

Article

A Path-Conservative ADER Discontinuous Galerkin Method for Non-Conservative Hyperbolic Systems: Applications to Shallow Water Equations

Xiaoxu Zhao ¹, Baining Wang ², Gang Li ^{1,*}  and Shouguo Qian ¹

¹ School of Mathematics and Statistics, Qingdao University, Qingdao 266071, China; 2018204110@qdu.edu.cn (X.Z.); sgqian@qdu.edu.cn (S.Q.)

² Qingdao No. 58 High School, Qingdao 266100, China; wangbaining6789@163.com

* Correspondence: gangli1978@qdu.edu.cn

Abstract: In this article, we propose a new path-conservative discontinuous Galerkin (DG) method to solve non-conservative hyperbolic partial differential equations (PDEs). In particular, the method here applies the one-stage ADER (Arbitrary DERivatives in space and time) approach to fulfill the temporal discretization. In addition, this method uses the differential transformation (DT) procedure rather than the traditional Cauchy–Kowalewski (CK) procedure to achieve the local temporal evolution. Compared with the classical ADER methods, the current method is free of solving generalized Riemann problems at inter-cells. In comparison with the Runge–Kutta DG (RKDG) methods, the proposed method needs less computer storage, thanks to the absence of intermediate stages. In brief, this current method is one-step, one-stage, and fully-discrete. Moreover, this method can easily obtain arbitrary high-order accuracy both in space and in time. Numerical results for one- and two-dimensional shallow water equations (SWEs) show that the method enjoys high-order accuracy and keeps good resolution for discontinuous solutions.

Keywords: non-conservative hyperbolic systems; ADER approach; DG method; DT procedure

MSC: 65M06; 65N06



Citation: Zhao, X.; Wang, B.; Li, G.; Qian, S. A Path-Conservative ADER Discontinuous Galerkin Method for Non-Conservative Hyperbolic Systems: Applications to Shallow Water Equations. *Mathematics* **2024**, *12*, 2601. <https://doi.org/10.3390/math12162601>

Academic Editor: Raimondas Ciegis

Received: 20 June 2024

Revised: 30 July 2024

Accepted: 9 August 2024

Published: 22 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many fluid problems from the fields of physics and engineering can be described as the conservation laws

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \underline{\mathbf{F}}(\mathbf{W}) = 0, \quad (1)$$

in the light of the first principles. Here, \mathbf{W} denotes the vector of conservative variables, and $\underline{\mathbf{F}}(\mathbf{W}) = (\mathbf{f}(\mathbf{W}), \mathbf{g}(\mathbf{W}))$ stands for the tensor of physical flux with $\mathbf{f}(\mathbf{W})$ and $\mathbf{g}(\mathbf{W})$ being the physical fluxes in the x - and y -directions, respectively. So far, we can deeply understand most physical movements in nature using the conservation principles. However, in the modeling of compressible multi-phase flow/multi-media flow from aerodynamics, astrophysics, aerospace, petroleum industry, etc., the non-conservative product term (i.e., spatial derivative of unknown solution) appears, thanks to the complex interaction between different phases (media). Therefore, the relevant mathematical model cannot be expressed in the conservative form. Nevertheless, the above fluid problems can be characterized by the quasi-linear non-conservative hyperbolic systems as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \underline{\mathbf{A}}(\mathbf{W}) \cdot \nabla \mathbf{W} = 0, \quad (2)$$

with $\underline{\mathbf{A}}(\mathbf{W}) = (\mathbf{A}(\mathbf{W}), \mathbf{B}(\mathbf{W}))$ being the system matrix. The block-matrix syntax is used here to give a compact notation of the 4×4 matrices $\mathbf{A}(\mathbf{W})$ and $\mathbf{B}(\mathbf{W})$. We will provide

detailed introductions to $\mathbf{A}(\mathbf{W})$ and $\mathbf{B}(\mathbf{W})$ in Section 4.2 for the two-dimensional SWEs. Herein, this system (2) is assumed to be hyperbolic, namely $\mathbf{A}(\mathbf{W})$ and $\mathbf{B}(\mathbf{W})$ have m real eigenvalues and a full set of m linearly independent eigenvectors, respectively. In particular, the system (2) will reduce to the hyperbolic systems of conservation laws (1), provided that $\mathbf{A}(\mathbf{W})$ and $\mathbf{B}(\mathbf{W})$ are Jacobian matrices of $\mathbf{f}(\mathbf{W})$ and $\mathbf{g}(\mathbf{W})$. This observation implies that the system (2) is suitable to express both conservation laws and non-conservative systems at the same time.

However, the major issue of the system (2) consists in the deficiency of the classical definition of the weak solution in the case of discontinuities. It was not until the appearance of the theory of Dal Maso, Le Floch, and Murat (DLM theory) [1] that this major problem made great progress. The DLM theory gives a definition of the weak solutions to the system (2) by introducing a path $\Psi = \Psi(\xi; \mathbf{W}^-, \mathbf{W}^+)$ to connect two states \mathbf{W}^- and \mathbf{W}^+ in phase space. Subsequently, in view of the DLM theory [1], Castro et al. [2] and Pares [3] developed the path-conservative methods according to the non-conservative hyperbolic systems (2). Actually, the path-conservative methods [2,3] can also be known as the extension of the weak formulation of Roe's method by Toumi [4].

Later, following the original achievements [1,2], many researchers have made many attempts on the path-conservative methods. Representative research mainly include ADER schemes [5,6], FORCE schemes [7], HLLC Riemann solver methods [8], Osher Riemann solver methods [9], central schemes [10], central-upwind schemes [11,12], the ADER-DG method [13], and so on. For the latest progress and a brief historical review, we refer to [14,15].

The key purpose of this research is to propose a new path-conservative DG method for non-conservative hyperbolic systems (2). The resulting method uses the one-stage ADER approach to realize a high-order temporal discretization, and is called the *path-conservative ADER-DG method* accordingly. The fundamental idea is to employ the DT procedure [16–18] instead of the CK procedure to express the spatiotemporal expansion coefficients of the solution through the low order spatial expansion coefficients. In addition, the DT procedure can enable us to realize local temporal evolution along with arbitrary high-order accuracy. As far as we are concerned, this will be the first attempt at applying the DG method along with the DT procedure for the non-conservative system. Then, we extend the proposal to deal with the SWEs in a non-conservative form. Specifically, Li et al. [13] proposed an ADER-DG method for the SWEs in the form of hyperbolic balance laws. Moreover, Li et al. [19] extended the ADER-DG method to solve the Euler equations in gas dynamics. Herein, the success of the ADER-DG method for the SWEs in a non-conservative form will illustrate the universality of the ADER-DG method.

The structure of this article is as follows: Section 2 illustrates the general framework of the path-conservative ADER-DG method and then applies the resulting method to the 1D SWEs in a non-conservative form. Section 3 deals with the two-dimensional (2D) SWEs using the proposed method. Section 4 implements canonical examples to validate the performance of the proposed method. Finally, Section 5 gives some conclusions.

2. General Formulation of Path-Conservative ADER-DG Method

Herein, we give a general framework for the ADER-DG methods according to the non-conservative system (2). Actually, we first multiply the system (2) by a test function $\phi(\vec{x})$ from a given approximation space, then integrate on a space-time cell $C_j \times [t^n, t^{n+1})$, and obtain

$$\int_{t^n}^{t^{n+1}} \int_{C_j} \frac{\partial \mathbf{W}}{\partial t} \phi \, dV dt + \int_{t^n}^{t^{n+1}} \int_{C_j} (\underline{\mathbf{A}}(\mathbf{W}) \cdot \nabla \mathbf{W}) \phi \, dV dt = 0,$$

with C_j as the spatial cell. Afterward, we approximate $\mathbf{W}(\vec{x}, t)$ with $\mathbf{W}_\tau(\vec{x}, t)$ as a spatiotemporal polynomial. Since \mathbf{W}_τ usually shows jumps at inter-cells, we present a path-conservative method to deal with the jumps. In the following, we obtain the DG method for PDE (2): for any test function ϕ , the numerical solution W_τ satisfies the following equality:

$$\int_{C_j} \mathbf{W}_\tau(\vec{x}, t^{n+1}) \phi \, dV - \int_{C_j} \mathbf{W}_\tau(\vec{x}, t^n) \phi \, dV + \int_{t^n}^{t^{n+1}} \int_{C_j \setminus \partial C_j} (\underline{\mathbf{A}}(\mathbf{W}_\tau) \cdot \nabla \mathbf{W}_\tau) \phi \, dV dt + \int_{t^n}^{t^{n+1}} \int_{\partial C_j} \mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n}) \phi \, dS dt = 0, \tag{3}$$

with \vec{n} being the unit outward normal vector at the boundary of the cell C_j .

The DG method (3) uses the ADER approach to realize the temporal discretization. Hence, the DG method (3) is one-step, one-stage, and fully-discrete, accordingly. Therefore, we call the proposed method (3) the ADER-DG method. In addition, to achieve the time integration, we need to realize the local time evolution in the space-time cell $C_j \times [t^n, t^{n+1})$ from $\mathbf{W}_\tau(\vec{x}, t^n)$ using the DT procedure. The DT procedure will be described at length in Section 2.1.3. Here, \mathbf{W}_τ^- is obtained from inside the cell C_j , and \mathbf{W}_τ^+ is obtained from the outside the cell C_j , respectively. Moreover, the notation $\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+)$ stands for the jump terms at inter-cells and satisfies the below requirements [2,3,5]:

- For every \mathbf{W}_τ and \vec{n} , $\mathcal{D}^-(\mathbf{W}_\tau, \mathbf{W}_\tau) = 0$.
- For every $\mathbf{W}_\tau^-, \mathbf{W}_\tau^+$, and \vec{n} , $\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n}) + \mathcal{D}^-(\mathbf{W}_\tau^+, \mathbf{W}_\tau^-, -\vec{n}) = \int_0^1 \left(\underline{\mathbf{A}}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) \cdot \vec{n} \right) \frac{\partial \Psi}{\partial \xi} d\xi$.

In addition, $\Psi = \Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)$ denotes a sufficiently smooth path connecting the states \mathbf{W}_τ^- and \mathbf{W}_τ^+ . A method satisfying the above conditions is called the *path-conservative method*.

Moreover, according to [13,19], the conservative equivalent of the proposed ADER-DG method (3) to solve the hyperbolic system of conservation laws (1) is as follows:

$$\int_{C_j} \mathbf{W}_\tau(\vec{x}, t^{n+1}) \phi \, dV - \int_{C_j} \mathbf{W}_\tau(\vec{x}, t^n) \phi \, dV + \int_{t^n}^{t^{n+1}} \int_{C_j \setminus \partial C_j} \underline{\mathbf{F}}(\mathbf{W}_\tau) \cdot \nabla \phi \, dV dt + \int_{t^n}^{t^{n+1}} \int_{\partial C_j} \mathcal{F}_{j+\frac{1}{2}} \phi \, dS dt = 0, \tag{4}$$

with $\mathcal{F}_{j+\frac{1}{2}} := \mathcal{F}_{j+\frac{1}{2}}(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n})$ as the numerical flux to approximate the physical flux $\underline{\mathbf{F}}$ at inter-cells. For example, people often apply the following simple and efficient Lax–Friedrichs numerical flux $\mathcal{F}_{j+\frac{1}{2}} := \frac{1}{2} \left[\underline{\mathbf{F}}(\mathbf{W}_\tau^-) \cdot \vec{n} + \underline{\mathbf{F}}(\mathbf{W}_\tau^+) \cdot \vec{n} - \alpha(\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-) \right]$, where α is an estimate of the maximum wave propagation speed. Actually, the above path-conservative ADER-DG method (3) in the non-conservative form reduces to the conservative ADER-DG method (4) in this situation where the non-conservative system (2) is a conservation law (1), i.e., \mathbf{A} and \mathbf{B} are the Jacobian matrices of \mathbf{f} and \mathbf{g} , respectively. For this equivalence, Dumbser et al. present a detailed proof under the framework of finite volume schemes [5].

2.1. Applications to 1D SWEs

Based on the general framework of the path-conservative ADER-DG method, we then take 1D SWEs

$$\begin{aligned} h_t + (hu)_x &= 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2 \right)_x &= -ghb_x \end{aligned} \tag{5}$$

as an example to illustrate the concrete construction steps of the method (3). Herein, $h(x, t)$, $u(x, t)$ denote the water depth and the fluid velocity, respectively. The notation $b(x)$ represents the bottom topography, and the letter “g” stands for the gravitational acceleration.

Following the strategy in [2], we incorporated the geometrical source term in (5) into the term $A(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x}$ and obtain the following non-conservative form

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x} = 0, \tag{6}$$

where

$$\mathbf{W} = \begin{pmatrix} h \\ hu \\ b \end{pmatrix}, \quad \mathbf{A}(\mathbf{W}) = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & c^2 \\ 0 & 0 & 0 \end{pmatrix},$$

with $c = \sqrt{gh}$ being the sound velocity. This kind of operation makes it easy to achieve a well-balanced (WB) method for the system (6).

From a mathematical point of view, the non-conservative system (6) preserves the steady state solutions, which meet

$$\mathbf{A}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x} = 0.$$

In particular, the lake at rest steady state solutions enjoy the below forms

$$h + b = \text{Constant} \quad \text{and} \quad u = 0.$$

The traditional methods fail to preserve this steady state exactly and result in non-physical oscillations. Well-balanced (WB) methods [20,21] can preserve the steady state up to the machine accuracy at the discrete level and resolve small perturbations of the steady state even on a relatively coarse mesh [22], then increase the computational efficiency correspondingly.

2.1.1. Notations and Solution Space

Firstly, the spatial domain $[a, b]$ is discreted into N spatial cells with $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ for $j = 1, 2, \dots, N$. Herein, we take $x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})$ and $\tau_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ as the mesh center and size of the cell I_j . Moreover, the maximal mesh size is defined as $\tau = \max_{1 \leq j \leq N} \tau_j$. Here, we apply $\Omega_j = I_j \times [t^n, t^{n+1})$ as the space-time cell and set

$$V_\tau^k = \left\{ \phi(x, t) : \phi(x, t)|_{\Omega_j} \in P^k(\Omega_j) \right\} \tag{7}$$

as the approximation space, where $P^k(\Omega_j)$ denotes a set of space-time polynomials on the space-time cell Ω_j with a degree up to k .

2.1.2. Construction of 1D Path-Conservative ADER-DG Method

For the 1D system (6), the ADER-DG method (3) is as follows: for $\forall \phi(x) \in V_\tau^k$, the solution $\mathbf{W}_\tau(x, t)$ meets the below equality:

$$\begin{aligned} & \int_{I_j} \mathbf{W}_\tau(x, t^{n+1}) \phi(x) \, dx - \int_{I_j} \mathbf{W}_\tau(x, t^n) \phi(x) \, dx + \int_{t^n}^{t^{n+1}} \int_{j \setminus \partial_j} \left(\mathbf{A}(\mathbf{W}_\tau) \frac{\partial \mathbf{W}_\tau}{\partial x} \right) \phi(x) \, dx \, dt \\ & + \int_{t^n}^{t^{n+1}} \int_{\partial_j} \mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n}) \phi(x) \, dS \, dt = 0, \quad \text{for } j = 1, 2, \dots, N. \end{aligned} \tag{8}$$

Further, Equation (8) also enjoys the below equivalent form

$$\int_{I_j} \mathbf{W}_\tau(x, t^{n+1}) \phi(x) dx - \int_{I_j} \mathbf{W}_\tau(x, t^n) \phi(x) dx + \int_{t^n}^{t^{n+1}} \int_{j \setminus \partial_j} \left(\mathbf{A}(\mathbf{W}_\tau) \frac{\partial \mathbf{W}_\tau}{\partial x} \right) \phi(x) dx dt$$

$$+ \int_{t^n}^{t^{n+1}} \mathcal{D}^-\left(\mathbf{W}_{j+\frac{1}{2}}^-, \mathbf{W}_{j+\frac{1}{2}}^+\right) \phi(x) dt - \int_{t^n}^{t^{n+1}} \mathcal{D}^-\left(\mathbf{W}_{j-\frac{1}{2}}^-, \mathbf{W}_{j-\frac{1}{2}}^+\right) \phi(x) dt = 0, \quad \text{for } j = 1, \dots, N. \tag{9}$$

With regard to the jump term $\mathcal{D}^-(\cdot, \cdot)$ at inter-cells, there are different choices, such as

- The Osher jump term:

$$\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) = \frac{1}{2} \int_0^1 \left(\mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) - |\mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+))| \right) \frac{\partial \Psi}{\partial \xi} d\xi. \tag{10}$$

- The Roe jump term:

$$\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) = \left(\tilde{\mathbf{A}}_\Psi(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) \right)^- \cdot (\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-), \tag{11}$$

where $\tilde{\mathbf{A}}_\Psi(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+)$ denotes the Roe linearization matrix of $\mathbf{A}(\mathbf{W}_\tau)$ in some sense defined from [4] by Toumi, i.e., a function $\tilde{\mathbf{A}}_\Psi$ meets the below properties:

- For each $\mathbf{W}_\tau^-, \mathbf{W}_\tau^+$, the matrix $\tilde{\mathbf{A}}_\Psi(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+)$ owns m different real eigenvalues

$$\lambda_1 < \lambda_2 < \dots < \lambda_m.$$

- The compatibility property

$$\tilde{\mathbf{A}}_\Psi(\mathbf{W}_\tau, \mathbf{W}_\tau) = \mathbf{A}(\mathbf{W}_\tau), \quad \text{for each } \mathbf{W}_\tau.$$

- For arbitrary $\mathbf{W}_\tau^-, \mathbf{W}_\tau^+$, the matrix $\tilde{\mathbf{A}}_\Psi$ satisfies the below requirement

$$\tilde{\mathbf{A}}_\Psi(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) \cdot (\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-) = \int_0^1 \mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) \frac{\partial \Psi}{\partial \xi} d\xi,$$

in the light of the generalized Roe property.

In addition, with respect to the absolute value operator of the matrix in (10) and (11), we usually use the following notations:

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m),$$

$$|\mathbf{\Lambda}| = \text{diag}(|\lambda_1|, |\lambda_2|, \dots, |\lambda_m|),$$

$$\mathbf{\Lambda}^- = \frac{1}{2}(\mathbf{\Lambda} - |\mathbf{\Lambda}|),$$

$$\mathbf{A}^- = \mathbf{R} \mathbf{\Lambda}^- \mathbf{R}^{-1},$$

where \mathbf{R} is the matrix of right eigenvectors of the matrix \mathbf{A} , and \mathbf{R}^{-1} stands for its inverse.

Up to now, the current method (9) can be considered as the function of a given path $\Psi(\xi)$ in the below form:

$$\Psi(\xi) = \Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+), \quad 0 \leq \xi \leq 1.$$

Moreover, the function $\Psi(\xi) = \Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)$ is Lipschitz continuous and meets certain regularity as well as the compatibility condition

$$\begin{aligned} \Psi(0; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+) &= \mathbf{W}_\tau^-, \\ \Psi(1; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+) &= \mathbf{W}_\tau^+, \\ \Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+) &= \mathbf{W}_\tau. \end{aligned}$$

In this article, we apply the simple segment path

$$\Psi(\xi) = \Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+) = \mathbf{W}_\tau^- + \xi(\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-), \quad 0 \leq \xi \leq 1, \tag{12}$$

as in [2,3,5].

Despite its simplicity, this choice of path is useful since it guarantees that the proposed method is WB for SWEs [2,3,5]. In addition, on account of this simple path (12), the Osher jump term (10) reduces to the following form:

$$D^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) = \frac{1}{2} \int_0^1 \left(\mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) - \left| \mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) \right| \right) d\xi \cdot (\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-). \tag{13}$$

Meanwhile, the Roe jump term (11) is reducible to the below form:

$$D^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) = \frac{1}{2} \left(\int_0^1 \mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) d\xi - \left| \int_0^1 \mathbf{A}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) d\xi \right| \right) \cdot (\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-). \tag{14}$$

For the computation of path integrals in (13) and (14), we apply Gaussian quadrature rules with suitably high-order accuracy.

2.1.3. The DT Procedure

To build the method (9), we need to realize the local temporal evolution in the space-time cell starting from t^n in advance. The reason for this operation is that we need to obtain the numerical solution in the form of spatiotemporal polynomials in each space-time cell. Then, we can calculate the time integration as well as the space-time integration with high-order accuracy in (9). Actually, to realize this goal, the ADER methods [23–29] use the CK procedure to repeatedly differentiate the governing PDE and to obtain the temporal derivatives using the spatial ones. To obtain high-order time accuracy, we need high-order temporal derivatives. At this time, the CK procedure will become very cumbersome due to the usage of the chain rules. Dumbser and Munz [29] proposed an efficient algorithm on account of the Leibnize rule. More recently, Dumbser et al. [5,6] and Tang et al. [30] apply the local DG predictor approach [31] to take the place of the CK procedure. More recently, Li et al. developed an ADER-DG method for SWEs using the DT procedure [13].

In this study, we apply the DT procedure rather than the CK procedure. In fact, the DT procedure was originally developed with regard to the nonlinear initial value problems [32,33]. Afterward, Ayaz generalized the DT procedure to the 2D cases [16] as well as the system cases [17]. Kurnaza et al. [18] implemented the generalization to more general n -dimensional cases. In addition, Norman and Finkel [34] applied this procedure to build multi-moment finite volume schemes for the 1D SWEs.

In the following, we give a specific definition of the DT procedure as in [16–18]. Assume that a function $q(x, t)$ in the cell I_j at t^n is known, the DT is defined as follows:

$$\tilde{q}(k_x, k_t) = \frac{1}{k_x! k_t!} \left. \frac{\partial^{k_x+k_t} u(x, t)}{\partial x^{k_x} \partial t^{k_t}} \right|_{x=x_j, t=t^n}, \quad \text{with } q(x, t) = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \tilde{q}(k_x, k_t) (x - x_j)^{k_x} (t - t^n)^{k_t}. \tag{15}$$

Here, $\tilde{q}(k_x, k_t)$ denotes the transformed function according to the original function $q(x, t)$. Actually, $\tilde{q}(k_x, k_t)$ represents the expansion coefficients with respect to $q(x, t)$ in a form of truncated Taylor series. Table 1 shows some transformed functions used here.

Table 1. Transformed functions of some functions.

Original Function	Transformed Function
$q(x, t) = c \cdot u(x, t)$	$\tilde{q}(k_x, k_t) = c \cdot \tilde{u}(k_x, k_t)$
$q(x, t) = \partial_x u(x, t)$	$\tilde{q}(k_x, k_t) = (k_x + 1) \cdot \tilde{u}(k_x + 1, k_t)$
$q(x, t) = \partial_t u(x, t)$	$\tilde{q}(k_x, k_t) = (k_t + 1) \cdot \tilde{u}(k_x, k_t + 1)$
$q(x, t) = u(x, t) \cdot v(x, t)$	$\tilde{q}(k_x, k_t) = \sum_{r=0}^{k_x} \sum_{s=0}^{k_t} \tilde{u}(r, s) \cdot \tilde{v}(k_x - r, k_t - s)$
$q(x, t) = 1/u(x, t)$	$\tilde{q}(0, 0) = 1/\tilde{u}(0, 0)$ $\tilde{q}(k_x, k_t) = -\frac{1}{\tilde{u}(0, 0)} \sum_{r=0}^{k_x} \sum_{s=0, s+r>0}^{k_t} \tilde{u}(r, s) \cdot \tilde{q}(k_x - r, k_t - s)$

Then, we specifically illustrate the implementation steps of the DT procedure. Initially, we obtain

$$\begin{aligned}
 b_\tau(x) &= \sum_{k_x=0}^k \tilde{b}(k_x)(x - x_j)^{k_x}, \\
 h_\tau(x, 0) &= \sum_{k_x=0}^k \tilde{h}(k_x, 0)(x - x_j)^{k_x}, \\
 (hu)_\tau(x, 0) &= \sum_{k_x=0}^k (\tilde{hu})(k_x, 0)(x - x_j)^{k_x},
 \end{aligned}$$

using the L^2 projection to approximate $b(x)$, $h(x, 0)$ as well as $hu(x, 0)$ in cell I_j .

Subsequently, we exert the DT procedure on both ends of (5) and obtain the following recurrence formulae:

$$\begin{aligned}
 \tilde{h}(k_x, k_t + 1) &= -\frac{k_x + 1}{k_t + 1} \cdot (\tilde{hu})(k_x + 1, k_t), \\
 (\tilde{hu})(k_x, k_t + 1) &= -\frac{k_x + 1}{k_t + 1} \cdot \left(\tilde{G}_1(k_x + 1, k_t) + \tilde{G}_2(k_x + 1, k_t) \right) + \frac{1}{k_t + 1} \cdot \tilde{G}_3(k_x, k_t).
 \end{aligned} \tag{16}$$

Herein, we use the following auxiliary variables:

$$\begin{aligned}
 \tilde{G}_1(k_x, k_t) &= \sum_{s=0}^{k_t} \sum_{r=0}^{k_x} \tilde{G}_{1,a}(r, s) \tilde{G}_{1,b}(k_x - r, k_t - s), \\
 \tilde{G}_2(k_x, k_t) &= \frac{g}{2} \sum_{s=0}^{k_t} \sum_{r=0}^{k_x} \tilde{h}(r, s) \tilde{h}(k_x - r, k_t - s), \\
 \tilde{G}_3(k_x, k_t) &= -\sum_{r=0}^{k_x} \tilde{h}(r, k_t) \tilde{G}_{3,a}(k_x - r), \\
 \tilde{G}_{1,a}(k_x, k_t) &= \sum_{s=0}^{k_t} \sum_{r=0}^{k_x} (\tilde{hu})(r, s) (\tilde{hu})(k_x - r, k_t - s), \\
 \tilde{G}_{1,b}(k_x, k_t) &= -\frac{1}{\tilde{h}(0, 0)} \sum_{s=0}^{k_t} \sum_{r=0}^{k_x} \tilde{G}_{1,b}(k_x - r, k_t - s) \tilde{h}(r, s), \\
 \tilde{G}_{3,a}(k_x) &= g(k_x + 1) \tilde{b}(k_x + 1).
 \end{aligned}$$

Afterward, putting $\tilde{h}(k_x, 0)$, $(\tilde{hu})(k_x, 0)$ and $\tilde{b}(k_x)$, $k_x = 0, 1, \dots, k$ into (16), we can recursively acquire

$$\tilde{h}(k_x, k_t), (\tilde{hu})(k_x, k_t), \text{ for } k_t = 0, 1, \dots, k; k_x = 0, 1, \dots, k - k_t,$$

which result in

$$\mathbf{W}_\tau(x, t) := (h_\tau(x, t), (hu)_\tau(x, t), b_\tau(x))^\top = \sum_{k_t=0}^k \sum_{k_x=0}^{k-k_t} \widetilde{\mathbf{W}}(k_x, k_t)(x - x_j)^{k_x} (t - t^n)^{k_t} \in V_\tau^k$$

with $\widetilde{\mathbf{W}}(k_x, k_t) := (\widetilde{h}, (\widetilde{hu}), \widetilde{b})^\top(k_x, k_t)$ in each space-time cell Ω_j . In addition, we present the detailed algorithm of the DT procedure in Algorithm A1 for a better understanding of this procedure.

Remark 1. In a word, the key function of the DT procedure is to supply a high-order temporal evolution locally for every space-time cell according to the existing solution $\mathbf{W}_\tau(x, t^n)$.

Remark 2. The CK procedure directly uses the symbolic expansions of the governing PDE and, at a cost, requires the recomputation of many terms. So, the CK procedure leads to exponential growth according to the complexity. Nevertheless, the DT procedure is relatively simple, along with a predictable complexity.

Remark 3. In practice, there is no need to apply the DT procedure according to the bottom topography b , because the bottom b only depends on the spatial variable x .

2.1.4. The Slope Limiter

Generally, a slope limiter is indispensable for discontinuous problems. Herein, we use the total variation bounded (TVB) limiter [35–37] to control nonphysical oscillations. In fact, we only implement the TVB limiter steps in terms of the numerical solutions $\mathbf{U}_\tau := (h_\tau, (hu)_\tau)^\top$ excluding the bottom topography $b_\tau(x)$, which is independent of the time t . Specifically, we need to identify “troubled cells” (i.e., cells involving discontinuities) on the basis of cell averages $\bar{\mathbf{U}}_{\tau,j}(t^{n+1}), \bar{\mathbf{U}}_{\tau,j\pm 1}(t^{n+1})$ and inter-cell values $\mathbf{U}_\tau(x_{j+\frac{1}{2}}^\pm, t^{n+1})$ from $\mathbf{W}_\tau(x, t^{n+1})$ at t^{n+1} .

For illustrative purposes, we first give some notations

$$\begin{aligned} \tilde{\mathbf{U}}_j &= \mathbf{U}_{\tau,j+\frac{1}{2}}^-(t^{n+1}) - \bar{\mathbf{U}}_{\tau,j}(t^{n+1}), \\ \tilde{\tilde{\mathbf{U}}}_j &= \bar{\mathbf{U}}_{\tau,j}(t^{n+1}) - \mathbf{U}_{\tau,j-\frac{1}{2}}^+(t^{n+1}), \\ \Delta_+ \bar{\mathbf{U}}_j &= \bar{\mathbf{U}}_{\tau,j+1}(t^{n+1}) - \bar{\mathbf{U}}_{\tau,j}(t^{n+1}), \\ \Delta_- \bar{\mathbf{U}}_j &= \bar{\mathbf{U}}_{\tau,j}(t^{n+1}) - \bar{\mathbf{U}}_{\tau,j-1}(t^{n+1}). \end{aligned} \tag{17}$$

Then, we obtain the below updated values

$$\begin{aligned} \tilde{\mathbf{U}}_j^{(\text{mod})} &= m(\tilde{\mathbf{U}}_j, \Delta_+ \bar{\mathbf{U}}_j, \Delta_- \bar{\mathbf{U}}_j), \\ \tilde{\tilde{\mathbf{U}}}_j^{(\text{mod})} &= m(\tilde{\tilde{\mathbf{U}}}_j, \Delta_+ \bar{\mathbf{U}}_j, \Delta_- \bar{\mathbf{U}}_j), \end{aligned}$$

using the TVB limiter [37] for the variables in (17). Here, $m(\cdot, \cdot, \cdot)$ is a minmod function with

$$m(a_1, a_2, a_3) = \begin{cases} a_1, & \text{if } |a_1| \leq M\Delta x^2, \\ s \cdot \min_{1 \leq i \leq 3} |a_i|, & \text{if } |a_1| > M\Delta x^2 \text{ and } s = \text{sign}(a_1) = \text{sign}(a_2) = \text{sign}(a_3), \\ 0, & \text{otherwise,} \end{cases}$$

with $M > 0$. The cell I_j will be identified as a troubled cell provided that

$$\tilde{U}_j^{(\text{mod})} \neq \tilde{U}_j \quad \text{or} \quad \tilde{\tilde{U}}_j^{(\text{mod})} \neq \tilde{\tilde{U}}_j.$$

Subsequently, the limited inter-cell values $U_\tau(x_{j+\frac{1}{2}}^\pm, t^{n+1})$ are defined with

$$\begin{aligned} U_\tau^{(\text{mod})}(x_{j+\frac{1}{2}}^-, t^{n+1}) &= \bar{U}_j + \tilde{U}_j^{(\text{mod})}, \\ U_\tau^{(\text{mod})}(x_{j-\frac{1}{2}}^+, t^{n+1}) &= \bar{U}_j - \tilde{\tilde{U}}_j^{(\text{mod})}. \end{aligned} \tag{18}$$

Finally, a polynomial $U_\tau(x, t^{n+1})$ is determined in the light of the limited inter-cell values (18) along with the cell averages.

2.1.5. Implementation Details of 1D Path-Conservative ADER-DG Method

For the 1D system (6) and inside one time interval $[t^n, t^{n+1})$, the specific procedures of the proposed method are as follows:

1. Initially, obtain $W_\tau(x, 0) = \sum_{k_x=0}^k \tilde{W}(k_x, 0)(x - x_j)^{k_x}$ in I_j , for $j = 1, 2, \dots, N$ from $W(x, 0)$.
2. Using the recursive steps (16), acquire $\tilde{W}(k_x, k_t)$ according to $\tilde{W}(k_x, 0)$ from $W_\tau(x, t^n)$ at t^n and obtain $W_\tau(x, t)$ on each Ω_j , for $j = 1, 2, \dots, N$.
3. Construct jump terms $\mathcal{D}^-(W_\tau^-, W_\tau^+)$ in accordance with (13) and (14).
4. Update to $W_\tau(x, t^{n+1})$ using the one-stage Formula (9).
5. Employ the TVB slope limiter on $W_\tau(x, t^{n+1})$ when needed.
6. Repeat steps (2)–(5).

3. Extension to 2D System

This section generalizes the path-conservative ADER-DG method to handle the below 2D SWEs

$$\begin{aligned} h_t + (hu)_x + (hv)_y &= 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x + (huv)_y &= -ghb_x, \\ (hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2}gh^2\right)_y &= -ghb_y, \end{aligned} \tag{19}$$

with v being the fluid velocity in the y -direction. In addition, the system (19) can be expressed as the non-conservative form

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x} + \mathbf{B}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial y} = 0, \tag{20}$$

with

$$\mathbf{W} = \begin{pmatrix} h \\ hu \\ hv \\ b \end{pmatrix}, \quad \mathbf{A}(\mathbf{W}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ c^2 - u^2 & 2u & 0 & c^2 \\ -uv & v & u & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}(\mathbf{W}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ c^2 - v^2 & 0 & 2v & c^2 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Similarly to the form (2), Equation (20) enjoys the below compact form

$$\frac{\partial \mathbf{W}}{\partial t} + \underline{\underline{A}}(\mathbf{W}) \cdot \nabla \mathbf{W} = 0, \tag{21}$$

with $\underline{\underline{A}}(\mathbf{W}) = (\mathbf{A}(\mathbf{W}), \mathbf{B}(\mathbf{W}))$.

Herein, the Osher jump term is

$$\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n}) = \frac{1}{2} \int_0^1 \left(\underline{\underline{A}}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+)) - |\underline{\underline{A}}(\Psi(\xi; \mathbf{W}_\tau^-, \mathbf{W}_\tau^+))| \right) \frac{\partial \Psi}{\partial \xi} d\xi. \tag{22}$$

The Roe-type jump term reads as follows:

$$\mathcal{D}^-(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+, \vec{n}) = \left(\tilde{\underline{\underline{A}}}_\Psi(\mathbf{W}_\tau^-, \mathbf{W}_\tau^+) \cdot \vec{n} \right)^- \cdot (\mathbf{W}_\tau^+ - \mathbf{W}_\tau^-). \tag{23}$$

Then, we show the specific implementation steps of the DT procedure for the 2D cases. Initially, we have

$$\begin{aligned} b_\tau(x, y) &= \sum_{k_y=0}^k \sum_{k_x=0}^k \tilde{b}(k_x, k_y) (x - x_i)^{k_x} (y - y_j)^{k_y}, \\ h_\tau(x, y, 0) &= \sum_{k_y=0}^k \sum_{k_x=0}^k \tilde{h}(k_x, k_y, 0) (x - x_i)^{k_x} (y - y_j)^{k_y}, \\ (hu)_\tau(x, y, 0) &= \sum_{k_y=0}^k \sum_{k_x=0}^k \widetilde{(hu)}(k_x, k_y, 0) (x - x_i)^{k_x} (y - y_j)^{k_y}, \\ (hv)_\tau(x, y, 0) &= \sum_{k_y=0}^k \sum_{k_x=0}^k \widetilde{(hv)}(k_x, k_y, 0) (x - x_i)^{k_x} (y - y_j)^{k_y} \end{aligned}$$

to approximate $b(x, y)$, $h(x, y, 0)$, $(hu)(x, y, 0)$, and $(hv)(x, y, 0)$ in each space-time cell $\Omega_{i,j}$.

Suppose that the 2D numerical solutions at t^n are known as follows:

$$\mathbf{U}_\tau(x, y, t^n) := (h_\tau, (hu)_\tau, (hv)_\tau)^\top(x, y, t^n) = \sum_{k_x, k_y=0}^k \tilde{\mathbf{U}}(k_x, k_y, 0) (x - x_i)^{k_x} (y - y_j)^{k_y}. \tag{24}$$

Then, as in the 1D case, the DT idea is again used to build the space-time polynomials in every cell $\Omega_{i,j} \times [t^n, t^{n+1})$ and attain the local temporal evolution accordingly.

Actually, we apply the DT procedure on both ends of (19) and obtain the following recurrence formulae

$$\begin{aligned} \tilde{h}(k_x, k_y, k_t + 1) &:= -\frac{k_x + 1}{k_t + 1} \widetilde{(hu)}(k_x + 1, k_y, k_t) - \frac{k_y + 1}{k_t + 1} \widetilde{(hv)}(k_x, k_y + 1, k_t), \\ \widetilde{(hu)}(k_x, k_y, k_t + 1) &:= -\frac{k_x + 1}{k_t + 1} \left(\widetilde{G}_1(k_x + 1, k_y, k_t) + \widetilde{G}_2(k_x + 1, k_y, k_t) \right) \\ &\quad - \frac{k_y + 1}{k_t + 1} \widetilde{G}_3(k_x, k_y + 1, k_t) + \frac{1}{k_t + 1} \widetilde{G}_4(k_x, k_y, k_t), \\ \widetilde{(hv)}(k_x, k_y, k_t + 1) &:= -\frac{k_x + 1}{k_t + 1} \widetilde{G}_3(k_x + 1, k_y, k_t) - \frac{k_y + 1}{k_t + 1} \left(\widetilde{G}_5(k_x + 1, k_y, k_t) + \widetilde{G}_2(k_x + 1, k_y, k_t) \right) \\ &\quad + \frac{1}{k_t + 1} \widetilde{G}_6(k_x, k_y, k_t), \end{aligned} \tag{25}$$

with

$$\begin{aligned} \widetilde{G}_1(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} \widetilde{G}_{1,a}(l, m, n) \frac{\widetilde{1}}{h}(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_2(k_x, k_y, k_t) &= \frac{g}{2} \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} \widetilde{h}(l, m, n) \widetilde{h}(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_{1,a}(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} (\widetilde{hu})(l, m, n) (\widetilde{hu})(k_x - l, k_y - m, k_t - n), \\ \frac{\widetilde{1}}{h}(k_x, k_y, k_t) &= -\frac{1}{\widetilde{h}(0, 0, 0)} \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} \widetilde{h}(l, m, n) \widetilde{G}_{1,b}(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_3(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} \widetilde{G}_{3,a}(l, m, n) \frac{\widetilde{1}}{h}(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_{3,a}(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} (\widetilde{hu})(l, m, n) (\widetilde{hv})(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_4(k_x, k_y, k_t) &= -\sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \widetilde{h}(l, m, k_t) \widetilde{G}_{4,a}(k_x - l, k_y - m), \\ \widetilde{G}_{4,a}(k_x, k_y) &= g(k_x + 1) \widetilde{b}(k_x + 1, k_y), \\ \widetilde{G}_5(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} \widetilde{G}_{5,a}(l, m, n) \frac{\widetilde{1}}{h}(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_{5,a}(k_x, k_y, k_t) &= \sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \sum_{n=0}^{k_t} (\widetilde{hv})(l, m, n) (\widetilde{hv})(k_x - l, k_y - m, k_t - n), \\ \widetilde{G}_6(k_x, k_y, k_t) &= -\sum_{l=0}^{k_x} \sum_{m=0}^{k_y} \widetilde{h}(l, m, k_t) \widetilde{G}_{6,a}(k_x - l, k_y - m), \\ \widetilde{G}_{6,a}(k_x, k_y) &= g(k_y + 1) \widetilde{b}(k_x, k_y + 1), \end{aligned}$$

as auxiliary variables.

4. Numerical Results

Herein, we confirm the proposed method using several canonical examples. We use the space-time polynomial of degree up to two (i.e., $k = 2$) in the approximation space in all calculations. We take the Courant–Friedrichs–Levy (CFL) number as 0.18 to ensure the numerical stability. In addition, we set the gravitational constant $g = 9.812$. To save space, we only present the numerical results using the method with Roe-type jump terms because the Roe-type method maintains the WB property.

4.1. 1D System

4.1.1. WB Property Test

Firstly, we apply an example to numerically confirm the WB property as in [38]. The initial conditions are

$$u = 0 \quad \text{and} \quad h + b = 10, \quad x \in [0, 10].$$

Specifically, we handle two different bottoms:

$$b(x) = 5 \exp(-0.4(x - 5)^2) \quad \text{and} \quad b(x) = \begin{cases} 4 & \text{if } 4 \leq x \leq 8, \\ 0 & \text{else.} \end{cases}$$

The first bottom is smooth and the second is discontinuous.

Tables 2 and 3 present the errors at $t = 0.5$ according to the two different bottom topographies. We can clearly observe that the L^1 , L^2 , and L^∞ errors are all at the level of round-off errors for different precisions, verifying the WB property accordingly even for different bottoms. From the numerical perspective, we verify that the current method can maintain the WB property.

Table 2. Errors according to the example over the first bottom.

Precision	L^1 Error		L^2 Error		L^∞ Error	
	h	hu	h	hu	h	hu
Single	3.0426×10^{-7}	4.3787×10^{-7}	1.1571×10^{-7}	1.5946×10^{-7}	3.2549×10^{-7}	2.4293×10^{-7}
Double	1.0534×10^{-15}	4.2634×10^{-15}	3.5796×10^{-15}	1.6422×10^{-15}	3.5396×10^{-15}	2.7893×10^{-15}

Table 3. Errors according to the example over the second bottom.

Precision	L^1 Error		L^2 Error		L^∞ Error	
	h	hu	h	hu	h	hu
Single	4.2451×10^{-7}	9.2060×10^{-7}	2.2103×10^{-7}	6.1686×10^{-7}	5.0883×10^{-7}	3.0952×10^{-7}
Double	4.6322×10^{-15}	6.0178×10^{-15}	1.7475×10^{-15}	2.3722×10^{-15}	2.7528×10^{-15}	1.0885×10^{-15}

4.1.2. Accuracy Test

Next, we use an example from [38] to confirm the accuracy. The initial data read as

$$h(x, 0) = 5 + \exp(\cos(2\pi x)), \quad (hu)(x, 0) = \sin(\cos(2\pi x)), \quad x \in [0, 1]$$

over the bottom $b(x) = \sin^2(\pi x)$.

Table 4 shows the errors and the orders at $t = 0.1$ with different mesh resolutions. From Table 4, we can clearly observe that the third-order convergence is achieved for the current method.

Table 4. Errors and orders for h and hu .

Cells	h		hu	
	L^1 Error	Order	L^1 Error	Order
25	7.4757×10^{-5}		1.2847×10^{-5}	
50	1.4868×10^{-6}	2.33	2.3839×10^{-6}	2.43
100	2.7975×10^{-6}	2.41	4.1852×10^{-7}	2.51
200	4.7439×10^{-7}	2.56	6.6218×10^{-8}	2.66
400	7.3005×10^{-8}	2.70	1.0190×10^{-8}	2.70
800	1.0267×10^{-8}	2.83	1.4036×10^{-9}	2.86
1600	1.3472×10^{-9}	2.93	1.8804×10^{-10}	2.90
3200	1.6723×10^{-10}	3.01	2.3670×10^{-11}	2.99

4.1.3. Perturbations of a Steady State Water Flow

We apply an example from [39] to test the proposed method for the capability to catch small perturbations. The initial data are

$$h(x, 0) = \begin{cases} 1 - b(x) + \epsilon & \text{if } 1.1 \leq x \leq 1.2, \\ 1 - b(x) & \text{else,} \end{cases} \quad \text{and} \quad u(x, 0) = 0,$$

with $\epsilon > 0$ as a parameter over a bump

$$b(x) = \begin{cases} 0.25(\cos(10\pi(x - 1.5)) + 1) & \text{if } 1.4 \leq x \leq 1.6, \\ 0 & \text{else,} \end{cases} \quad x \in [0, 2].$$

With time evolution, the initial perturbation breaks down into two pulses, which move in two different directions, as shown in Figure 1. A good agreement is achieved as in [38–40]. Obviously, the two types of pulses are all well resolved. In addition, the downstream-traveling water pulse has already passed the bump. In the figures, we can clearly see that there are no spurious numerical oscillations, verifying the essentially non-oscillatory property of the resulting method.

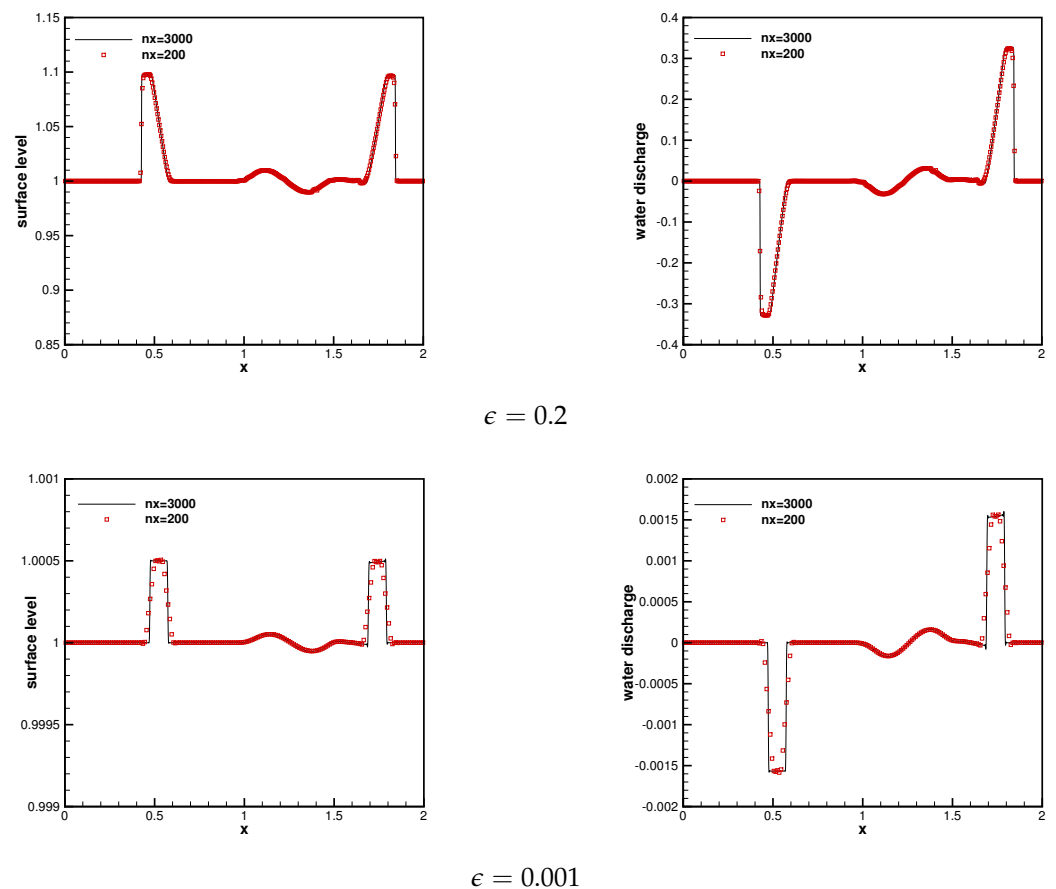


Figure 1. Surface level (left) and water discharge (right) at $t = 0.2$.

4.1.4. The Dam Break Problem over a Rectangular Bottom

Next, we handle a dam break problem [40–42] and use the below initial data

$$u(x, 0) = 0 \quad \text{and} \quad h(x, 0) = \begin{cases} 20 - b(x) & \text{if } x \leq 750, \\ 15 - b(x) & \text{else,} \end{cases}$$

over a rectangular-like bottom

$$b(x) = \begin{cases} 8 & \text{if } |x - 750| \leq 1500/8, \\ 0 & \text{else,} \end{cases} \quad x \in [0, 1500].$$

Figure 2 presents the surface level at $t = 15$ as well as $t = 60$. Our method works well for this example, producing well-resolved, non-oscillatory solutions using 500 cells which agree with the converged results using 5000 cells. In addition, the numerical results here can be compared with those from [40–42].

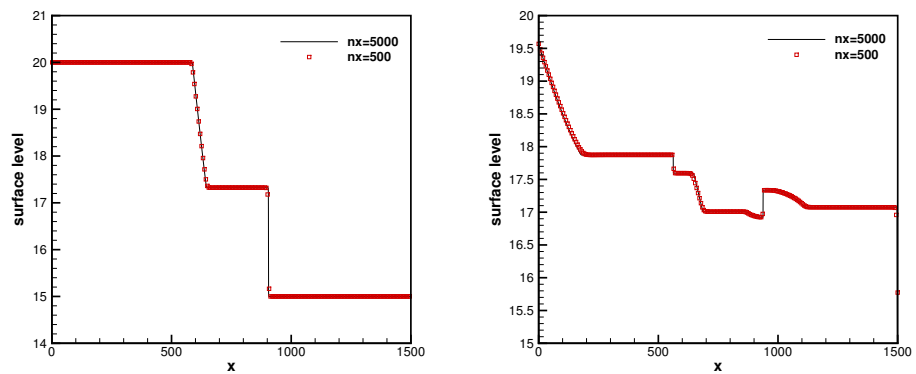


Figure 2. Surface level at $t = 15$ (left) and $t = 60$ (right).

4.1.5. Steady Flow over a Hump

Further, we validate this method using a widely-used example [43]. Actually, this example models transcritical and subcritical flows based on the initial conditions

$$u(x, 0) = 0 \quad \text{and} \quad h(x, 0) = 0.33, \quad x \in [0, 25]$$

over a hump

$$b(x) = \begin{cases} (0.2 - 0.05(x - 10)^2) & \text{if } 8 \leq x \leq 12, \\ 0 & \text{else.} \end{cases}$$

Subsequently, we exert different boundary conditions at the ends of the spatial interval.

- **Case A: the transcritical flow without a shock**
 $hu = 1.53$ at the upstream boundary; $h = 0.66$ on the downstream one.
- **Case B: the transcritical flow with a shock**
 $hu = 0.18$ on the upstream boundary; $h = 0.33$ on the downstream one.
- **Case C: the subcritical flow**
 $hu = 4.42$ on the upstream boundary; $h = 2$ on the downstream one.

Figure 3 shows the results of the above three cases. In addition, we also show the analytical solutions [44] to supply a better comparison. Regarding these three different examples, we can clearly see that the numerical results are all in good agreement with the analytical solutions, as shown in Figure 3.

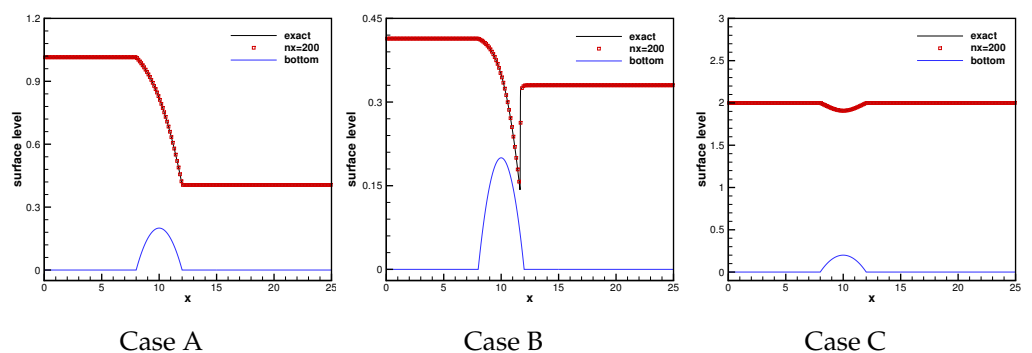


Figure 3. Surface level at $t = 200$.

4.1.6. The Dam Break Problem over a Step

Here, we consider a dam break example over different steps.

- **Case A**

Further, we implement an example from [45] with the below initial data

$$(h, u)(x, 0) = \begin{cases} (4, 0) & \text{if } x \leq 0, \\ (1, 0) & \text{else,} \end{cases} \quad x \in [-10, 10]$$

and over a step-like bottom

$$b(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 1 & \text{else.} \end{cases}$$

Over time, this example produces a rarefaction moving to the left and a shock moving to the right.

- **Case B**

The initial conditions are

$$(h, u)(x, 0) = \begin{cases} (4, 5) & \text{if } x \leq 0, \\ (1, -0.9) & \text{else,} \end{cases} \quad x \in [-10, 10]$$

over the same bottom (above). Over time, this example develops two shocks moving in different directions.

- **Case C**

This example is from [9] and the initial data are

$$(h, u)(x, 0) = \begin{cases} (0.75, -9.49365) & \text{if } x \leq 0, \\ (1.10594, -4.94074) & \text{else,} \end{cases} \quad x \in [-15, 5]$$

over a step bottom

$$b(x) = \begin{cases} 0 & \text{if } x \leq 0, \\ 0.2 & \text{else.} \end{cases}$$

- **Case D**

The initial data are

$$(h, u)(x, 0) = \begin{cases} (0.75, -1.35624) & \text{if } x \leq 0, \\ (1.10594, -4.94074) & \text{else,} \end{cases} \quad x \in [-10, 4]$$

over the same bottom of Case C.

Figure 4 presents the numerical results at $t = 1$ on the same mesh with 200 cells against the exact ones, and good agreement is clearly achieved. Moreover, all numerical results are free of spurious numerical oscillations and maintain steep discontinuous transitions at the same time.

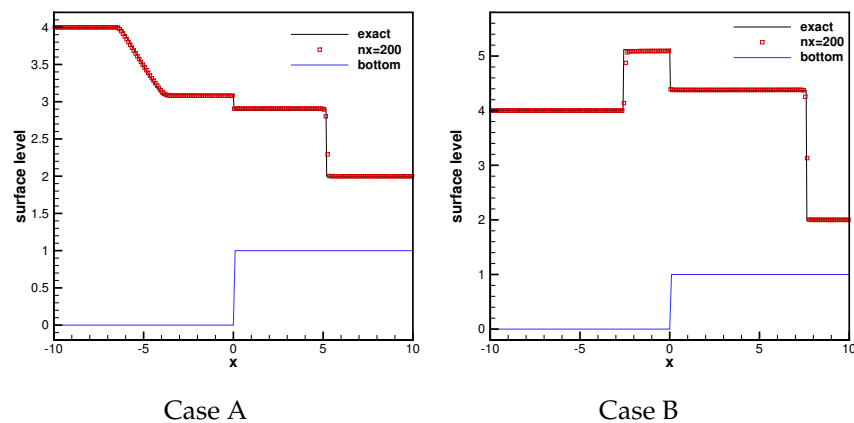


Figure 4. Cont.

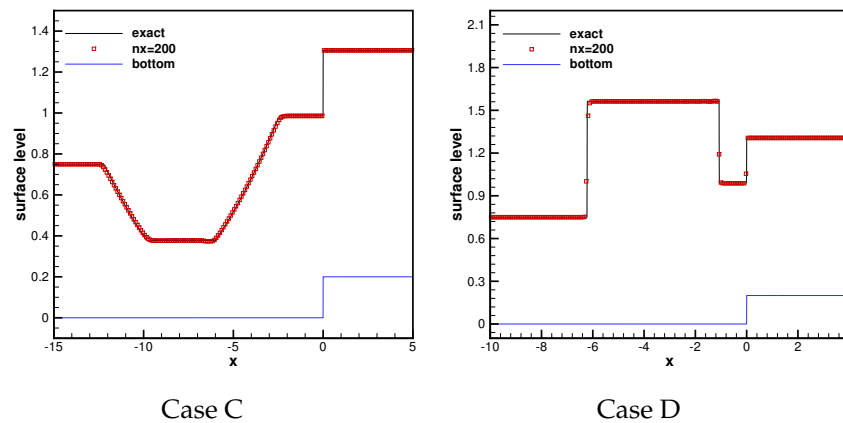


Figure 4. Surface level at $t = 1$.

4.2. 2D System

In the following, we deal with the 2D examples.

4.2.1. WB Property Test

Firstly, we borrow a numerical example from [38] to confirm the WB property and use the below initial data

$$h(x, y, 0) = 1 - b(x, y), \quad u(x, y, 0) = v(x, y, 0) = 0, \quad (x, y) \in [0, 1] \times [0, 1],$$

with $b(x, y) = \frac{4}{5} \exp(-50((x - 0.5)^2 + (y - 0.5)^2))$ as the bottom.

Table 5 shows the errors on a mesh with 100×100 cells at $t = 0.1$. Clearly, all the errors are the same as the machine accuracy; this implies the successful achievement of the WB property accordingly, even for the 2D system.

Table 5. Errors with different precisions.

Precision	L^1 Error		
	h	hu	hv
Single	4.38×10^{-7}	4.71×10^{-7}	7.49×10^{-7}
Double	7.43×10^{-15}	5.32×10^{-15}	5.95×10^{-15}

4.2.2. A Small Perturbation of a 2D Steady State Flow

In the end, we deal with an example from [39]. This example has been widely used; see [38,40,42,46,47]. The initial data are

$$h(x, y, 0) = \begin{cases} 1 - b(x, y) + 0.01 & \text{if } 0.05 \leq x \leq 0.15, \\ 1 - b(x, y) & \text{else,} \end{cases} \quad u(x, y, 0) = v(x, y, 0) = 0$$

over an elliptical-like bottom $b(x, y) = \frac{4}{5} \exp(-5(x - 0.9)^2 - 50(y - 0.5)^2)$ on $[0, 2] \times [0, 1]$. So, the surface level is almost flat, except for $0.05 \leq x \leq 0.15$, where h is perturbed upward by 0.01. Actually, the initial data can be considered to be a small perturbation to a steady state.

Figure 5 presents thirty contours of the surface level $h + b$ at different ending times and displays the right-going disturbance as it propagates past the hump on 600×300 cells. It can be observed that small complex features are all obviously resolved by the current method and the results are comparable with those in [38,40,42,46,47].

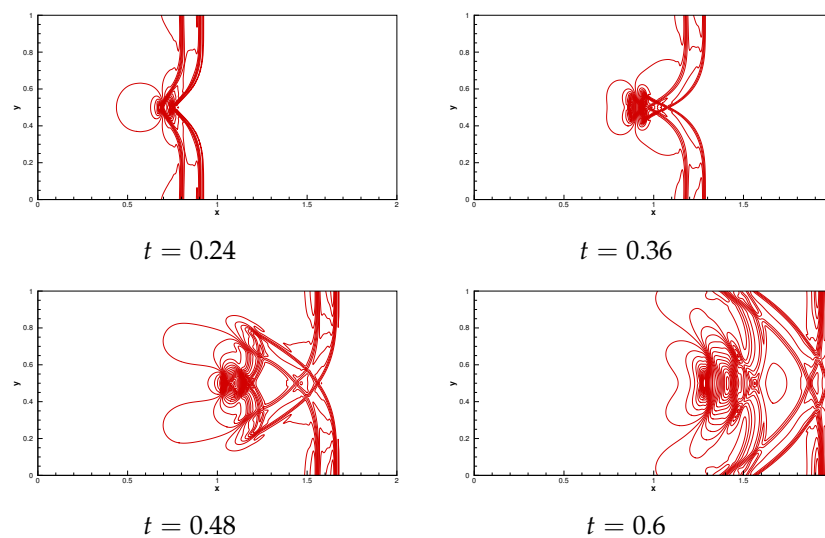


Figure 5. The contours of surface level on a mesh with 600×300 cells. Top left: at time $t = 0.24$ from 0.994836 to 1.01604; Top right: at time $t = 0.36$ from 0.988582 to 1.01117; Bottom left: at time $t = 0.48$ from 0.990344 to 1.00497; Bottom right: at time $t = 0.6$ from 0.995065 to 1.0056.

5. Conclusions

In this study, we propose a new DG method according to non-conservative hyperbolic systems. The proposed method uses the one-stage ADER approach to realize the temporal discretization. To realize the high-order local time evolution, we use the DT procedure instead of the CK procedure to recursively obtain the spatiotemporal expansion coefficients using the spatial expansion coefficients. Compared with the CK procedure, the DT procedure is more concise, and the programming is more convenient. Moreover, the proposed method needs less computer storage due to the absence of intermediate stages, and is free of solving the generalized Riemann problems at inter-cells. The resulting method is an ideal candidate for parallel computing on supercomputers, thanks to the explicit one-step nature as well as the compact stencil. We can easily proceed to arbitrary high-order accuracy both in space and time without much coding effort based on the DT procedure. In conclusion, the proposed method is one-step, one-stage, and fully-discrete. In addition, we apply the proposed method to solve the one- and two-dimensional SWEs over non-flat bottom topographies. Extensive numerical results illustrate the high-order accuracy, the WB property, and the good resolutions for discontinuous solutions. In the future, we will extend this method to solve the two-layer SWEs and two-phase flow problems.

Author Contributions: Methodology, S.Q.; Software, B.W.; Writing—original draft, X.Z.; Supervision, G.L. All authors have read and agreed to the published version of the manuscript.

Funding: The author Gang Li is supported the National Natural Science Foundation of China (Grant No. 11771228) and by the Natural Science Foundation of Shandong Province (Grant No. ZR2023MA012). The author Shouguo Qian is supported by the Natural Science Foundation of Shandong Province (Grant No. ZR2021MA072).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

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

Appendix A

Algorithm A1 The algorithm of the DT procedure for 1D SWEs

Input: $\tilde{U}(k_x, 0, 1 : 2)$ and $\tilde{b}(k_x)$
Output: $\tilde{U}(k_x, k_t, 1 : 2)$, $\tilde{F}(k_x, k_t, 1 : 2)$ and $\tilde{S}(k_x, k_t)$,
 $\% \tilde{U}(k_x, k_t, 1) \triangleq \tilde{h}(k_x, k_t)$, $\tilde{U}(k_x, k_t, 2) \triangleq (\tilde{hu})(k_x, k_t)$
 $\% \tilde{F}(k_x, k_t, 1) \triangleq \tilde{hu}(k_x, k_t)$, $\tilde{F}(k_x, k_t, 2) \triangleq (\tilde{hu}^2)(k_x, k_t) + \frac{1}{2}g(\tilde{h}^2)(k_x, k_t)$
while cell I_j , $j = 1, \dots, N$ **do**
 while $k_x = 0, \dots, k$ **do**
 $\tilde{b}_x(k_x) = (k_x + 1)b(k_x + 1)$ $\% b_x$
 end while
 while $k_t = 0, \dots, k$ **do**
 while $k_x = 0, \dots, k - k_t$ **do**
 $tot_hu2 = 0$ $\% (hu)^2$
 $tot_h_inv = 0$ $\% \frac{1}{h}$
 while $r_t = 0, \dots, k_t$ **do**
 while $r_x = 0, \dots, k_x$ **do**
 $tot_hu2 = tot_hu2 + \tilde{U}(r_x, r_t, 2) * \tilde{U}(k_x - r_x, k_t - r_t, 2)$
 end while
 end while
 $hu2(k_x, k_t) = tot_hu2$
 if $k_x + k_t > 0$ **then**
 $h_inv(k_x, k_t) = -tot_h_inv / \tilde{U}(0, 0, 1)$
 end if
 $tot_f2_a = 0$ $\% hu^2$
 $tot_h2 = 0$ $\% h^2$
 while $r_t = 0, k_t$ **do**
 while $r_x = 0, k_x$ **do**
 $tot_f2_a = tot_f2_a + hu2(r_x, r_t) * h_inv(k_x - r_x, k_t - r_t)$
 $tot_h2 = tot_h2 + \tilde{U}(r_x, r_t, 1) * \tilde{U}(k_x - r_x, k_t - r_t, 1)$
 end while
 end while
 $\tilde{F}(k_x, k_t, 1) = \tilde{U}(k_x, k_t, 2)$ $\% hu$
 $\tilde{F}(k_x, k_t, 2) = tot_f2_a + \frac{1}{2} * g * tot_h2$ $\% hu^2 + \frac{1}{2}gh^2$
 end while
 if $k_t < k$ **then**
 while $k_x = 0, \dots, k - (k_t + 1)$ **do**
 $tot_S = 0$
 while $r_x = 0, \dots, k_x$ **do**
 $tot_S = tot_S - g * \tilde{U}(r_x, r_t, 1) * \tilde{b}_x(k_x - r_x)$
 end while
 $\tilde{S}(k_x, k_t) = tot_S$ $\% \text{the source term: } -ghb_x$
 $\tilde{U}(k_x, k_t + 1, 1) = -\frac{k_x + 1}{k_t + 1} * \tilde{F}(k_x + 1, k_t, 1)$
 $\tilde{U}(k_x, k_t + 1, 2) = -\frac{k_x + 1}{k_t + 1} * \tilde{F}(k_x + 1, k_t, 2) + \frac{1}{k_t + 1} * \tilde{S}(k_x, k_t)$
 end while
 end if
 end while
end while

References

1. Maso, G.D.; LeFloch, P.G.; Murat, F. Definition and weak stability of nonconservative products. *J. Math. Pures Appl.* **1995**, *74*, 483–548.
2. Castro, M.J.; Gallardo, J.M.; Parés, C. High-order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. applications to shallow-water systems. *Math. Comput.* **2006**, *75*, 1103–1134. [[CrossRef](#)]
3. Parés, C. Numerical methods for nonconservative hyperbolic systems: A theoretical framework. *Siam J. Numer. Anal.* **2006**, *44*, 300–321. [[CrossRef](#)]
4. Tóumi, I. A weak formulation of Roe's approximate Riemann solver. *J. Comput. Phys.* **1992**, *102*, 360–373. [[CrossRef](#)]
5. Dumbser, M.; Castro, M.; Parés, C.; Toro, E.F. ADER schemes on unstructured meshes for nonconservative hyperbolic systems: Applications to geophysical flows. *Comput. Fluids* **2009**, *38*, 1731–1748. [[CrossRef](#)]
6. Dumbser, M.; Hidalgo, A.; Zanotti, O. High order space–time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. *Comput. Methods Appl. Mech. Eng.* **2014**, *268*, 359–387. [[CrossRef](#)]
7. Dumbser, M.; Hidalgo, A.; Castro, M.J.; Parés, C.; Toro, E.F. FORCE schemes on unstructured meshes II: Nonconservative hyperbolic systems. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 625–647. [[CrossRef](#)]
8. Tian, B.L.; Toro, E.F.; Castro, C.E. A path-conservative method for a five-equation model of two-phase flow with an HLLC-type Riemann solver. *Comput. Fluids* **2011**, *46*, 122–132. [[CrossRef](#)]
9. Dumbser, M.; Toro, E.F. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *J. Sci. Comput.* **2011**, *48*, 70–88. [[CrossRef](#)]
10. Castro, M.J.; Parés, C.; Puppo, G.; Russo, G. Central schemes for nonconservative hyperbolic systems. *Siam J. Sci. Comput.* **2012**, *34*, B523–B558. [[CrossRef](#)]
11. Díaz, M.J.C.; Kurganov, A.; Luna, T.M. Path-conservative central-upwind schemes for nonconservative hyperbolic systems. *Esaim Math. Model. Numer. Anal.* **2019**, *53*, 959–985. [[CrossRef](#)]
12. Chu, S.S.; Kurganov, A.; Na, M.Y. Fifth-order A-WENO schemes based on the path-conservative central-upwind method. *J. Comput. Phys.* **2022**, *469*, 111508. [[CrossRef](#)]
13. Li, G.; Li, J.J.; Qian, S.G.; Gao, J.M. A well-balanced ADER discontinuous Galerkin method based on differential transformation procedure for shallow water equations. *Appl. Math. Comput.* **2021**, *395*, 125848. [[CrossRef](#)]
14. Parés, C. Path-conservative numerical methods for nonconservative hyperbolic systems. In *Numerical Methods for Balance Laws; Quaderni di Matematica; Dipto. di Matematica della Seconda Università di Napoli: Naples, Italy, 2014; Volume 24*, pp. 67–122.
15. Castro, M.J.; Asunción, M.; Fernández-Nieto, E.D.; Gallardo, J.M.; Vida, J.M.G.; Macías, J.; Moraes, T.; Ortega, S.; Parés, C. A review on high order well-balanced path-conservative finite volume schemes for geophysical flows. *Proc. Int. Congr. Math.* **2018**, *4*, 3533–3558.
16. Ayaz, F. On the two-dimensional differential transform method. *Appl. Math. Comput.* **2003**, *143*, 361–374. [[CrossRef](#)]
17. Ayaz, F. Solutions of the system of differential equations by differential transform method. *Appl. Math. Comput.* **2004**, *147*, 547–567. [[CrossRef](#)]
18. Kurnaza, A.; Oturanc, G.; Kiris, M.E. n -Dimensional differential transformation method for solving PDEs. *Int. J. Comput. Math.* **2005**, *82*, 369–380. [[CrossRef](#)]
19. Zhang, Y.J.; Li, G.; Qian, S.G.; Gao, J.M. A new ADER discontinuous Galerkin method based on differential transformation procedure for hyperbolic conservation laws. *Comput. Appl. Math.* **2021**, *40*, 139. [[CrossRef](#)]
20. Greenberg, J.M.; Leroux, A.Y. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *Siam J. Numer. Anal.* **1996**, *33*, 1–16. [[CrossRef](#)]
21. Greenberg, J.M.; Leroux, A.Y.; Baraille, R.; Noussair, A. Analysis and approximation of conservation laws with source terms. *Siam J. Numer. Anal.* **1997**, *34*, 1980–2007. [[CrossRef](#)]
22. Noelle, S.; Xing, Y.L.; Shu, C.-W. High-order well-balanced schemes. In *Numerical Methods for Balance Laws; Puppo, G., Russo, G., Eds.; Quaderni di Matematica: Naples, Italy, 2010*.
23. Dumbser, M.; Munz, C.D. ADER discontinuous Galerkin schemes for aeroacoustics. *Comput. Rendus Mec.* **2005**, *333*, 683–687. [[CrossRef](#)]
24. Dumbser, M. *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*; Shaker Verlag: Aachen, Germany, 2005.
25. Dumbser, M.; Munz, C.D. Arbitrary high order discontinuous Galerkin schemes. In *Numerical Methods for Hyperbolic and Kinetic Problems; Cordier, S., Goudon, T., Gutnic, M., Sonnendrucker, E., Eds.; IRMA series in mathematics and theoretical physics; EMS Publishing House: Helsinki, Finland, 2005; pp. 295–333*.
26. Toro, E.F.; Millington, R.C.; Nejad, L.A.M. Towards very high order Godunov schemes. In *Godunov Methods. Theory and Applications*; Toro, E.F., Ed.; Kluwer/Plenum Academic Publishers: New York, NY, USA, 2001; pp. 905–938.
27. Titarev, V.A.; Toro, E.F. ADER: Arbitrary high order Godunov approach. *J. Sci. Comput.* **2002**, *17*, 609–618. [[CrossRef](#)]
28. Titarev, V.A.; Toro, E.F. ADER schemes for three-dimensional non-linear hyperbolic systems. *J. Comput. Phys.* **2005**, *204*, 715–736. [[CrossRef](#)]
29. Dumbser, M.; Munz, C.D. Building blocks for arbitrary high order discontinuous Galerkin schemes. *J. Sci. Comput.* **2006**, *27*, 215–230. [[CrossRef](#)]

30. Duan, J.; Tang, H. An efficient ADER discontinuous Galerkin scheme for directly solving Hamilton-Jacobi equation. *J. Comput. Math.* **2020**, *38*, 58–83.
31. Dumbser, M.; Balsara, D.; Toro, E.F.; Munz, C.D. A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes. *J. Comput. Phys.* **2008**, *227*, 8209–8253. [[CrossRef](#)]
32. Pukhov, G.E. Differential transforms and circuit theory. *Int. J. Circuit Theory Appl.* **1982**, *10*, 265–276. [[CrossRef](#)]
33. Zhou, J.K. *Differential Transformation and Its Applications for Electrical Circuits*; Huazhong University Press: Wuhan, China, 1986.
34. Norman, M.R.; Finkel, H. Multi-moment ADER-Taylor methods for systems of conservation laws with source terms in one dimension. *J. Comput. Phys.* **2012**, *231*, 6622–6642. [[CrossRef](#)]
35. Cockburn, B.; Shu, C.-W. The Runge–Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.* **1998**, *141*, 199–224. [[CrossRef](#)]
36. Shu, C.-W. TVB uniformly high-order schemes for conservation laws. *Math. Comput.* **1987**, *49*, 105–121. [[CrossRef](#)]
37. Shu, C.-W. Total-variation-diminishing time discretizations. *Siam J. Sci. Stat. Comput.* **1988**, *9*, 1073–1084. [[CrossRef](#)]
38. Xing, Y.L.; Shu, C.-W. High order finite difference WENO schemes with the exact conservation property for the shallow water equations. *J. Comput. Phys.* **2005**, *208*, 206–227. [[CrossRef](#)]
39. LeVeque, R.J. Balancing source terms and flux gradient on high-resolution Godunov methods: The quasi-steady wave-propagation algorithm. *J. Comput. Phys.* **1998**, *146*, 346–365. [[CrossRef](#)]
40. Xing, Y.L.; Shu, C.-W. A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *Commun. Comput. Phys.* **2006**, *1*, 100–134.
41. Vukovic, S.; Sopta, L. ENO and WENO schemes with the exact conservation property for one-dimensional shallow water equations. *J. Comput. Phys.* **2002**, *179*, 593–621. [[CrossRef](#)]
42. Li, G.; Caleffi, V.; Qi, Z.K. A well-balanced finite difference WENO scheme for shallow water flow model. *Appl. Math. Comput.* **2015**, *265*, 1–16. [[CrossRef](#)]
43. Vazquez-Cendon, M.E. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *J. Comput. Phys.* **1999**, *148*, 497–526. [[CrossRef](#)]
44. Goutal, N.; Maurel, F. *Proceedings of the Second Workshop on Dam-Break Wave Simulation*; Technical Report HE-43/97/016/A; Electricité de France, Département Laboratoire National d’Hydraulique, Groupe Hydraulique Fluviale: Chatou, France, 1997.
45. Alcrudo, F.; Benkhaldoun, F. Exact solutions to the Riemann problem of the shallow water equations with a bottom step. *Comput. Fluids* **2001**, *30*, 643–671. [[CrossRef](#)]
46. Li, G.; Lu, C.; Qiu, J. Hybrid well-balanced WENO schemes with different indicators for shallow water equations. *J. Sci. Comput.* **2012**, *51*, 527–559. [[CrossRef](#)]
47. Caleffi, V. A new well-balanced Hermite weighted essentially non-oscillatory scheme for shallow water equations. *Int. J. Numer. Methods Fluids* **2011**, *67*, 1135–1159. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.