such as the ST localized RBFCM [38], ST kernel-based method [39], ST localized method of fundamental solutions [40,41], ST Trefftz Method [42,43], ST backward substitution method [44], and ST-GFDM [45–54]. Based on the flexibility of the ST-GFDM, researchers have applied this meshless method for engineering problems in the past few years. Examples of these applications include heat conduction problems [45,54], Burgers' equations [46], parabolic PDEs [47], unsteady double-diffusive natural convection [48], thin elastic plate bending problems [49], thermoelasticity problems [50], equal-width equation [51], Zakharov–Kuznetsov-Modified equal-width equation [52], and hyperbolic PDEs [53], etc. The aforementioned advancements demonstrate that ST-GFDM is a promising meshless method with potential for engineering applications. In this study, the ST-GFDM is utilized in combination with the two-step Newton's method and the time-marching approach to tackle the problem of solving dispersive shallow water waves.

After providing the motivation for the research. The governing equations of the dispersive shallow water wave are represented in Section 2. In the section on proposed ST-GFDM, the numerical process is described, and four numerical examples are simulated to verify the proposed method in Section 4. Then, the conclusions are represented in the final section. All abbreviations and descriptions in this paper are listed in Table 1 for quick and easy reference for the reader.

Table 1. The abbreviations and descriptions in this paper.

Abbreviations	Description
1st	first
1D	one-dimensional
2D	two-dimensional
2nd	second
3rd	third
ADM	Adomian's decomposition method
Eq	equation
FDM-RK4	finite difference method with the fourth-order Runge–Kutta method
GFDM	generalized finite difference method
HAM	homotopy analysis method
MAE	maximum absolute error
MVIA	modified variational iteration algorithm
PDE	partial differential equation
RBFCM	radial basis functions collocation method
RKHSM	reproducing kernel Hilbert space method
RMSE	root-mean-squared error
ST	space–time
ST-GFDM	space–time GFDM
VIM	He's variational iteration method
Superscript	
ext	exact solution
num	numerical solution

2. The Mathematical Models of the Dispersive Shallow Water Waves

The dispersive shallow water wave represents the behavior of the wave propagation, breaking wave, and dynamic system of the fluid. It is an important phenomenon in physics and engineering. In the past few decades, several researchers have proposed the

mathematical model of the dispersive shallow water wave problem, and its general form of the governing equation [1,2,11] is described as,

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + (\alpha + \beta u^{\omega})\frac{\partial u}{\partial x} = q\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3},\ a \le x \le b,\ t \ge 0, \tag{1}$$

where u is the wave speed; x and t are the spatial- and temporal-axes; α, β, ω, and q are the constants. The initial conditions and boundary conditions are represented as follows:

$$\begin{cases} u(x,0) = U_0(x) \\ u(a,t) = f_1(t) \\ u(b,t) = f_2(t) \end{cases}. \tag{2}$$

By applying the constants a, b, ω, and q, three different types of the dispersive shallow water wave equation are defined. The first one is the Fornberg–Whitham equation [5–8,11] that is given with $\alpha = 1$, $\beta = 1$, $\omega = 1$, and $q = 3$ as,

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + (1 + u)\frac{\partial u}{\partial x} = 3\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}, a \le x \le b,\ t \ge 0, \tag{3}$$

and its modified form [5,7,10] is given ($\alpha = 1$, $\beta = 1$, $k = 2$, and $q = 3$) as,

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + \left(1 + u^2\right)\frac{\partial u}{\partial x} = 3\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}. \tag{4}$$

The remaining two are similar: the Camassa–Holm equation [1,2] and the Degasperis–Procesi equation [2–4], respectively. By giving $\alpha = 0$, $\beta = 3$, $\omega = 1$, and $q = 2$, the Camassa–Holm equation can be denoted as

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + 3u\frac{\partial u}{\partial x} = 2\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}, \tag{5}$$

and the Degasperis–Procesi equation ($\alpha = 0$, $\beta = 4$, $\omega = 0$, and $q = 3$) is described as

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + 4u\frac{\partial u}{\partial x} = 3\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}, \tag{6}$$

As with the relationship between the Fornberg–Whitham equation and its modified form, the modified forms of the Camassa–Holm equation [2] and the Degasperis–Procesi equation [2–4] are represented by changing the nonlinear term $u\frac{\partial u}{\partial x}$ into $u^2\frac{\partial u}{\partial x}$. Then, they can be described as follows:

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + 3u^2\frac{\partial u}{\partial x} = 2\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}, \tag{7}$$

and

$$\frac{\partial u}{\partial t} - \frac{\partial^3 u}{\partial x^2 \partial t} + 4u^2\frac{\partial u}{\partial x} = 3\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + u\frac{\partial^3 u}{\partial x^3}, \tag{8}$$

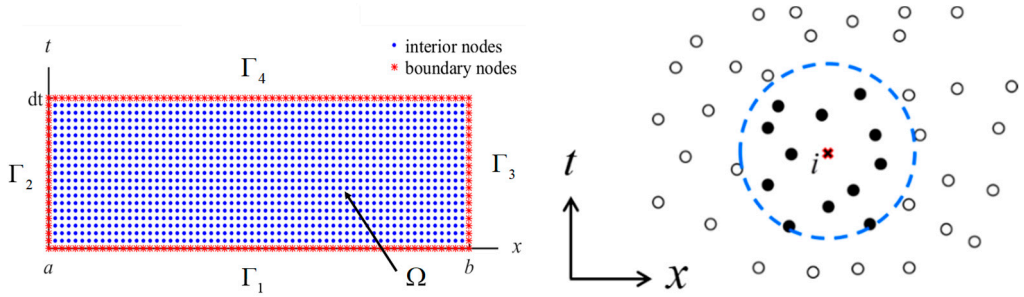
Equations (1) and (3)–(7) are highly nonlinear PDEs and have 3rd-order mixed partial derivatives as $\frac{\partial^3 u}{\partial x^2 \partial t}$. The complexity of dispersive shallow water wave equations has posed challenges to solving them using original explicit or implicit numerical schemes. As a result, researchers have primarily focused on proposing approximate or exact solutions for these equations. Consequently, applying numerical schemes to such equations is still in its early stages. In this study, the proposed ST-GFDM scheme is employed to solve the Fornberg–Whitham equation and its modified form, the modified Camassa–Holm equation, and the modified Degasperis–Procesi equation. The next section provides an introduction to the numerical procedures involved in the proposed ST-GFDM scheme.

3. The Proposed Meshless Numerical Scheme

The numerical procedures of the proposed ST-GFDM scheme are described in this section. The proposed numerical scheme uses ST-GFDM for the numerical discretization to obtain the nonlinear algebraic equations. Then, the nonlinear algebraic system is solved by the two-step Newton's method. A simple technique called the time-marching method is applied to save computational resources and solve numerical problems within an unpredictable end-time.

3.1. Space–Time Generalized Finite Difference Method

In this subsection, the numerical process of ST-GFDM is described. Once the time-dependent problem is determined, the distribution nodes can be set in the corresponding computational domain. For the one-dimensional (1D) time-dependent problem, the ST approach defines a two-dimensional (2D) steady-state problem in an *x-t* field. Since the ST approach is applied, the distribution nodes are set both in the space- and time-axes and named ST-domain [38,40,41], as shown in Figure 1a.



(**a**) (**b**)

Figure 1. The schematic diagram of (**a**) nodes and (**b**) a supporting domain of the GFDM in the *x-t* domain.

A supporting domain is set up by choosing the n_s nearest nodes within the central i node in Figure 1b (see the "×" symbol), and (x_i, t_i), $i = 1, 2, \ldots, N_T$ is defined as the coordinate of the ith node. On the other hand, $(x_{i,j}, t_{i,j})$, $j = 1, 2, \ldots, n_s$ represent the coordinates of the nodes in the supporting domain. By applying the n-order Taylor expansion and the weighting function, a residual function $B_n(u_i)$ can be written as [36,51],

$$B_n(u_i) = \sum_{j=1}^{n_s} \left[\left(-u_{i,j} + \sum_{0 \le o,m \le n,\ o+m \le n} \frac{1}{o!m!} \cdot \frac{\partial^{o+m} u_i}{\partial^o x \partial^m t} \cdot h_{i,j}^o \cdot l_{i,j}^m + E_n \right) w(h_{i,j}, l_{i,j}) \right]^2, \quad (9)$$

where u_i and $u_{i,j}$ denotes the unknowns at the ith node and unknowns inside the ith supporting domain, respectively; E_n is the truncation error of the n-order Taylor expansion; o and m is the natural number; $h_{i,j} = x_i - x_{i,j}$; $l_{i,j} = t_i - t_{i,j}$; and w is the quartic spline function and describes as follows:

$$w(h_{i,j}, l_{i,j}) = \begin{cases} 1 - 6\left(\frac{d_{i,j}}{dm_i}\right)^2 + 8\left(\frac{d_{i,j}}{dm_i}\right)^3 - 3\left(\frac{d_{i,j}}{dm_i}\right)^4, & d_{i,j} \le dm_i \\ 0 & , d_{i,j} > dm_i \end{cases}, \quad (10)$$

where $d_{i,j}$ is the distance between the ith node and each node inside the ith supporting domain, and $d_{i,j} = \sqrt{h_{i,j}^2 + l_{i,j}^2}$; and dm_i is the maximum distance.

In this study, the 3rd-order ST-GFDM was applied, and Equation (9) can be written as,

$$B_3(u_i) = \sum_{j=1}^{n_s} \left[\left(\begin{array}{c} u_i - u_{i,j} + h_{i,j} \left.\frac{\partial u}{\partial x}\right|_i + l_{i,j} \left.\frac{\partial u}{\partial t}\right|_i + \left.\frac{h_{i,j}^2}{2} \frac{\partial^2 u}{\partial x^2}\right|_i + h_{i,j} l_{i,j} \left.\frac{\partial^2 u}{\partial x \partial t}\right|_i + \left.\frac{l_{i,j}^2}{2} \frac{\partial^2 u}{\partial t^2}\right|_i + \cdots \\ \left.\frac{h_{i,j}^3}{6} \frac{\partial^3 u}{\partial x^3}\right|_i + \left.\frac{h_{i,j}^2 l_{i,j}}{2} \frac{\partial^3 u}{\partial x^2 \partial t}\right|_i + \left.\frac{h_{i,j} l_{i,j}^2}{2} \frac{\partial^3 u}{\partial x \partial t^2}\right|_i + \left.\frac{l_{i,j}^3}{6} \frac{\partial^3 u}{\partial t^3}\right|_i + E_3 \end{array} \right) w_{i,j} \right]^2, \tag{11}$$

The order of approximations in Equation (11), which accounts for the truncation errors, is determined by the Taylor series expansion up to the 3rd order. By minimalizing $B_3(u_i)$ with respect to,

$$\mathbf{D}_{u_i} = \left[\left.\frac{\partial u}{\partial x}\right|_i, \left.\frac{\partial u}{\partial t}\right|_i, \left.\frac{\partial^2 u}{\partial x^2}\right|_i, \left.\frac{\partial^2 u}{\partial x \partial t}\right|_i, \left.\frac{\partial^2 u}{\partial t^2}\right|_i, \left.\frac{\partial^3 u}{\partial x^3}\right|_i, \left.\frac{\partial^3 u}{\partial x^2 \partial t}\right|_i, \left.\frac{\partial^3 u}{\partial x \partial t^2}\right|_i, \left.\frac{\partial^3 u}{\partial t^3}\right|_i \right]^{\mathrm{T}}, \tag{12}$$

a linear system $\mathbf{A}\mathbf{D}_{u_i} = \mathbf{b}\mathbf{U}_i$ is obtained. In this linear system, the matrix $\mathbf{A}$ is denoted as $\mathbf{A} = \mathbf{P}^{\mathrm{T}}\mathbf{W}^2\mathbf{P}$, where

$$\mathbf{P} = \begin{bmatrix} h_{i,1} & l_{i,1} & \frac{h_{i,1}^2}{2} & h_{i,1}l_{i,1} & \frac{l_{i,1}^2}{2} & \frac{h_{i,1}^3}{6} & \frac{h_{i,1}^2 l_{i,1}}{2} & \frac{h_{i,1} l_{i,1}^2}{2} & \frac{l_{i,1}^3}{6} \\ h_{i,2} & l_{i,2} & \frac{h_{i,2}^2}{2} & h_{i,2}l_{i,2} & \frac{l_{i,2}^2}{2} & \frac{h_{i,2}^3}{6} & \frac{h_{i,2}^2 l_{i,2}}{2} & \frac{h_{i,2} l_{i,2}^2}{2} & \frac{l_{i,2}^3}{6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{i,j} & l_{i,j} & \frac{h_{i,j}^2}{2} & h_{i,j}l_{i,j} & \frac{l_{i,j}^2}{2} & \frac{h_{i,j}^3}{6} & \frac{h_{i,j}^2 l_{i,j}}{2} & \frac{h_{i,j} l_{i,j}^2}{2} & \frac{l_{i,j}^3}{6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{i,n_s} & l_{i,n_s} & \frac{h_{i,n_s}^2}{2} & h_{i,n_s}l_{i,n_s} & \frac{l_{i,n_s}^2}{2} & \frac{h_{i,n_s}^3}{6} & \frac{h_{i,n_s}^2 l_{i,n_s}}{2} & \frac{h_{i,n_s} l_{i,n_s}^2}{2} & \frac{l_{i,n_s}^3}{6} \end{bmatrix}_{n_s \times 9}, \tag{13}$$

$\mathbf{W} = diag(w_{i,1}, w_{i,2}, \cdots, w_{i,n_s})$; $\mathbf{b}$ is a coefficient matrix whose size is $9 \times (n_s + 1)$ and is written as $\mathbf{b} = \begin{bmatrix} \mathbf{P}^{\mathrm{T}}\mathbf{w} & \mathbf{P}^{\mathrm{T}}\mathbf{W}^2 \end{bmatrix}$; $\mathbf{U}_i = [u_i, u_{i,1}, u_{i,2}, \ldots, u_{i,n_s}]^{\mathrm{T}}$; and the superscript T is a symbol of transpose. By solving this linear system, the approximate expression of $\mathbf{D}_{u_i}$ can be written as,

$$\mathbf{D}_{u_i} = \mathbf{A}^{-1}\mathbf{b}\mathbf{U}_i = \mathbf{E}_i\mathbf{U}_i, \tag{14}$$

where $\mathbf{E}_i$ is the matrix of weighting coefficients within sized $9 \times (n_s + 1)$. Then, the derivatives are denoted as follows:

$$\begin{cases} \left.\frac{\partial u}{\partial x}\right|_i = wx_{i,0}u_i + \sum_{j=1}^{n_s} wx_{i,j}u_{i,j} \\ \left.\frac{\partial u}{\partial t}\right|_i = wt_{i,0}u_i + \sum_{j=1}^{n_s} wt_{i,j}u_{i,j} \\ \left.\frac{\partial^2 u}{\partial x^2}\right|_i = wxx_{i,0}u_i + \sum_{j=1}^{n_s} wxx_{i,j}u_{i,j} \\ \left.\frac{\partial^2 u}{\partial x \partial t}\right|_i = wxt_{i,0}u_i + \sum_{j=1}^{n_s} wxt_{i,j}u_{i,j} \\ \left.\frac{\partial^2 u}{\partial t^2}\right|_i = wtt_{i,0}u_i + \sum_{j=1}^{n_s} wtt_{i,j}u_{i,j} \\ \left.\frac{\partial^3 u}{\partial x^3}\right|_i = wxxx_{i,0}u_i + \sum_{j=1}^{n_s} wxxx_{i,j}u_{i,j} \\ \left.\frac{\partial^3 u}{\partial x^2 \partial t}\right|_i = wxxt_{i,0}u_i + \sum_{j=1}^{n_s} wxxt_{i,j}u_{i,j} \\ \left.\frac{\partial^3 u}{\partial x \partial t^2}\right|_i = wxtt_{i,0}u_i + \sum_{j=1}^{n_s} wxtt_{i,j}u_{i,j} \\ \left.\frac{\partial^3 u}{\partial t^3}\right|_i = wttt_{i,0}u_i + \sum_{j=1}^{n_s} wttt_{i,j}u_{i,j} \end{cases}, \tag{15}$$

where $\{wx_{i,j}\}_{j=0}^{n_s}$, $\{wt_{i,j}\}_{j=0}^{n_s}$, $\{wxx_{i,j}\}_{j=0}^{n_s}$, $\{wxt_{i,j}\}_{j=0}^{n_s}$, $\{wtt_{i,j}\}_{j=0}^{n_s}$, $\{wxxx_{i,j}\}_{j=0}^{n_s}$, $\{wxxt_{i,j}\}_{j=0}^{n_s}$, $\{wxtt_{i,j}\}_{j=0}^{n_s}$, and $\{wttt_{i,j}\}_{j=0}^{n_s}$ are the weighting coefficients, and obtained by solving Equation (14). Equation (15) is a significant part of the proposed ST-GFDM in that the partial derivatives are approximated by a linear combination of the weight coefficients and function values in a local support domain. Thus, the discretized governing equations and boundary conditions only relate to the central i node and its nearby n_s nodes in the local support domain. In addition, the numerical process of Equations (9)–(15) will apply for each node to obtain the weighting coefficients matrix $\mathbf{E}_i$ of each node. Finally, the resultant matrix system, formed by the algebraic equations after discretization, is a sparse matrix.

The following contexts demonstrate the numerical discretization for the governing equation and boundary conditions to form the nonlinear algebraic systems. Since the simulated start, N_{b1}, N_{b2}, N_{b3}, and N_{b4} nodes have been set up along the Γ_1, Γ_2, Γ_3 and Γ_4, respectively. Meanwhile, N_i nodes are distributed inside the computational domain Ω. (see Figure 1a). The nodes along Γ_1 are satisfied with the initial conditions $U_0(x)$, and the nodes along Γ_2 and Γ_3 are satisfied with the boundary conditions $f_1(a,t)$ and $f_2(b,t)$, respectively. Therefore, the algebraic equations along the Γ_1, Γ_2 and Γ_3 are described as,

$$\begin{cases} F_i = u_i - U_0(x_i),\ i = 1, 2, \ldots, N_{b1}, \\ F_i = u_i - f_2(a,t_i),\ i = N_{b1}+1, N_{b1}+2, \ldots, N_{b1}+N_{b2}, \\ F_i = u_i - f_1(b,t_i),\ i = N_{b1}+N_{b2}+1, N_{b1}+N_{b2}+2, \ldots, N_{b1}+N_{b2}+N_{b3}, \end{cases} \tag{16}$$

The remaining domains, Γ_4 and Ω, are both satisfied by the governing equation, Equation (1), and they can be discretized by Equation (15) as follows:

$$F_i = \left\{ \begin{array}{l} \left(wt_{i,0}u_i + \sum\limits_{j=1}^{n_s} wt_{i,j}u_{i,j} \right) - \left(wxxt_{i,0}u_i + \sum\limits_{j=1}^{n_s} wxxt_{i,j}u_{i,j} \right) + \ldots \\ \left(\alpha + \beta u_i^{\omega}\right) \left(wx_{i,0}u_i + \sum\limits_{j=1}^{n_s} wx_{i,j}u_{i,j} \right) - \ldots \\ q \left(wx_{i,0}u_i + \sum\limits_{j=1}^{n_s} wx_{i,j}u_{i,j} \right) \left(wxx_{i,0}u_i + \sum\limits_{j=1}^{n_s} wxx_{i,j}u_{i,j} \right) - \ldots \\ u_i \left(wxxx_{i,0}u_i + \sum\limits_{j=1}^{n_s} wxxx_{i,j}u_{i,j} \right) \end{array} \right\}, \tag{17}$$
$$i = N_{b1}+N_{b2}+N_{b3}+1, N_{b1}+N_{b2}+N_{b3}+2, \ldots, N_B, N_B+1, N_B+2, \ldots, N_T,$$

where N_B is a number of nodes set up on the whole boundary, and defined as $N_B = N_{b1} + N_{b2} + N_{b3} + N_{b4}$; N_T is the number of total nodes in the computational domain, and defined as $N_T = N_B + N_i$. In Equations (16) and (17), N_T algebraic equations are acquired and denoted as $\mathbf{F} = \left[F_1, F_2, \ldots, F_i\right]^{\mathrm{T}}$, $i = 1, 2, \ldots, N_T$. Meanwhile, N_T unknown values are defined as $\mathbf{U} = \left[u_1, u_2, \ldots, u_i\right]^{\mathrm{T}}$, $i = 1, 2, \ldots, N_T$. Then, the two-step Newton's method is applied to solve this nonlinear algebraic system, and the specific numerical process is described in the next subsection.

3.2. Two-Step Newton's Method

In this research, the famous nonlinear solver, Newton's method, was applied for iterating the numerical solution of the proposed ST-GFDM. Newton's method is a classic solver for nonlinear systems based on the first-order Taylor expansion formula. Newton's method possesses characteristics that make it easy to program and ensure fast convergence. The iteration formula for Newton's method is written as follows:

$$\mathbf{U}^{k+1} = \mathbf{U}^k - \left(\mathbf{J}^{-1}\right)^k \mathbf{F}^k, k = 1, 2, 3, \ldots, \tag{18}$$

where **J** is the Jacobian matrix, and the elements are expressed as $J_{i,j} = \partial F_i / \partial U_j$; k is the number of iterations; $\mathbf{U}^{k+1}$ and $\mathbf{U}^k$ are the vectors of unknowns at the (k +1)-th and k-th iterations, respectively. Although Newton's method is simple to apply, the inverse matrix of **J** is a troublesome issue during the simulation. To avoid the computation of the inverse matrix, the two-step iteration of Newton's method is given [46,48,51] as follows:

$$\mathbf{J}^k \Delta \mathbf{U}^k = -\mathbf{F}^k, \tag{19}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \Delta \mathbf{U}^k, \tag{20}$$

where $\Delta \mathbf{U}^k$ is the numerical increment at the kth-step iteration.

The two-step process described in Equations (19) and (20) is employed as a means to avoid computing the inverse of the Jacobian matrix, and the $\Delta \mathbf{U}^k$ can be solved as a linear system. Thus, the nonlinear algebraic system obtained by the ST-GFDM can be solved efficiently. Furthermore, the iterative system will converge once the convergence condition $\delta = Max\left|\Delta \mathbf{U}^k\right| \leq 10^{-9}$ is reached and the numerical solutions of the proposed ST-GFDM are obtained.

3.3. Time-Marching Approach

In the previous subsections, the numerical process of the proposed ST-GFDM scheme has actually been introduced, but those processes only work in a single ST domain, $\Omega = [a, b] \times [0, \mathrm{dt}]$. To deal with numerical problems with unpredictable end-times or long temporal scales, a simple technique, the time-marching approach, is used in the proposed numerical scheme. The schematics of the time-marching approach are demonstrated in Figure 2. In Figure 2a, a small-scale ST domain, D1 $= [a, b] \times [0, \mathrm{dt}]$, is formed when the simulation starts and the distribution of nodes is set up like in Figure 1a. In this ST domain, the initial conditions $U_0(x)$ are satisfied along Γ_1^1, Γ_2^1, and Γ_3^1, which satisfy the boundary conditions $f_1(a, t)$ and $f_2(b, t)$, respectively. The governing equation (Equation (1)) is satisfied on Γ_4^1 and Ω. Then, the numerical processes in Sections 3.1 and 3.3 are applied to obtain the numerical solutions in the D1 domain. Once the numerical procedures within D1 are completed, the D1 will be shifted by distance dt along the t-axis, and the 2nd small-scale ST domain D2 $= [a, b] \times [\mathrm{dt}, 2\mathrm{dt}]$ is formed as shown in Figure 2b.

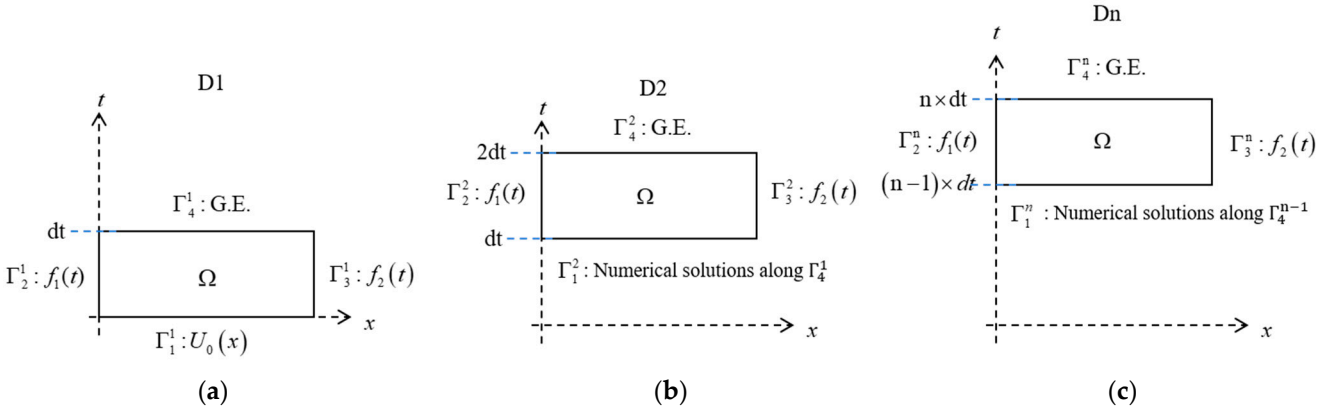


Figure 2. The schematic diagrams of the time-marching approach in the proposed ST-GFDM. (**a**) 1st domain, (**b**) 2nd domain, and (**c**) n-th domain.

The numerical processes within D2 are the same as the D1, but the boundary conditions along Γ_1^2 are different. In Γ_1^2, the numerical solutions at Γ_4^1 in D1 are treated as the initial conditions for D2. Thus, the numerical solutions in D2 can be solved by the proposed ST-GFDM and the two-step Newton's method. It should be noted that the coordinates between D1 and D2 are relative locations. Thus, the weighting coefficients of the proposed ST-GFDM in D1 are the same as those of each ST domain generated by the

time-marching approach. After moving the ST domain within n times, the n-th ST domain, $\mathrm{Dn} = [a, b] \times [(\mathrm{n}-1)\mathrm{dt}, \mathrm{n} \times \mathrm{dt}]$, is generated (see Figure 2c), and the numerical solutions are solved. The simulation will finish when $\mathrm{n} \times \mathrm{dt}$ reaches the specified end-time determined by the numerical example. To summarize the above numerical process, the flowchart is shown in Figure 3. In the next section, four numerical examples are provided to verify the proposed ST-GFDM scheme.

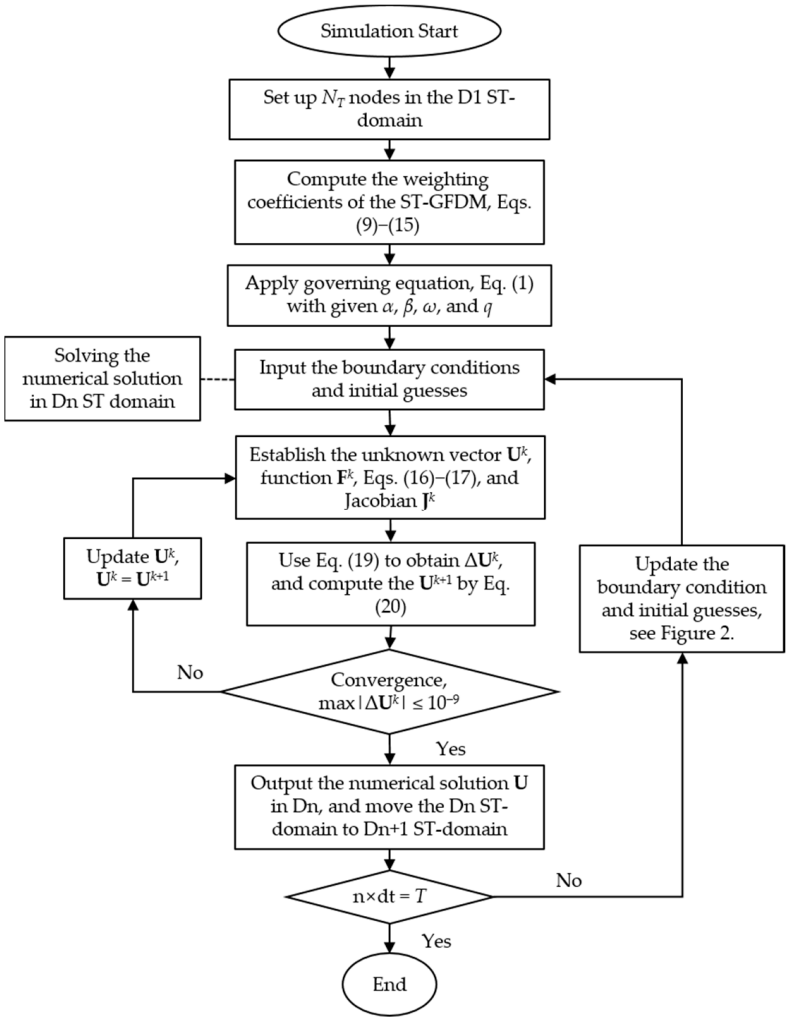


Figure 3. The flowchart of the numerical process of the proposed ST-GFDM scheme.

4. Numerical Examples

In this section, the proposed ST-GFDM is validated by testing four numerical examples. These examples include the Fornberg–Whitham equation and its modified form, the modified Camassa–Holm equation, and the modified Degasperis–Procesi equation. The following symbols are defined: T is the end time of the numerical problem, N_D is the number of times that the ST domain has been moved by the time-marching method, and is written as $N_D = T/\mathrm{dt}$. The distribution of nodes for each example is displayed in Figure 1a. To analyze the accuracy of the proposed ST-GFDM, the numerical errors are defined in this study as follows:

$$\begin{cases} L_{\infty,\mathrm{n}} = \max\limits_{1 \le i \le N_T} \left| u_i^{ext} - u_i^{num} \right|^{\mathrm{n}}, \\ L_\infty = \max\limits_{1 \le \mathrm{n} \le N_D} \{ L_{\infty,\mathrm{n}} \}, \end{cases} \tag{21}$$

$$\begin{cases} \text{RMSE}_{\text{n}} = \left| \sqrt{\sum_{i=1}^{N_T} \left(u_i^{ext} - u_i^{num}\right)^2 / N_T} \right|^{\text{n}}, \\ \text{RMSE} = \max_{1 \le \text{n} \le N_D} \{\text{RMSE}_{\text{n}}\}, \end{cases} \tag{22}$$

where superscript *ext* is the exact solution; *num* denotes numerical solutions. In Equation (21), $L_{\infty,\text{n}}$ is the maximum absolute errors (MAEs) in Dn, and L_∞ is the maximum MAEs in the entire simulation. Meanwhile, the root-mean-squared error (RMSE) in each ST domain is defined as RMSE_{n} in Equation (22), and RMSE is the maximum RMSE in the whole simulation. In addition, the initial guesses are the initial conditions of each numerical problem for iteration in D1, and the numerical solutions in Dn $-$ 1 are initial guesses of numerical processes in Dn.

4.1. Example 1

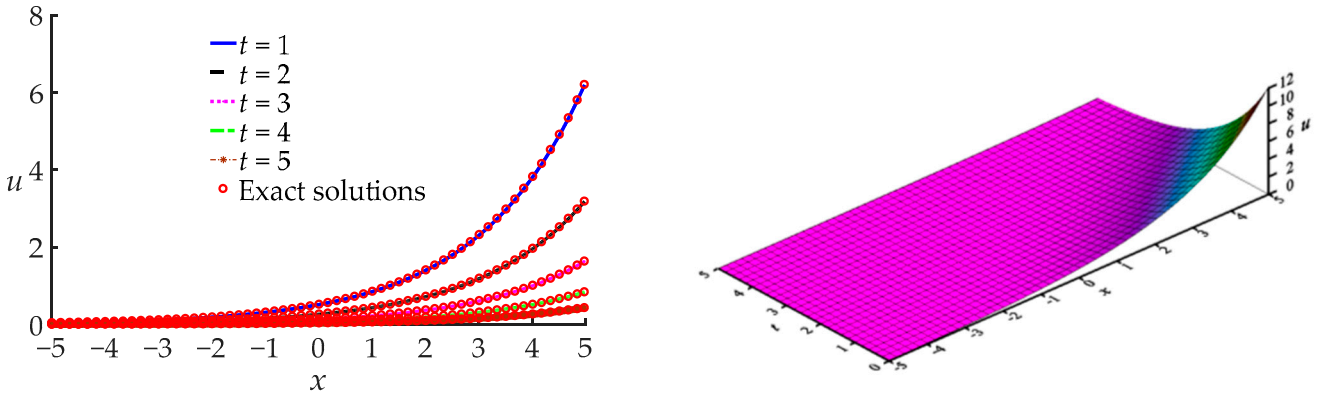
The Fornberg–Whitham equation [9,11] is solved in this subsection. This mathematical model is presented to describe the behavior of the breaking wave and has solitary or traveling wave solutions. The exact solution of the first numerical example is described as,

$$u^{ext}(x,t) = e^{\left(\frac{x}{2} - \frac{2t}{3}\right)}, \tag{23}$$

and the initial condition and boundary conditions are represented as,

$$U_0(x) = e^{\frac{x}{2}},\ f_1(t) = e^{\left(\frac{a}{2} - \frac{2t}{3}\right)},\ f_2(t) = e^{\left(\frac{b}{2} - \frac{2t}{3}\right)}. \tag{24}$$

The following parameters are applied in this numerical example: $a = -5$, $b = 5$, dt = 0.5, $T = 5$, and $N_D = 10$. In Figure 4, the numerical solutions are obtained by applying $N_T = 8421$ and $n_s = 23$. The numerical solutions at different time levels are compared with the exact solutions and demonstrated in Figure 4a. This figure shows that the numerical solutions agree well with the exact solutions. Figure 4b is a 3D surface plot of numerical solutions to observe the numerical behavior of the Fornberg–Whitham equation through time.



(**a**) (**b**)

Figure 4. (**a**) The comparisons of the u^{ext} and u^{num} at different t and (**b**) the surface plot of u^{num} in the entire x-t domain of Example 1.

Tables 2–4 are provided to verify the accuracy and parameter sensitivity of the proposed method. The MAEs and RMSEs are calculated using N_T and n_s at different time levels. In Table 2, the results for different time levels are displayed, showing the MAEs and RMSEs obtained for various N_T. These error values demonstrate that increasing N_T leads to more accurate numerical solutions. Meanwhile, the convergence rate and CPU times are shown in Table 2. Since the proposed numerical scheme is a 3rd-order ST-GFDM combined with

Newton's method, according to the theoretical convergence rates of these two methods, the theoretical convergence rates of the proposed numerical scheme should be 2nd-order convergence. From Table 2, the numerical performance is between 1.5–2. On the other hand, the CPU times are provided in Table 2 as well. Table 2 reveals that simulations are completed within 4 s when N_T is below 9000, and the errors table indicates that satisfactory numerical solutions can already be achieved at this N_T. With an end time of 5 for the first example, the proposed numerical scheme can successfully complete the simulation within the problem's specified end time using approximately 9000 nodes. Despite the increase in CPU time as N_T increases, the CPU time remains below 10 s, demonstrating the efficiency of the proposed ST-GFDM in solving problems with long-time scales.

Table 2. Numerical errors of Example 1 with n_s = 20 using different N_T at different t, convergence rates, and CPU-time.

	t = 1 (n = 2)		t = 3 (n = 6)		t = 5 (n = 10)		Convergence Rates ($L_{\infty,10}$ Are Used)	CPU-Time
N_T	$L_{\infty,2}$	$\mathbf{RMSE_2}$	$L_{\infty,6}$	$\mathbf{RMSE_6}$	$L_{\infty,10}$	$\mathbf{RMSE_{10}}$		
2211	1.725×10^{-4}	2.424×10^{-5}	2.251×10^{-5}	4.081×10^{-6}	3.250×10^{-5}	5.891×10^{-6}	-	0.7829 s
4215	7.653×10^{-5}	8.978×10^{-6}	1.206×10^{-5}	1.780×10^{-6}	1.703×10^{-5}	2.693×10^{-6}	1.918	1.5629 s
8421	3.267×10^{-5}	3.071×10^{-6}	6.677×10^{-6}	8.855×10^{-7}	9.153×10^{-6}	1.214×10^{-6}	1.742	3.5874 s
12,025	2.147×10^{-5}	1.773×10^{-6}	5.152×10^{-6}	6.536×10^{-7}	6.891×10^{-6}	8.334×10^{-7}	1.543	5.9342 s
18,631	1.313×10^{-5}	9.162×10^{-7}	3.856×10^{-6}	4.747×10^{-7}	4.927×10^{-6}	5.949×10^{-7}	1.528	10.6508 s

Table 3. MAEs and RMSEs of Example 1 with N_T = 8421 by using different n_s at different t.

	t = 1 (n = 2)		t = 3 (n = 6)		t = 5 (n = 10)	
n_s	$L_{\infty,2}$	$\mathbf{RMSE_2}$	$L_{\infty,6}$	$\mathbf{RMSE_6}$	$L_{\infty,10}$	$\mathbf{RMSE_{10}}$
20	3.267×10^{-5}	3.071×10^{-6}	6.677×10^{-6}	8.855×10^{-7}	9.153×10^{-6}	1.214×10^{-6}
23	5.158×10^{-5}	5.737×10^{-6}	1.961×10^{-5}	2.011×10^{-6}	1.301×10^{-5}	1.563×10^{-6}
26	5.093×10^{-5}	5.713×10^{-6}	3.067×10^{-5}	3.064×10^{-6}	1.264×10^{-5}	1.566×10^{-6}

Table 4. Comparisons of the numerical results of Example 1 with exact solutions at various locations at different time levels (N_T = 8421 and n_s = 23).

		t = 1 (n = 2)		t = 3 (n = 6)		t = 5 (n = 10)	
		u^{ext}	ST-GFDM	u^{ext}	ST-GFDM	u^{ext}	ST-GFDM
x	−4	0.06948345	0.06948238	0.01831564	0.01831622	0.00482795	0.00483149
	−2	0.18887560	0.18887409	0.04978707	0.04978139	0.01312373	0.01312198
	0	0.51341712	0.51342701	0.13533528	0.13532764	0.03567399	0.03566356
	2	1.39561243	1.39565330	0.36787944	0.36788606	0.09697197	0.09696040
	4	3.79366789	3.79371117	1.00000000	1.00001868	0.26359714	0.26359633
$L_{\infty,n}$		5.158×10^{-5}		1.961×10^{-5}		1.301×10^{-5}	

In Table 3, MAEs and RMSEs are also presented using different n_s. The errors indicate that the parameter n_s exhibits low sensitivity. Additionally, both tables demonstrate no accumulation of errors throughout the numerical simulation. This observation supports the effectiveness of the ST approach in mitigating the accumulation of numerical errors, a concern that hybrid numerical schemes often encounter when applied to time-dependent problems.

In order to further validate the capability of the proposed numerical scheme, the numerical solutions located at specific positions (x,t) are presented in Table 4 and compared with the exact solutions. In this table, the function values are taken to eight decimal places to show the precision accuracy of the proposed ST-GFDM. Additionally, the proposed ST-GFDM is compared with the HAM [10], ADM [7], RKHSM [7], MVIA-I [5], and the 2nd-order finite difference method combined with the fourth-order Runge–Kutta method

(FDM-RK4). Those comparisons are shown in Table 5. In these comparative data, it can be seen that the proposed ST-GFDM performs better than the HAM and ADM. Although the accuracy is weaker than the RKHSM and MVIA-I, solutions with numerical errors below 10^{-6} or even 10^{-7} can be obtained, which is already an acceptable numerical result for engineering applications.

Table 5. Comparisons of the numerical results of Example 1 with the exact solution and the other approximation methods at various locations at $t = 5$ (n = 10, $N_T = 8421$, and $n_s = 23$).

x	u^{ext}	HAM [10]	ADM [7]	RKHSM [7]	MVIA-I [5]	FDM-RK4	ST-GFDM	MAEs
−4	0.00482795	0.00487526	0.00317192	0.00482795	0.00482795	0.0048810	0.00482958	1.626×10^{-6}
−2	0.01312373	0.01325233	0.00862217	0.01312373	0.01312373	0.0127930	0.01312294	7.893×10^{-7}
0	0.03567399	0.03602358	0.02343750	0.03567400	0.03567399	0.0349080	0.03566866	5.337×10^{-6}
2	0.09697197	0.09792224	0.06370972	0.09697195	0.09697197	0.1008894	0.09696298	8.992×10^{-6}
4	0.26359714	0.26618027	0.17318095	0.26359729	0.26359714	0.3172236	0.26359021	6.925×10^{-6}

To compare with the original numerical scheme for time-dependent problems, the FDM-RK4 is applied to obtain the numerical results within $\Delta x = 0.5$ and $\Delta t = 0.0001$. The numerical solutions of the FDM-RK4 show that the results have a maximum absolute error of the order of 10^{-1}. In our test, the smaller Δx needs a smaller Δt for FDM-RK4, and the simulation will diverge during the simulation because the end time could not reach $t = 5$. In Figure 5, it can be clearly seen that the numerical error of ST-GFDM is maintained at the same magnitude during the simulation, and the numerical results are more accurate than those of FDM-RK4. This numerical performance shows that using the space–time approach, the numerical errors will not accumulate during simulation for each ST domain. The above discussions show that the proposed ST-GFDM can accurately and stably solve the Fornberg–Whitham equation. Since the capability of the proposed ST-GFDM has been verified in this numerical example, the following numerical tests are solved using specific parameters, and the obtained numerical results are compared with other approximation algorithms.

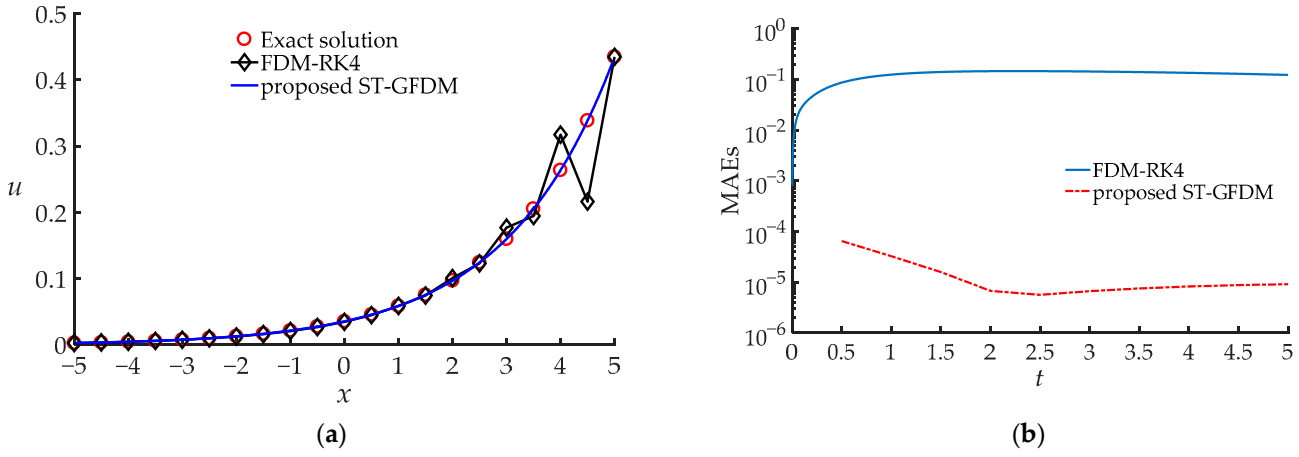


Figure 5. The comparisons of the FDM-RK4 and proposed ST-GFDM. (**a**) The numerical solution at $t = 5$ and (**b**) the MAEs curves in the entire simulation of Example 1.

4.2. Example 2

The second example is a numerical test for the modified Fornberg–Whitham equation [3,5–7,9]. The exact solution is given as follows:

$$u^{ext}(x,t) = \frac{3}{4}\left(\sqrt{15}-5\right)\mathrm{sech}^2\left(c\left(x-\left(5-\sqrt{15}\right)t\right)\right), \tag{25}$$

where $c = \frac{1}{20}\sqrt{10\left(5-\sqrt{15}\right)}$. The initial condition and boundary conditions are denoted as follows:

$$\begin{cases} U_0(x) = \frac{3}{4}\left(\sqrt{15}-5\right)\mathrm{sech}^2(cx), \\ f_1(t) = \frac{3}{4}\left(\sqrt{15}-5\right)\mathrm{sech}^2\left(c\left(a-\left(5-\sqrt{15}\right)t\right)\right), \\ f_2(t) = \frac{3}{4}\left(\sqrt{15}-5\right)\mathrm{sech}^2\left(c\left(b-\left(5-\sqrt{15}\right)t\right)\right). \end{cases} \tag{26}$$

The following parameters are used in this numerical example: $a = -10$, $b = 10$, dt = 0.2, $T = 1$, $N_D = 10$, $N_T = 40{,}421$, and $n_s = 20$. The numerical solutions are displayed in Figure 6. In Figure 6a, the comparison of the exact solutions and the numerical solutions at specific time levels (t = 0.2, 0.4, 0.8, and 1) is shown. These numerical results are in good agreement with the exact solutions. In addition, the MAEs at specific points (x,t) are listed in Table 6 with the errors of the MVIA [5] and VIM [8]. Obviously, the error performance of the proposed numerical scheme is better than that of the other two approximation algorithms. Also, it verifies that the proposed ST-GFDM is fully capable of handling such complex PDEs.

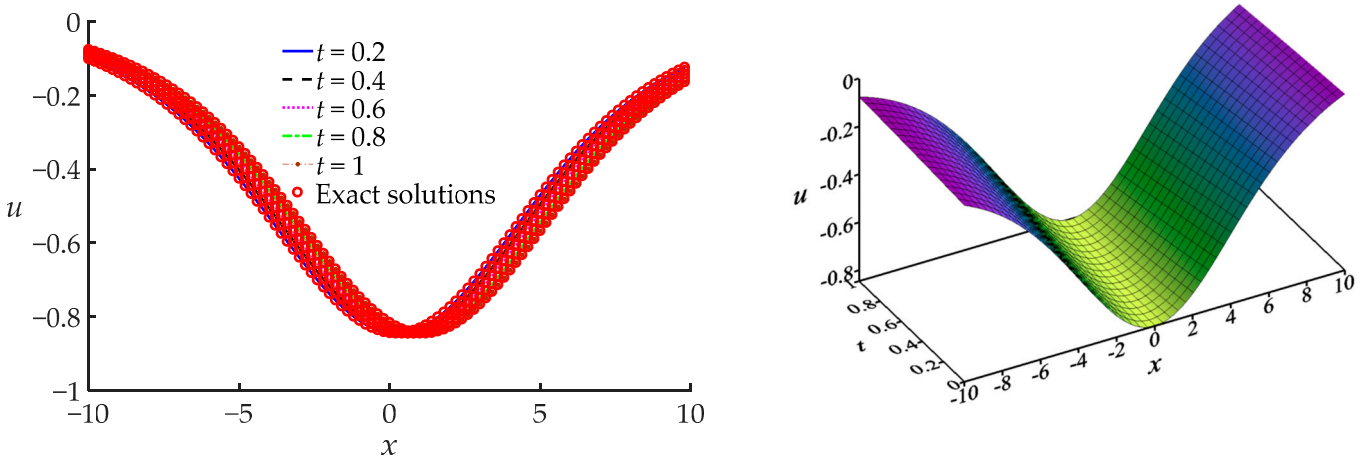


Figure 6. (**a**) The comparisons of the u^{ext} and u^{num} at different t and (**b**) the surface plot of u^{num} in the entire x-t domain of Example 2.

Table 6. MAEs of Example 2 with the exact solution and the other numerical methods.

	x = 2.5			x = 5		
t	MVIA [5]	VIM [8]	ST-GFDM	MVIA [5]	VIM [8]	ST-GFDM
0.02	3.780×10^{-5}	1.180×10^{-4}	1.883×10^{-7}	9.966×10^{-6}	2.124×10^{-5}	4.272×10^{-8}
0.04	7.240×10^{-5}	2.363×10^{-4}	1.949×10^{-7}	1.778×10^{-5}	4.797×10^{-5}	4.077×10^{-8}
0.06	1.036×10^{-4}	3.547×10^{-4}	1.996×10^{-7}	2.333×10^{-5}	8.029×10^{-5}	3.867×10^{-8}
0.08	1.313×10^{-4}	4.731×10^{-4}	2.042×10^{-7}	2.653×10^{-5}	1.183×10^{-4}	3.647×10^{-8}
0.1	1.552×10^{-4}	5.914×10^{-4}	2.087×10^{-7}	2.727×10^{-5}	1.622×10^{-4}	3.419×10^{-8}

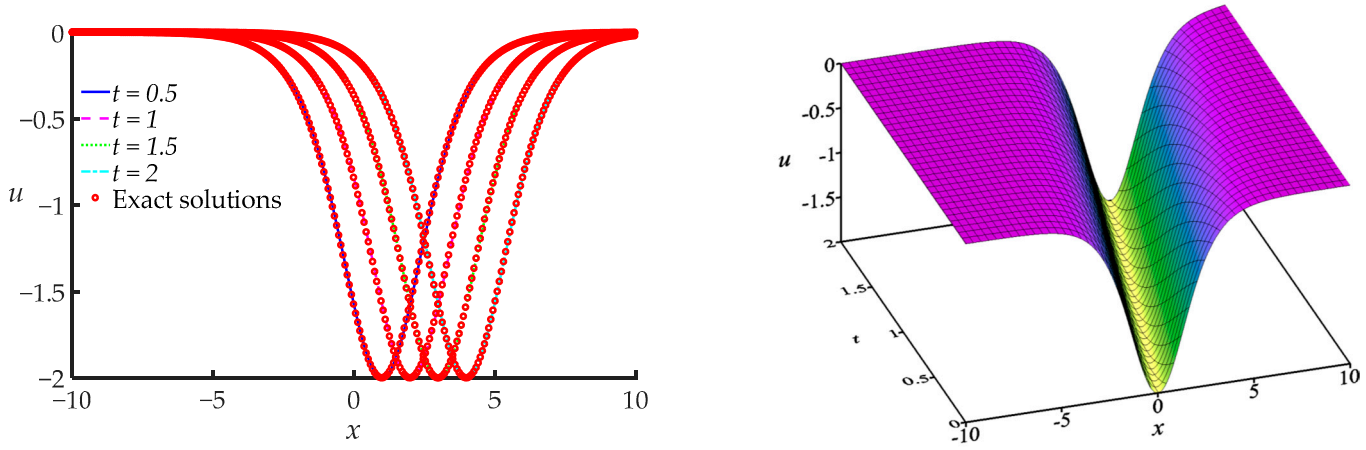
4.3. Example 3

The 3rd example is the modified Camassa–Holm equations [1,2]. The exact solution is given as follows:

$$u^{ext}(x,t) = -2\mathrm{sech}^2\left(\frac{x}{2}-t\right), \tag{27}$$

The following are applied in this numerical example: $a = -10$, $b = 10$, dt = 0.5, $T = 2$, $N_D = 4$, $N_T = 37{,}231$, and $n_s = 20$. The initial condition and boundary conditions are obtained by introducing the above parameters into Equation (27). The numerical solutions are illustrated in Figure 7. In these plots, the obtained numerical results are well compared

with the exact solution in Figure 7a. Moreover, Figure 7b demonstrates the physical behavior of the wave transmission. Additionally, Table 7 presents the numerical results at various locations, comparing them with the numerical results of the VIM [2]. The table shows that the proposed ST-GFDM successfully solves the Camassa–Holm equation and provides accurate numerical solutions.



(**a**) (**b**)

Figure 7. (**a**) The comparisons of the u^{ext} and u^{num} at different t and (**b**) the surface plot of u^{num} in the entire x-t domain of Example 3.

Table 7. Comparisons of the numerical results of Example 3 with the exact solution and the other numerical methods at various locations at t = 0.05 and t = 0.1.

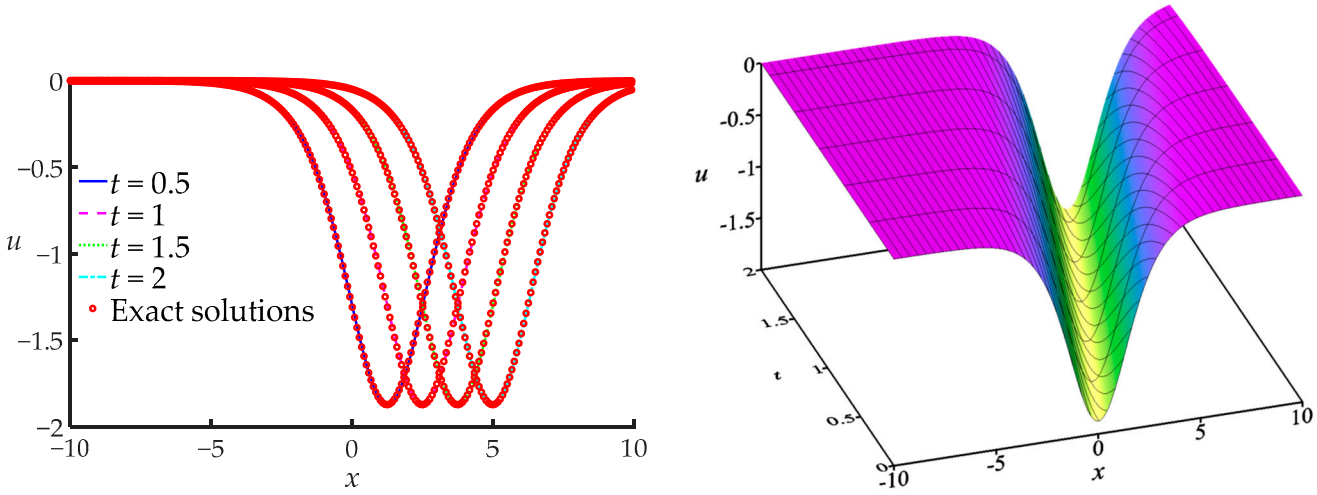
x	t	u^{ext}	ST-GFDM	VIM [2]	MAEs
6	0.05	−0.021795977	−0.021798486	−0.019790189	2.508360×10^{-6}
8	0.05	−0.002963750	−0.002960395	−0.002682964	3.355673×10^{-6}
6	0.1	−0.024074444	−0.024070611	−0.019848303	3.832791×10^{-6}
8	0.1	−0.003275195	−0.003265414	−0.002684058	9.781412×10^{-6}

4.4. Example 4

The final example is the modified Degasperis−Procesi equation [2,3]. The exact solution is given as follows:

$$u^{ext}(x,t) = -\frac{15}{8}\mathrm{sech}^2\left(\frac{x}{2} - \frac{5t}{4}\right), \tag{28}$$

The boundary condition and initial condition can be obtained from Equation (28). The following parameters are used: a = -10, b = 10, dt = 0.5, T = 2, N_D = 4, N_T = 37,231, and n_s = 20. The numerical solutions and data are given in Figure 8 and Table 8, respectively. Due to the similarity between the modified Degasperis−Procesi equation and the modified Camassa–Holm equation, the numerical solutions' behavior is also similar. These comparisons show that the proposed ST-GFDM consistently yields accurate numerical results, as demonstrated in the previous numerical examples. As described earlier, the proposed ST-GFDM effectively solves the shallow water wave problem and demonstrates stability in simulating problems over long time scales.



(**a**) (**b**)

Figure 8. (**a**) The comparisons of the u^{ext} and u^{num} at different t and (**b**) the surface plot of u^{num} in the entire x-t domain of Example 4.

Table 8. Comparisons of the numerical results of Example 4 with the exact solution and the other numerical methods at various locations at $t = 0.05$ and $t = 0.1$.

x	t	u^{ext}	ST-GFDM	VIM [2]	MAEs
6	0.05	−0.020951831	−0.020948113	−0.018566923	3.717910×10^{-6}
8	0.05	−0.002851900	−0.002848801	−0.002682964	3.099397×10^{-6}
6	0.1	−0.023714184	−0.023719627	−0.018635025	5.443036×10^{-6}
8	0.1	−0.003229560	−0.003227788	−0.002516809	1.771923×10^{-6}

5. Conclusions

This study uses the proposed ST-GFDM to numerically solve nonlinear shallow water wave equations, specifically the modified Camassa–Holm equation, the modified Degasperis–Procesi equation, the Fornberg–Whitham equation, and its modified form. The ST-GFDM discretizes the governing equations in the ST domain, eliminating the need for mesh generation and simplifying the numerical procedures. The resulting nonlinear algebraic system is solved using the two-step Newton's method. Adopting ST-GFDM makes the computational process more straightforward and avoids the complexities associated with mesh generation. One significant innovation of the ST-GFDM is treating the temporal derivative as a spatial derivative. The ST-GFDM enables the discretization of all partial derivatives and the efficient handling of mixed derivatives. Due to the Taylor series expansion and the moving least-squares method, the proposed meshless numerical scheme forms a nonlinear algebraic system, and this system has a sparse Jacobian matrix. Then, the two-step Newton–Raphson method is applied to efficiently solve this nonlinear algebraic system. To deal with long-time-scale numerical problems, the time-marching method is applied for moving the ST domain and saving computational resource, thereby addressing the shortcomings of the ST approach.

Four numerical examples are presented to verify the proposed ST-GFDM scheme. In the first numerical case, by varying different parameters, it is observed that increasing the N_T leads to more accurate numerical solutions. Additionally, adjusting the parameter n_s in a supporting domain demonstrates that n_s has low sensitivity. The numerical solutions have good agreements with the exact solutions and the solutions of other approximation methods in four examples. Furthermore, the accumulation of numerical errors during the time progression is insignificant. The proposed ST-GFDM scheme offers several advantages, including accurate treatment of transient problems, computational efficiency,

avoidance of mesh generation, and flexibility in handling complex PDEs. This observation highlights the scheme's capability to effectively solve transient problems while maintaining numerical accuracy. Future research can explore the application of the ST-GFDM scheme in computational fluid dynamics and tackle more challenging problem domains.

Author Contributions: Conceptualization, P.-W.L. and S.H.; methodology, P.-W.L.; software, S.H. and M.Z.; validation, S.H. and M.Z.; writing—original draft preparation, P.-W.L.; writing—review and editing, P.-W.L.; funding acquisition, P.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Shandong Province of China, grant number ZR2021QA012.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available from the corresponding author upon reasonable request.

Acknowledgments: The authors sincerely thank the Natural Science Foundation of Shandong Province, China, for providing financial support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Camassa, R.; Holm, D.D. An Integrable Shallow Water Equation with Peaked Solitons. *Phys. Rev. Lett.* **1993**, *71*, 1661–1664. [CrossRef]
2. Wazwaz, A.M. New Solitary Wave Solutions to the Modified Forms of Degasperis-Procesi and Camassa-Holm Equations. *Appl. Math. Comput.* **2007**, *186*, 130–141. [CrossRef]
3. Shaheen, S.; Haq, S.; Ghafoor, A. A Meshfree Technique for the Numerical Solutions of Nonlinear Fornberg–Whitham and Degasperis–Procesi Equations with Their Modified Forms. *Comput. Appl. Math.* **2022**, *41*, 1–22. [CrossRef]
4. Zhang, J.H.; Zheng, J.S.; Gao, Q.J. Numerical Solution of the Degasperis–Procesi Equation by the Cubic B-Spline Quasi-Interpolation Method. *Appl. Math. Comput.* **2018**, *324*, 218–227. [CrossRef]
5. Ahmad, H.; Seadawy, A.R.; Ganie, A.H.; Rashid, S.; Khan, T.A.; Abu-Zinadah, H. Approximate Numerical Solutions for the Nonlinear Dispersive Shallow Water Waves as the Fornberg–Whitham Model Equations. *Results Phys.* **2021**, *22*, 103907. [CrossRef]
6. Hörmann, G.; Okamoto, H. Weak Periodic Solutions and Numerical Case Studies of the Fornberg-Whitham Equation. *Discret. Contin. Dyn. Syst. Ser. A* **2019**, *39*, 4455–4469. [CrossRef]
7. Boutarfa, B.; Akgül, A.; Inc, M. New Approach for the Fornberg–Whitham Type Equations. *J. Comput. Appl. Math.* **2017**, *312*, 13–26. [CrossRef]
8. Lu, J. An Analytical Approach to the FornbergWhitham Type Equations by Using the Variational Iteration Method. *Comput. Math. Appl.* **2011**, *61*, 2010–2013. [CrossRef]
9. He, B.; Meng, Q.; Li, S. Explicit Peakon and Solitary Wave Solutions for the Modified Fornberg-Whitham Equation. *Appl. Math. Comput.* **2010**, *217*, 1976–1982. [CrossRef]
10. Abidi, F.; Omrani, K. The Homotopy Analysis Method for Solving the Fornberg-Whitham Equation and Comparison with Adomian's Decomposition Method. *Comput. Math. Appl.* **2010**, *59*, 2743–2750. [CrossRef]
11. Fornberg, B.; Whitham, G.B. A Numerical and Theoretical Study of Certain Nonlinear Wave Phenomena. *Philos. Trans. R. Soc. London. Ser. A Math. Phys. Sci.* **1978**, *289*, 373–404.
12. Haziot, S.V. Wave Breaking for the Fornberg–Whitham Equation. *J. Differ. Equ.* **2017**, *263*, 8178–8185. [CrossRef]
13. Segur, H.; Henderson, D.; Carter, J.; Hammack, J.; Li, C.M.; Pheiff, D.; Socha, K. Stabilizing the Benjamin-Feir Instability. *J. Fluid Mech.* **2005**, *539*, 229–271. [CrossRef]
14. Grimshaw, R.H.J.; Annenkov, S.Y. Water Wave Packets Over Variable Depth. *Stud. Appl. Math.* **2011**, *126*, 409–427. [CrossRef]
15. Rajan, G.K.; Henderson, D.M. The Linear Stability of a Wavetrain Propagating on Water of Variable Depth. *SIAM J. Appl. Math.* **2016**, *76*, 2030–2041. [CrossRef]
16. Maleewong, M.; Grimshaw, R.H.J. Amplification of Wave Groups in the Forced Nonlinear Schrödinger Equation. *Fluids* **2022**, *7*, 1–19. [CrossRef]
17. Rajan, G.K.; Bayram, S.; Henderson, D.M. Periodic Envelopes of Waves over Non-Uniform Depth. *Phys. Fluids* **2016**, *28*, 042106. [CrossRef]
18. Djidjeli, K.; Price, W.G.; Twizell, E.H.; Cao, Q. A Linearized Implicit Pseudo-Spectral Method for Some Model Equations: The Regularized Long Wave Equations. *Commun. Numer. Methods Eng.* **2003**, *19*, 847–863. [CrossRef]

19. Izadkhah, M.M.; Saberi-Nadjafi, J. Gegenbauer Spectral Method for Time-Fractional Convection-Diffusion Equations with Variable Coefficients. *Math. Methods Appl. Sci.* **2015**, *38*, 3183–3194. [CrossRef]
20. Ling, J.; Wang, H.; Mou, H. An Efficient Boundary-Type Meshless Computational Approach for the Axial Compression on the Part Boundary of the Circular Shaft (Brazilian Test). *Appl. Sci.* **2022**, *12*, 11806. [CrossRef]
21. Guo, K.; Li, S.; Zhong, Y.; Chen, R.; Wang, M.; Qiu, S.; Tian, W.; Su, G. Heat Transfer Mechanism Investigation of Bubble Growth on the Superhydrophilic Nano-Structured Surface Using Moving Particle Semi-Implicit Method. *Appl. Sci.* **2023**, *13*, 4114. [CrossRef]
22. Belinha, J.; Aires, M. Elastoplastic Analysis of Plates with Radial Point Interpolation Meshless Methods. *Appl. Sci.* **2022**, *12*, 12842. [CrossRef]
23. Cheng, S.; Wang, F.; Li, P.-W.; Qu, W. Singular Boundary Method for 2D and 3D Acoustic Design Sensitivity Analysis. *Comput. Math. Appl.* **2022**, *119*, 371–386. [CrossRef]
24. Yue, X.; Wang, F.; Li, P.-W.; Fan, C.-M. Local Non-Singular Knot Method for Large-Scale Computation of Acoustic Problems in Complicated Geometries. *Comput. Math. Appl.* **2021**, *84*, 128–143. [CrossRef]
25. Liu, S.; Li, P.W.; Fan, C.M.; Gu, Y. Localized Method of Fundamental Solutions for Two- and Three-Dimensional Transient Convection-Diffusion-Reaction Equations. *Eng. Anal. Bound. Elem.* **2021**, *124*, 237–244. [CrossRef]
26. Fu, Z.; Tang, Z.; Xi, Q.; Liu, Q.; Gu, Y.; Wang, F. Localized Collocation Schemes and Their Applications. *Acta Mech. Sin.* **2022**, *38*, 422167. [CrossRef]
27. Liszka, T.; Orkisz, J. The Finite Difference Method at Arbitrary Irregular Grids and Its Application in Applied Mechanics. *Comput. Struct.* **1980**, *11*, 83–95. [CrossRef]
28. Benito, J.J.; Ureña, F.; Gavete, L. Influence of Several Factors in the Generalized Finite Difference Method. *Appl. Math. Model.* **2001**, *25*, 1039–1053. [CrossRef]
29. Zheng, Z.; Li, X. Theoretical Analysis of the Generalized Finite Difference Method. *Comput. Math. Appl.* **2022**, *120*, 1–14. [CrossRef]
30. Li, P.-W.; Fan, C.M. Generalized Finite Difference Method for Two-Dimensional Shallow Water Equations. *Eng. Anal. Bound. Elem.* **2017**, *80*, 58–71. [CrossRef]
31. Li, P.-W.; Fan, C.-M.; Grabski, J.K. A Meshless Generalized Finite Difference Method for Solving Shallow Water Equations with the Flux Limiter Technique. *Eng. Anal. Bound. Elem.* **2021**, *131*, 159–173. [CrossRef]
32. Rao, X.; Zhao, H.; Liu, Y. A Novel Meshless Method Based on the Virtual Construction of Node Control Domains for Porous Flow Problems. *Eng. Comput.* **2023**. [CrossRef]
33. Flores, J.; Salete, E.; Benito, J.J.; Vargas, A.M.; Conde, E.R. Generalized Finite Difference Method Applied to Solve Seismic Wave Propagation Problems. Examples of 3D Simulations. *Math. Methods Appl. Sci.* **2023**. [CrossRef]
34. Qin, Q.; Song, L.; Wang, Q. High-Order Meshless Method Based on the Generalized Finite Difference Method for 2D and 3D Elliptic Interface Problems. *Appl. Math. Lett.* **2023**, *137*, 108479. [CrossRef]
35. Ju, B.; Qu, W. Three-Dimensional Application of the Meshless Generalized Finite Difference Method for Solving the Extended Fisher–Kolmogorov Equation. *Appl. Math. Lett.* **2023**, *136*, 108458. [CrossRef]
36. Li, P.W.; Fan, C.M.; Yu, Y.Z.; Song, L. A Meshless Generalized Finite Difference Scheme for the Stream Function Formulation of the Naiver-Stokes Equations. *Eng. Anal. Bound. Elem.* **2023**, *152*, 154–168. [CrossRef]
37. Liu, S.; Zhou, Z.; Zeng, W. Simulation of Elastic Wave Propagation Based on Meshless Generalized Finite Difference Method with Uniform Random Nodes and Damping Boundary Condition. *Appl. Sci.* **2023**, *13*, 1312. [CrossRef]
38. Hamaidi, M.; Naji, A.; Charafi, A. Space-Time Localized Radial Basis Function Collocation Method for Solving Parabolic and Hyperbolic Equations. *Eng. Anal. Bound. Elem.* **2016**, *67*, 152–163. [CrossRef]
39. Uddin, M.; Ali, H. The Space-Time Kernel-Based Numerical Method for Burgers' Equations. *Mathematics* **2018**, *6*, 2–11. [CrossRef]
40. Wang, F.; Fan, C.-M.; Zhang, C.; Lin, J. A Localized Space-Time Method of Fundamental Solutions for Diffusion and Convection-Diffusion Problems. *Adv. Appl. Math. Mech.* **2020**, *12*, 940–958. [CrossRef]
41. Qiu, L.; Lin, J.; Qin, Q.-H.; Chen, W. Localized Space–Time Method of Fundamental Solutions for Three-Dimensional Transient Diffusion Problem. *Acta Mech. Sin.* **2020**, *36*, 1051–1057. [CrossRef]
42. Ku, C.-Y.; Liu, C.-Y.; Yeih, W.; Liu, C.-S.; Fan, C.-M. A Novel Space–Time Meshless Method for Solving the Backward Heat Conduction Problem. *Int. J. Heat Mass Transf.* **2019**, *130*, 109–122. [CrossRef]
43. Ku, C.-Y.; Hong, L.-D.; Liu, C.-Y.; Xiao, J.-E.; Huang, W.-P. Modeling Transient Flows in Heterogeneous Layered Porous Media Using the Space–Time Trefftz Method. *Appl. Sci.* **2021**, *11*, 3421. [CrossRef]
44. Lin, J.; Zhang, Y.; Reutskiy, S.; Feng, W. A Novel Meshless Space-Time Backward Substitution Method and Its Application to Nonhomogeneous Advection-Diffusion Problems. *Appl. Math. Comput.* **2021**, *398*, 125964. [CrossRef]
45. Lei, J.; Wang, Q.; Liu, X.; Gu, Y.; Fan, C.-M. A Novel Space-Time Generalized FDM for Transient Heat Conduction Problems. *Eng. Anal. Bound. Elem.* **2020**, *119*, 1–12. [CrossRef]
46. Li, P.W. Space–Time Generalized Finite Difference Nonlinear Model for Solving Unsteady Burgers' Equations. *Appl. Math. Lett.* **2021**, *114*, 106896. [CrossRef]
47. Benito, J.J.; García, Á.; Negreanu, M.; Ureña, F.; Vargas, A.M. A Novel Spatio-Temporal Fully Meshless Method for Parabolic PDEs. *Mathematics* **2022**, *10*, 1870. [CrossRef]
48. Li, P.W.; Grabski, J.K.; Fan, C.M.; Wang, F. A Space-Time Generalized Finite Difference Method for Solving Unsteady Double-Diffusive Natural Convection in Fluid-Saturated Porous Media. *Eng. Anal. Bound. Elem.* **2022**, *142*, 138–152. [CrossRef]

49. Liu, F.; Song, L.; Jiang, M. Space-Time Generalized Finite Difference Method for Solving the Thin Elastic Plate Bending under Dynamic Loading. *Eng. Anal. Bound. Elem.* **2022**, *143*, 632–638. [CrossRef]
50. Lei, J.; Wei, X.; Wang, Q.; Gu, Y.; Fan, C.-M. A Novel Space–Time Generalized FDM for Dynamic Coupled Thermoelasticity Problems in Heterogeneous Plates. *Arch. Appl. Mech.* **2022**, *92*, 287–307. [CrossRef]
51. Li, P.W. The Space–Time Generalized Finite Difference Scheme for Solving the Nonlinear Equal-Width Equation in the Long-Time Simulation. *Appl. Math. Lett.* **2022**, *132*, 108181. [CrossRef]
52. Benito, J.J.; García, A.; Negreanu, M.; Ureña, F.; Vargas, A.M. Two Finite Difference Methods for Solving the Zakharov–Kuznetsov-Modified Equal-Width Equation. *Eng. Anal. Bound. Elem.* **2023**, *153*, 213–225. [CrossRef]
53. Flores, J.; García, A.; Negreanu, M.; Salete, E.; Ureña, F.; Vargas, A.M. A Spatio-Temporal Fully Meshless Method for Hyperbolic PDEs. *J. Comput. Appl. Math.* **2023**, *430*, 115194. [CrossRef]
54. Qu, W.; He, H. A Spatial–Temporal GFDM with an Additional Condition for Transient Heat Conduction Analysis of FGMs. *Appl. Math. Lett.* **2020**, *110*, 106579. [CrossRef]