Unstaggered CWENO with the subtraction method for the shallow water equations

Rony Touma[1*], Elissa Malaeb[1] and Christian Klingenberg[2]

[1*]Computer Science and Mathematics, Lebanese American University, Byblos, Lebanon.
[2*]Mathematics, University of Wuezburg, Wuezburg, Germany.

*Corresponding author(s). E-mail(s): rony.touma@lau.edu.lb;
Contributing authors: elissa.malaeb@lau.edu.lb;
christian.klingenberg@uni-wuerzburg.de;

Abstract

In this paper, we propose a third-order, essentially non-oscillatory, well-balanced, unstaggered central finite volume scheme for approximating solutions of hyperbolic systems of balance laws. The constructed scheme deploys a non-oscillating numerical solution on a single grid and avoids the heavy process of solving Riemann problems that arise at the cell interfaces. The scheme's main idea is to evolve the error function defined to be the deviation between cell average values of the numerical solution of hyperbolic balance laws and a predefined steady state of the system, as prescribed by the subtraction method. This will fulfill C-property in the case of the shallow water equations (SWE) at the discrete level without any additional treatment. The resulting method is a blend between the new unstaggered version of the central weighted essentially non-oscillatory (UCWENO) and the deviation method. The efficacy of this new method is illustrated when considering several rigorous test problems, including both scalar and SWE problems. The obtained results are inline with corresponding outcomes reported in the recent literature which confirms the robustness and efficacy of the developed scheme.

Keywords: CWENO, Unstaggered Central Schemes, Subtraction method, Hyperbolic balance laws, Shallow Water Equations.

1 Introduction

Modern high-resolution schemes for approximating solutions of hyperbolic systems of conservation laws have been developed and put into practice through extensive research in recent years. An overview of these numerical techniques is available, e.g., in [1–4]. Several of those methods and others were tried on shallow water equations, e.g. in [5–7].

Among the wide range of techniques for approximating solutions to such problems, whether on hyperbolic balance laws in general or SWEs in specific, we focus on unstaggered well-balanced central schemes, WENO schemes, and the subtraction method to be used along those methods.

To avoid the time-consuming process of solving the Riemann problems arising at the cell interfaces, Nessyahu and Tadmor (NT) [8] first proposed a non-oscillatory central scheme for the approximate solution of general hyperbolic systems conservation laws. This scheme deploys the numerical solution on two staggered grids in successive time steps. Due to the slope limitation and a piecewise linear numerical solution defined at the center of the control cells, the NT scheme is second-order accurate and prevents spurious oscillations near discontinuities. A potential drawback of the approach is that the numerical solution in NT schemes switches between two staggered grids at consecutive time steps. More specifically, a synchronization problem occurs when a treatment of the updated solution requires the previously generated solution values obtained at previous time steps. To address this issue, a great amount of research and work on developing and using Unstaggered Central Schemes (UCS) have been recently developed [4, 5, 9].

High-order accuracy is crucial for capturing details in solutions to hyperbolic systems. It can be difficult to achieve this accuracy close to discontinuities or non-smooth areas, though. This problem was addressed by Essentially Non-Oscillatory (ENO) schemes [1, 10], Weighted Essentially Non-Oscillatory (WENO) schemes [11, 12], and Central WENO (CWENO) schemes [13, 14] by utilizing stencils to minimize oscillations. While ENO schemes use single stencil selection, WENO schemes use a convex combination of all possible stencils. CWENO schemes combine both viewpoints of WENO schemes and central schemes, being a centered version of WENO schemes. CWENO schemes perform a centered reconstruction from cell averages, followed by flux approximation using a continuous extension of Runge-Kutta solvers.

Numerous new schemes, such as compact CWENO reconstruction [15], were developed using CWENO schemes as their foundation. CWENO schemes were also extended to 2D [16], coupled with a diffuse interface method under the UCNS3D framework for underwater explosion scenarios [17], and combined with a decomposition algorithm to construct numerical fluxes of SWEs over uneven bottoms [18]. They were also developed and proven effective in solving complex geometric problems and enabling simulations with multiple fluid components by extending them on mixed-element unstructured meshes and multicomponent flows using unstructured meshes [19]. Numerous applications were solved with high-order accuracy by employing CWENO schemes and their extensions. One such scheme is the fourth-order scheme for SWEs on a movable bed [20].

The subtraction method, introduced and implemented in [5, 9, 21], involves evolving

the error function, leading to obtaining the well-balanced property easily and more efficiently, as will be shown in Section 2. The error function would be obtained by subtracting a pre-defined stationary solution from the vector of conserved variables.
This paper presents a novel numerical scheme, the Unstaggered Central WENO with Subtraction method (UCWENO-Sub), for solving SWE problems in one dimension. UCWENO-Sub takes advantage of unstaggered schemes, CWENO reconstructions, and the subtraction method. Using the introduced subtraction method, the scheme develops an error function ($\Delta\bar{U}$) that represents the difference between the vector of conserved variables ($\bar{U}$) and a given steady state ($\tilde{U}$) instead of solving directly for $\bar{U}$. In this work, the chosen stationary solution ($\tilde{U}$) is the lake at rest. UCWENO-Sub maintains an unstaggered grid structure, simplifying data storage and reducing communication costs. Furthermore, third-order accuracy is achieved by applying the CWENO reconstruction method described in [14] to the error function. After the solution update, UCWENO-Sub projects the solution from the error function onto a staggered grid at the next time step (n+1) and then back onto the original unstaggered grid, preserving the unstaggered property.
This paper is organized as follows. Section 2 details the UCWENO-Sub scheme, outlining its key features and implementation steps. Section 3 presents the application of UCWENO-Sub to various benchmark problems commonly encountered in the literature. The results will be compared with those obtained using UCS-Sub (Unstaggered Central Scheme with the Subtraction method) described in [5], and other existing schemes. Finally, Section 4 provides concluding remarks and potential future research work.

2 UCWENO-Sub method

Among the many useful applications of hyperbolic partial differential equations (PDEs) is the class of shallow water equations (SWEs). These equations are extensively employed in the modeling of various free-surface flow phenomena, including river dynamics, equatorial tsunamis [22], weather forecasting [23], and optimization of hydropower generation [24].
The derivation of the SWE system is well documented in the literature (e.g., [5, 25]. Here are the governing equations that apply:

$$\begin{cases} \partial_t \bar{U} + \partial_x f(\bar{U}) = S(\bar{U}), x \in \mathcal{D} \subset \mathbb{R}, t \in \mathbb{R}^* \\ \bar{U}(x,0) = \bar{U}_0(x). \end{cases} \tag{1}$$

In this system, $\bar{U} = \begin{pmatrix} h \\ hu \end{pmatrix}$ represents the unknown vector of conserved quantities, $f(\bar{U}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix}$ denotes the flux function, and $S(\bar{U}) = \begin{pmatrix} 0 \\ -gh\frac{db}{dx} \end{pmatrix}$ represents the source term. The variables $h(x,t)$ and $u(x,t)$ represent the water depth and velocity, respectively, while $b(x)$ defines the bottom topography. The constant g represents the gravitational acceleration.

System (1) can be explicitly written as:

$$\begin{pmatrix} h \\ hu \end{pmatrix}_t + \begin{pmatrix} hu \\ hu^2 + g\dfrac{h^2}{2} \end{pmatrix}_x = \begin{pmatrix} 0 \\ -gh\dfrac{db}{dx} \end{pmatrix}. \tag{2}$$

Considering a system with flat bottom topography, the governing equations become a homogeneous hyperbolic system of partial differential equations (PDEs) with real eigenvalues and a set of linearly independent eigenvectors.
The Jacobian matrix, $J(f)$, of this sytem if given by:

$$J(f) = \begin{pmatrix} 0 & 1 \\ gh - u^2 & 2u \end{pmatrix}. \tag{3}$$

The eigenvalues and eigenvectors of the system are as follows

$$\lambda = u \pm \sqrt{gh}, \tag{4}$$

and

$$r = \begin{pmatrix} 1 \\ u \pm \sqrt{gh} \end{pmatrix}.$$

Let $\tilde{U}$ represent a known steady state solution of system (2), implying

$$\tilde{U}_t = 0. \tag{5}$$

Next, we define the error variable $\Delta\bar{U}$ by subtracting the steady state $\tilde{U}$ from the unknown solution $\bar{U}$ of system (1,2). This gives us $\Delta\bar{U} = \bar{U} - \tilde{U}$; consequently, $\bar{U} = \Delta\bar{U} + \tilde{U}$.
We can now utilize equation (5) and the balance law (1) in $\tilde{U}$:

$$f\left(\tilde{U}\right)_x = S\left(\tilde{U}\right). \tag{6}$$

Furthermore, we substitute $\bar{U}$ with $\Delta\bar{U} + \tilde{U}$ in system (1) and apply $\tilde{U}_t = 0$:

$$\Delta\bar{U}_t + f\left(\Delta\bar{U} + \tilde{U}\right)_x = S\left(\Delta\bar{U} + \tilde{U}\right). \tag{7}$$

Finally, we subtract equation (6) from equation (7):

$$\Delta\bar{U}_t + \left[f\left(\Delta\bar{U} + \tilde{U}\right) - f\left(\tilde{U}\right)\right]_x = S\left(\Delta\bar{U}\right).$$

Instead of solving system (1) at this stage, we focus on the resulting equation for the error variable, $\Delta\bar{U}$. This equation forms an IVP as follows:

$$\begin{cases} \partial_t \Delta\bar{U} + \partial_x \left[f\left(\Delta\bar{U} + \tilde{U}\right) - f\left(\tilde{U}\right) \right] = S\left(\Delta\bar{U}\right), \quad x \in \mathcal{D} \subset \mathbb{R}, t \in \mathbb{R}^{*+} \\ \Delta\bar{U}(x,0) = \Delta\bar{U}_0(x). \end{cases} \tag{8}$$

For system (8), we partition the computational domain $\mathcal{D}$ into cells of length Δx sub-categorized into 2 types: $C_i = [x_{i-1/2}, x_{i+1/2}]$, center x_i as the control cells and $D_{i+1/2} = [x_i, x_{i+1}]$, center $x_{i+1/2} = x_i + \dfrac{\Delta x}{2}$ as the staggered dual cells. Let $t^{n+1} = t^n + \Delta t$ denote the incremented time, where Δt represents the time step to be dynamically obtained as illustrated in section 3.
The error $\Delta\bar{U}_i^n = \bar{U}_i^n - \tilde{U}_i$ is evaluated at the center, x_i, of the i^{th} cell at time t^n. This

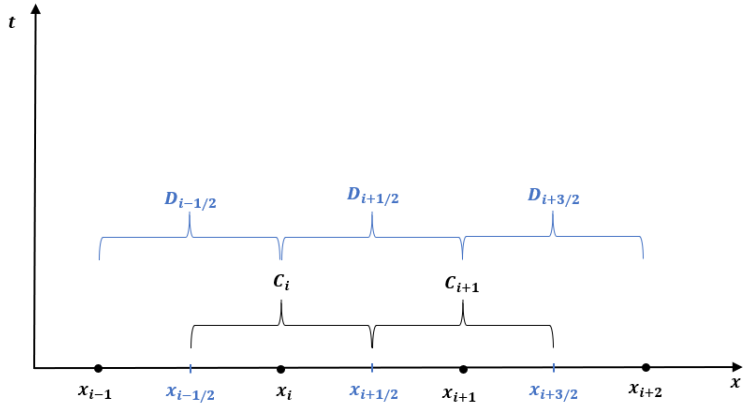

Fig. 1: Domain $\mathcal{D}$ divided into control cells C_i and staggered dual cells $D_{i+1/2}$

implies that the numerical solution, $\bar{U}_i^n$, serves as an estimation to the exact solution, $U\left(x_i, t^n\right)$, of the system (1).
To approximate the unknown error function, $\Delta U(x,t)$, a piecewise quadratic function, $M_i(x,t)$, is constructed for each cell, C_i.

$$M_i(x,t^n) = \Delta U_i^n + \left(\Delta U_i^n\right)'(x - x_i) + \frac{1}{2}\left(\Delta U_i^n\right)''(x - x_i)^2, \quad \forall x \in C_i \tag{9}$$

where ΔU_i^n and $\left(\Delta U_i^n\right)'$,$\left(\Delta U_i^n\right)''$ represent the reconstructed point-values of the error function at x_i and their derivatives respectively. The process for obtaining these reconstructed values will be addressed later in this section.
This function, $M_i(x,t)$, aims to capture the behavior of the error within that cell. This connection between the average value of $M_i(x,t)$ within cell C_i and the integral

of the exact error function is established through the following equation:

$$\Delta\bar{U}_i^n = \frac{1}{\Delta x}\int_{C_i} M_i(x,t)\,dx \approx \frac{1}{\Delta x}\int_{C_i} \Delta U(x,t)\,dx.$$

Furthermore, $M_i(x)$ leverages a convex combination of quadratic polynomials, denoted by $Q_i(x)$. These component polynomials will be defined in detail later. Equation (10) expresses this combination:

$$M_i(x,t^n) = \omega_{i-1}^i Q_{i-1}(x) + \omega_i^i Q_i(x) + \omega_{i+1}^i Q_{i+1}(x), \tag{10}$$

where ω_{i-1}^i, ω_i^i, and ω_{i+1}^i represent the weights assigned to each $Q_i(x)$ polynomial within the combination for cell C_i. These weights will also be determined later in this section, and

$$Q_k(x) = \Delta\check{U}_k + \left(\Delta\check{U}_k\right)'(x-x_k) + \frac{1}{2}\left(\Delta\check{U}_k\right)''(x-x_k)^2, \quad k = i-1,\ i,\ i+1. \tag{11}$$

Each $Q_k(x)$ in equation (11) is a quadratic polynomial centered at a specific neighboring cell. The notation k can take on values $i-1$, i, or $i+1$, corresponding to the left, center, or right neighboring cell of C_i, respectively. Similar to equation (9), these polynomials incorporate reconstructed point-values and their derivatives of the error function at the corresponding cell center.
We proceed by integrating the balance law (8) over $D_{i+1/2}\times[t^n, t^{n+1}]$. This integration yields the following equation:

$$\Delta\bar{U}_{i+1/2}^{n+1} = T_1 + T_2 + T_3, \tag{12}$$

where

$$\begin{aligned}
T_1 &= \Delta\bar{U}_{i+1/2}^n = \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \Delta\bar{U}(x,t^n)dx,\\
T_2 &= \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \Big(f_s(x_i,t) - \tilde{f}(x_i,t)\Big) - \Big(f_s(x_{i+1},t) - \tilde{f}(x_{i+1},t)\Big)\,dt,\\
T_3 &= \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} S\left(\Delta\bar{U}\right)\,dxdt,\\
f_s\left(x_i,t\right) &= f\left(\Delta\bar{U}_i + \tilde{U}_i\right),\\
\tilde{f}\left(x_i,t\right) &= f\left(\tilde{U}_i\right).
\end{aligned}$$

Our approach utilizes stencils defined within the computational domain. Each stencil comprises three sub-stencils, encompassing a total of five points. Within each cell, C_i, of the stencil, we reconstruct three quadratic polynomials, denoted by $Q_k(x)$, for $k = i-1$, i, and $i+1$. The key requirement is that each $Q_k(x)$ integrates to the

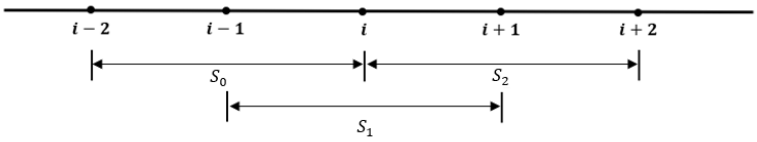

Fig. 2: Stencil made of three sub-stencils for the reconstruction of $M_i(x, t^n)$

corresponding cell-averaged value of the error function over its respective cell:

$$\begin{aligned}
&\frac{1}{\Delta x}\int_{C_{k-1}} Q_k(x)\,dx = \Delta\bar{U}_{k-1},\\
&\frac{1}{\Delta x}\int_{C_k} Q_k(x)\,dx = \Delta\bar{U}_k,\\
&\frac{1}{\Delta x}\int_{C_{k+1}} Q_k(x)\,dx = \Delta\bar{U}_{k+1}, \quad k = i-1,\ i,\ i+1.
\end{aligned}$$

These conditions, along with equation (11) defining the form of $Q_k(x)$, allow for the direct determination of the reconstructed point-values, $\Delta\check{U}$, and their derivatives:

$$\begin{aligned}
\Delta\check{U}_k &= \Delta\bar{U}_k - \frac{\Delta\bar{U}_{k-1} - 2\Delta\bar{U}_k + \Delta\bar{U}_{k+1}}{24},\\
\Delta\check{U}'_k &= \frac{\Delta\bar{U}_{k+1} - \Delta\bar{U}_{k-1}}{2\Delta x},\\
\Delta\check{U}''_k &= \frac{\Delta\bar{U}_{k+1} - 2\Delta\bar{U}_k + \Delta\bar{U}_{k+1}}{\Delta x^2}, \quad k = i-1,\ i,\ i+1.
\end{aligned} \tag{13}$$

Building upon the definitions in equations (9,10,13), we can express the reconstructed

Table 1: The constants of the UCWENO-Sub reconstruction

	c_{i-1}	c_i	c_{i+1}
cell-averages	3/16	5/8	3/16
derivatives	1/6	2/3	1/6

point-values of the error function, ΔU_i^n, and their derivatives as follows:

$$\begin{aligned}
\Delta U_i = \omega_{i-1}^i &\left(\Delta\check{U}_{i-1} + \Delta x\Delta\check{U}'_{i-1} + \frac{1}{2}\Delta x^2\Delta\check{U}''_{i-1}\right) + \omega_i^i\left(\Delta\check{U}_i\right)\\
&+ \omega_{i+1}^i\left(\Delta\check{U}_{i+1} - \Delta x\Delta\check{U}'_{i+1} + \frac{1}{2}\Delta x^2\Delta\check{U}''_{i+1}\right),
\end{aligned}$$

$$\Delta U_i' = \omega_{i-1}^i \left(\Delta \check{U}_{i-1}' + \Delta x \Delta \check{U}_{i-1}''\right) + \omega_i^i \left(\Delta \check{U}_i'\right) + \omega_{i+1}^i \left(\Delta \check{U}_{i+1}' - \Delta x \Delta \check{U}_{i+1}''\right),$$
$$\Delta U_i'' = \omega_{i-1}^i \left(\Delta \check{U}_{i-1}''\right) + \omega_i^i \left(\Delta \check{U}_i''\right) + \omega_{i+1}^i \left(\Delta \check{U}_{i+1}''\right). \tag{14}$$

The weights, denoted by ω_k^i, are calculated to ensure convex combination coefficients in the high-order numerical scheme. These weights satisfy the following properties:

$$\sum_{k=i-1}^{i+1} \omega_k^i = 1, \quad \text{and} \quad \omega_k^i \geq 0, \quad \text{for} \quad k = i-1, i, i+1.$$

We achieve this by defining the weights as:

$$\omega_k^i = \frac{\alpha_k^i}{\alpha_{i-1}^i + \alpha_i^i + \alpha_{i+1}^i}, \quad k = i-1,\ i,\ i+1,$$

where α_k^i are positive coefficients given by:

$$\alpha_k^i = \frac{c_k}{\left(\epsilon + IS_k^i\right)^2}, \quad k = i-1,\ i,\ i+1.$$

Here, ϵ is a small positive constant (typically 10^{-6}) introduced to prevent the denominator in the term α_k^i from vanishing. The constants c_k (detailed in table (1)) ensure high order accuracy of the scheme, however, they do not guarantee non-oscillatory behavior.
To address the issue of potential oscillations, we introduce smoothness indicators, IS_k^i. These indicators quantify the smoothness of the solution, $\bar{U}(x,t)$, within a specific sub-stencil, S_k, $k = i-1, i, i+1$.
The smoothness indicators are calculated using the L^2-norms of the derivatives of the reconstructed polynomials, $Q_k(x)$, within the sub-stencil:

$$IS_k^i = \sum_{l=1}^{2} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x^{2l-1} \left(Q_k^l\right)^2 dx, \quad k = i-1,\ i,\ i+1,$$

which explicitly writes as:

$$IS_{i-1}^i = \frac{1}{3}\Big(4\Delta\bar{U}_{i-2}^2 + 25\Delta\bar{U}_{i-1}^2 + 10\Delta\bar{U}_i^2 - 19\Delta\bar{U}_{i-2}\Delta\bar{U}_{i-1} + 11\Delta\bar{U}_{i-2}\Delta\bar{U}_i - 31\Delta\bar{U}_{i-1}\Delta\bar{U}_i\Big),$$
$$IS_i^i = \frac{1}{3}\Big(4\Delta\bar{U}_{i-1}^2 + 13\Delta\bar{U}_i^2 + 4\Delta\bar{U}_{i+1}^2 - 13\Delta\bar{U}_{i-1}\Delta\bar{U}_i + 5\Delta\bar{U}_{i-1}\Delta\bar{U}_{i+1} - 13\Delta\bar{U}_i\Delta\bar{U}_{i+1}\Big),$$

$$IS_{i+1}^{i} = \frac{1}{3}\Bigg(10\Delta\bar{U}_i^2 + 25\Delta\bar{U}_{i+1}^2 + 4\Delta\bar{U}_{i+2}^2 - 31\Delta\bar{U}_i\Delta\bar{U}_{i+1} + 11\Delta\bar{U}_i\Delta\bar{U}_{i+2} - 19\Delta\bar{U}_{i+1}\Delta\bar{U}_{i+2}\Bigg). \tag{15}$$

With the definitions for all constituent elements in the first $T1$ established, we can now proceed to its evaluation through explicit integration:

$$\begin{aligned} T_1 = \Delta\bar{U}_{i+1/2}^n &= \frac{1}{\Delta x}\int_{x_i}^{x_{i+1}} \Delta\bar{U}(x,t^n)dx \\ &= \frac{1}{\Delta x}\left(\int_{x_i}^{x_{i+1/2}} M_i(x)\,dx + \int_{x_{i+1/2}}^{x_{i+1}} M_{i+1}(x)\,dx\right) \\ &= \frac{\Delta U_i + \Delta U_{i+1}}{2} - \frac{\Delta U_{i+1}' - \Delta U_i'}{8}\Delta x + \frac{\Delta U_i'' + \Delta U_{i+1}''}{48}\Delta x^2. \end{aligned} \tag{16}$$

For the second term T_2, we employ Simpson's rule for numerical integration:

$$\begin{aligned} T_2 &= \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \Big(f_s(x_i,t) - \tilde{f}(x_i,t)\Big) - \Big(f_s(x_{i+1},t) - \tilde{f}(x_{i+1},t)\Big)\,dt \\ &= \frac{\Delta t}{6\Delta x}\left(\left[F_i^n + 4F_i^{n+1/2} + F_i^{n+1}\right] - \left[F_{i+1}^n + 4F_{i+1}^{n+1/2} + F_{i+1}^{n+1}\right]\right), \end{aligned} \tag{17}$$

where $F_i^n = f\left(\Delta\hat{U}_i^n + \tilde{U}_i\right) - f\left(\tilde{U}_i\right)$. Here, F_i^n represents a term evaluated at the n^{th} time step and cell interface x_i. The time-evolved terms, $\Delta\hat{U}_i^n$, $\Delta\hat{U}_i^{n+1/2}$, and $\Delta\hat{U}_i^{n+1}$, will be determined using a Runge-Kutta method with NCE.
The predicted values at $t = n+1/2$ and at $t = n+1$ are obtained through the following equations:

$$\begin{aligned} &\Delta\hat{U}\left(x_i,t^n\right) = \Delta U_i^n, \\ &\Delta\hat{U}\left(x_i,t^{n+1/2}\right) = \Delta U_i^n + \Delta t\left(d_1(1/2)K_i^1 + d_2(1/2)K_i^2\right), \\ &\quad \text{and} \\ &\Delta\hat{U}\left(x_i,t^{n+1}\right) = \Delta U_i^n + \Delta t\left(d_1(1)K_i^1 + d_2(1)K_i^2\right). \end{aligned} \tag{18}$$

The intermediate state variables, K_i^j, appearing in equation (18), are calculated based on the balance law (8) and defined as:

$$K_i^j = -f'\left(U_i^j\right) + f'\left(\tilde{U}_i\right) + S\left(Y_i^j\right). \tag{19}$$

The auxiliary variables, Y_i^j, used within the source term in equation (19), are defined as:

$$Y_i^j = \Delta U_i + \Delta t \sum_{s=1}^{j-1} a_{js} K_i^s, \tag{20}$$

where $a = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$, specific to the chosen Runge-Kutta.
U_i^j used within the pointwise reconstructed space derivative f' of the flux $f_i = f(U_i)$ in equation (19) are defined as:

$$U_i^j = Y_i^j + \tilde{U}_i,$$

where f' will be derived later in this section.
The source term appearing in equation (19) writes as:

$$S\left(Y_i^j\right) = \begin{pmatrix} 0 \\ -gY_{1,i}^j b_{xi} \end{pmatrix},$$

where b_x is a limiter coefficient. b_x can be determined using a fourth-order MC-theta limiter:

$$b_x = \text{MinMod}\left[\theta \frac{25b_i - 48b_{i-1} + 36b_{i-2} - 16b_{i-3} + 3b_{i-4}}{12\Delta x}, \right.$$
$$\left. \frac{-b_{i+2} + 8b_{i+1} - 8b_{i-1} + b_{i-2}}{12\Delta x}, \theta \frac{-3b_{i+4} + 16b_{i+3} - 36b_{i+2} + 48b_{i+1} - 25b_i}{12\Delta x}\right], \tag{21}$$

with $1 < \theta < 2$ and MinMod function as defined in [26].
Finally, d_1 and d_2 in equation (18) are defined as:

$$d_1(\alpha) = (d(1) - 1)\alpha^2 + \alpha \quad \text{and} \quad d_2(\alpha) = d(2)\alpha^2,$$

where $d = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$, also specific to the chosen Runge-Kutta.
To find the point-wise space derivative f' used in equation (19), we proceed as follows. We show the following on any dummy vector V.
We first find point-values of the flux, $f_i = f(V_i^n)$. These values are then used to establish specific interpolation requirements:

$$\begin{aligned} &Q_k(x_{k-1}) = f_{k-1}, \\ &Q_k(x_k) = f_k, \\ &Q_k(x_{k+1}) = f_{k+1}, \quad k = i-1,\ i,\ i+1, \end{aligned}$$

These requirements lead directly to the reconstructed point-values, $\check{f}_k$, and their derivatives:

$$\begin{aligned}
\check{f}_k &= f_k, \\
\check{f}'_k &= \frac{f_{k+1} - f_{k-1}}{2\Delta x}, \\
\check{f}''_k &= \frac{f_{k+1} - 2f_k + f_{k-1}}{\Delta x^2}, \quad k = i-1,\ i,\ i+1.
\end{aligned}$$

Similar to the point-wise derivative, $\Delta U'_i$, we can now find the point-wise reconstruction of the space derivative, f'_i, through a weighted average of reconstructed derivatives and second derivatives at neighboring points:

$$f'_i = \omega^i_{i-1}\left(\check{f}'_{i-1} + \Delta x \check{f}''_{i-1}\right) + \omega^i_i \check{f}'_i + \omega^i_{i+1}\left(\check{f}'_{i+1} - \Delta x \check{f}''_{i+1}\right).$$

Note that, here, we utilize the second-row coefficients, c_k, from table (1), and employ the point-wise flux values, f_i, instead of cell averages, when calculating the smoothness indicators (15).
The third term, T_3, is evaluated using Simpson's rule for both the time and space integrations. We begin by considering the time integral:

$$\begin{aligned}
T_3 &= \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}}\int_{x_i}^{x_{i+1}} S\left(\Delta \bar{U}\right)\, dxdt \\
&= \frac{\Delta t}{6\Delta x}\left[\int_{x_i}^{x_{i+1}} S\left(\Delta \hat{U}^n\right)\, dx + \int_{x_i}^{x_{i+1}} S\left(\Delta \hat{U}^{n+1/2}\right)\, dx + \int_{x_i}^{x_{i+1}} S\left(\Delta \hat{U}^{n+1}\right)\, dx\right].
\end{aligned} \tag{22}$$

Building upon the work in [27], we will employ a change of variable and integration by parts for each of the three integrals mentioned above. This manipulation aims to transform the integrals such that they involve the derivative of the water level, WL, instead of the topography derivative. We will demonstrate the process for the first integral, and the approach extends similarly to the remaining two.

$$\begin{aligned}
\int_{x_i}^{x_{i+1}} S\left(\Delta \hat{U}^n\right)\, dx &= -\int_{x_i}^{x_{i+1}} g\Delta h^n \frac{db}{dx}\, dx \\
&= -\int_{x_i}^{x_{i+1}} g(h^n - \tilde{h})\frac{db}{dx}\, dx \\
&= -\int_{x_i}^{x_{i+1}} gh^n \frac{db}{dx}\, dx + \int_{x_i}^{x_{i+1}} g\tilde{h}\frac{db}{dx}\, dx.
\end{aligned}$$

Using the relations $h = WL - b$ and $\tilde{h} = \tilde{WL} - b$, the term on the right becomes

$$-\int_{x_i}^{x_{i+1}} g((WL)^n - b)\frac{db}{dx}\, dx + \int_{x_i}^{x_{i+1}} g(\tilde{WL} - b)\frac{db}{dx}\, dx$$

$$= g\left(b_i \Delta\hat{W}L_i^n - b_{i+1}\Delta\hat{W}L_{i+1}^n\right) + g\int_{x_i}^{x_{i+1}} b\frac{d(\Delta\hat{W}L^n)}{dx}\,dx. \quad (23)$$

While analytically both integral forms are valid, the latter form involving the water level gradient is preferable for numerical computations. This is because the water level gradient tends to be smoother than the bed slope gradient, especially in scenarios with irregular bottom topography or discontinuous water levels. Consequently, the latter form is less prone to numerical errors.
Therefore, we proceed with the integration of the source term using the final form obtained after integration by parts (Equation (23)):

$$g\int_{x_i}^{x_{i+1}} b\frac{d(\Delta\hat{W}L^n)}{dx}\,dx = g\left[\frac{\Delta x}{6}\left(b_i\left(\Delta\hat{W}L^n\right)'_i + 4b_{i+1/2}\left(\Delta\hat{W}L^n\right)'_{i+1/2} + b_{i+1}\left(\Delta\hat{W}L^n\right)'_{i+1}\right)\right],$$

where $\left(\Delta\hat{W}L^n\right)'_i$ can be determined using a fourth-order MC-theta limiter as shown in equation (21).
The water level gradient at the half-point is defined as follows:

$$\left(\Delta\hat{W}L^n\right)'_{i+1/2} \approx \frac{1}{\Delta x}\left(-\frac{\Delta\hat{W}L_{i+2}^n}{24} + \frac{27\Delta\hat{W}L_{i+1}^n}{24} - \frac{27\Delta\hat{W}L_{i-1}^n}{24} + \frac{\Delta\hat{W}L_{i-2}^n}{24}\right).$$

We can now obtain the projected error function at time t^{n+1} on the dual cells $D_{i+1/2}$, $\Delta\bar{U}_{i+1/2}^{n+1}$, by summing the contributions from the three terms, T_1, T_2, and T_3:

$$\Delta\bar{U}_{i+1/2}^{n+1} = T_1 + T_2 + T_3.$$

With $\Delta\bar{U}_{i+1/2}^{n+1}$ in hand, we need the reconstructed point-values, $\Delta U_{i+1/2}^{n+1}$, and their derivatives. This can be achieved by repeating the steps outlined in equations (13-15), but using $\Delta\bar{U}_{i+1/2}^{n+1}$ as the starting point instead of $\Delta\bar{U}_i^n$.
Finally, to obtain the evolved solution, $\bar{U}_i^{n+1}$, at the cell center, C_i, and time step, t^{n+1}, we employ Taylor series back projection on $\Delta U_{i+1/2}^{n+1}$. This correlation accounts for the spatial variation within the cell. The resulting formula is shown below:

$$\Delta U_i^{n+1} = \frac{\Delta U_{i-1/2} + \Delta U_{i+1/2}}{2} - \frac{\Delta U'_{i+1/2} - \Delta U'_{i-1/2}}{8}\Delta x + \frac{\Delta U''_{i-1/2} + \Delta U''_{i+1/2}}{48}\Delta x^2.$$

The final evolved solution at t^{n+1} on C_i can then be determined by adding the update, ΔU_i^{n+1}, to the reference state, $\tilde{U}_i$:

$$\bar{U}_i^{n+1} = \Delta U_i^{n+1} + \tilde{U}_i.$$

To accurately simulate still water (or to fulfill the C-property) in the context of shallow water equations, one needs to apply a well-balanced numerical technique to maintain such steady states. This means that if the initial conditions of the SWE system correspond to a lake at rest, then the numerical scheme should capture this steady state at any time during the simulations; this means that the water depth should remain constant and still. The proposed UCWENO-Sub numerical scheme achieves this key property by perfectly fulfilling the steady states requirements at the discrete level. In fact, we will prove that if $\bar{U}_i^n$ is a stationary solution of system (1), $\bar{U}_i^n = \tilde{U}_i$, (i.e., $\Delta\bar{U}_i^n = 0$) then $\bar{U}_i^{n+1}$, the newly generated numerical solution at time t^{n+1}, is also a stationary solution of system (1), so it remains unchanged over time, $\bar{U}_i^{n+1} = \tilde{U}_i$, (i.e. $\Delta\bar{U}_i^{n+1} = 0$). The proof is as follows:

1. In the first step we show that $T_1 = 0$.
   Using $\Delta\bar{U}_i^n = 0$, true for any cell center x_i, substitute $\Delta\bar{U}_l = 0$, for $l = k-1$, k, and $k+1$ in equation (13):

$$\begin{aligned}
\Delta\check{U}_k &= \Delta\bar{U}_k - \frac{\Delta\bar{U}_{k-1} - 2\Delta\bar{U}_k + \Delta\bar{U}_{k+1}}{24} = 0,\\
\Delta\check{U}'_k &= \frac{\Delta\bar{U}_{k+1} - \Delta\bar{U}_{k-1}}{2\Delta x} = 0,\\
\Delta\check{U}''_k &= \frac{\Delta\bar{U}_{k+1} - 2\Delta\bar{U}_k + \Delta\bar{U}_{k+1}}{\Delta x^2} = 0.
\end{aligned}$$

   Using those results, substitute $\Delta\check{U}_l = 0$, $\Delta\check{U}'_l = 0$, and $\Delta\check{U}''_l = 0$ for $l = i-1$, i, and $i+1$ in equation (14):

$$\begin{aligned}
\Delta U_i = \omega_{i-1}^i &\left(\Delta\check{U}_{i-1} + \Delta x\Delta\check{U}'_{i-1} + \frac{1}{2}\Delta x^2\Delta\check{U}''_{i-1}\right) + \omega_i^i\left(\Delta\check{U}_i\right)\\
&+ \omega_{i+1}^i\left(\Delta\check{U}_{i+1} - \Delta x\Delta\check{U}'_{i+1} + \frac{1}{2}\Delta x^2\Delta\check{U}''_{i+1}\right) = 0,\\
\Delta U'_i = \omega_{i-1}^i &\left(\Delta\check{U}'_{i-1} + \Delta x\Delta\check{U}''_{i-1}\right) + \omega_i^i\left(\Delta\check{U}'_i\right) + \omega_{i+1}^i\left(\Delta\check{U}'_{i+1} - \Delta x\Delta\check{U}''_{i+1}\right) = 0,\\
\Delta U''_i = \omega_{i-1}^i &\left(\Delta\check{U}''_{i-1}\right) + \omega_i^i\left(\Delta\check{U}''_i\right) + \omega_{i+1}^i\left(\Delta\check{U}''_{i+1}\right) = 0.
\end{aligned}$$

   Using those results, substitute $\Delta U_l = 0$, $\Delta U'_l = 0$, and $\Delta U''_l = 0$, fr $l = i-1$, i, and $i+1$ in equation (16):

$$\begin{aligned}
T_1 &= \frac{\Delta U_i + \Delta U_{i+1}}{2} - \frac{\Delta U'_{i+1} - \Delta U'_i}{8}\Delta x + \frac{\Delta U''_i + \Delta U''_{i+1}}{48}\Delta x^2\\
&= 0.
\end{aligned}$$

2. Using $\Delta U_i^n = 0$ and equations (17,18,19,20), we get $T_2 = 0$.
3. Using equation (22), we get $T_3 = 0$.
4. Finally we deduce that $\Delta\bar{U}_{i+1/2}^{n+1} = T_1 + T_2 + T_3 = 0$.

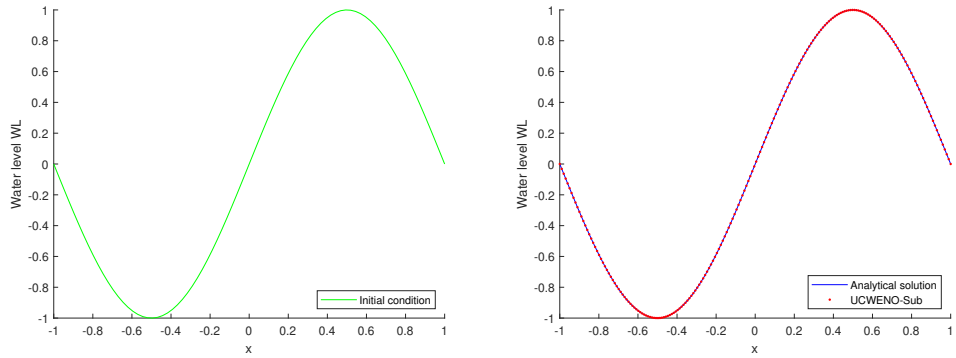

Fig. 3: Linear advection problem 1: Initial profile at $t_0 = 0$ (left) and profile acquired analytically versus UCWENO-Sub at $t_f = 10$ (right)

Naturally, we get $\Delta U_i^{n+1} = 0$, and then the lake at rest is preserved and the C-property is possessed.

3 Numerical Experiments

In this section, we apply the numerical method we presented in this paper and solve classical problems in SWEs. The developed numerical scheme is first validated and we confirm its order of convergence following a grid convergence approach; then we validate its well-balanced characteristic at the discrete level.

3.1 Scalar equation problems

We first start with scalar equations previously discussed in [14] as test problems to check the performance of our scheme. In all the scalar test problems, the boundary conditions, BCs, are periodic. In the advection problem, the time step is $\Delta t = 0.9 \times (3/7) \times \Delta x$; whereas in Burger's equation problem, the time step is $\Delta t = 0.66 \times (3/7) \times \Delta x$ as per the conditions stated in [14].

3.1.1 Advection test 1

The first test problem is a linear advection.

$$U_t + U_x = 0.$$

$[-1, 1]$ is the computational domain, and 200 grid points are used to partition it into equal-sized subintervals. The numerical solution is resolved at the final time $t_f = 10$. The initial condition, IC, is sinusoidal defined by $U(x, t = 0) = \sin(\pi x)$.

The profile of the IC at $t_0 = 0$ is displayed in figure (3) to the left, while the generated solution acquired at t_f both analytically and utilizing UCWENO-Sub scheme is presented in figure (3) to the right. The obtained findings reveal a great fit between our scheme and the analytical solution. The order of convergence obtained by following

grid refinements is illustrated in table (2); the obtained results reveal that developed numerical scheme is third-order accurate.

Table 2: Linear advection 1, $U_0 = sin(\pi x)$: L_1 error and order of convergence

N	L_1 error h	L_1 order
200	$3.21e-06$	
400	$3.80e-07$	3.08
800	$4.67e-08$	3.02

3.1.2 Burgers equation problem

The fourth test problem is a Burgers equation problem as discussed in [14]:

$$U_t + \left(\frac{1}{2}U^2\right)_x = 0.$$

80 points are used to divide the domain defined by $[-1, 1]$. The numerical solution is found at $t = 0.5$ and then at $t_f = 1.5$. Findings obtained at $t = 0.5$ are used for the computation of the convergence rate, while findings obtained at time $t = 1.5$ are used for the shock capturing test (the shock develops at $t = 2/\pi$). The IC is $U(x, t = 0) = 1 + \frac{1}{2}\sin(\pi x)$.

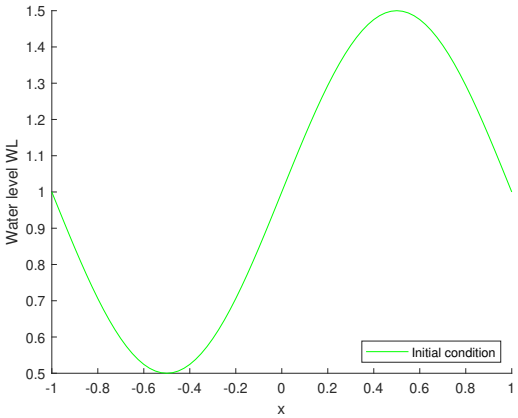

Fig. 4: Burgers equation problem: Initial profile at $t_0 = 0$

Figure (4) shows the initial data profile at $t_0 = 0$. Figure (5) shows the profile acquired at t_f using CWENO scheme [14] versus UCWENO-Sub scheme developed in

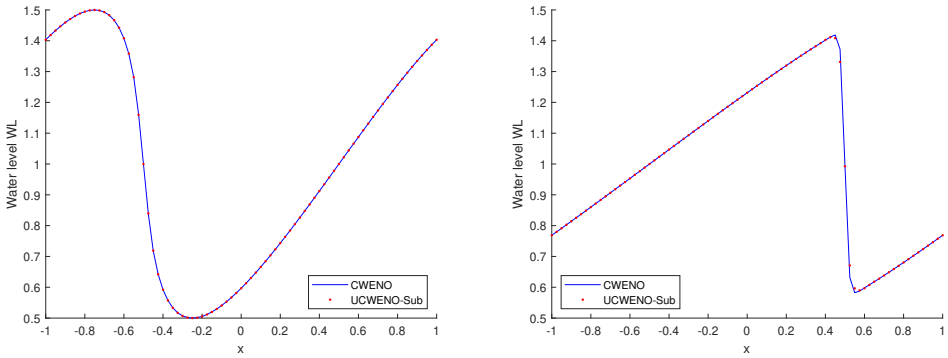

Fig. 5: Burgers equation problem: Profile acquired using CWENO versus UCWENO-Sub at $t_f = 0.5$ (left) and at $t_f = 1.5$ after the formation of the shock (right)

this paper. The reported findings show a perfect match between the results. Table 3 shows the computed order of convergence obtained prior to the formation of the shock wave and confirms the third order of accuracy of the developed scheme.

Table 3: Burgers equation problem: L_1 error and order of convergence prior to the formation of the shock wave

N	L_1 error h	L_1 order
80	$5.73e-07$	
160	$6.96e-08$	3.04
320	$8.59e-09$	3.02

3.2 System of equations: Shallow water system of equations

Setting $F(\Delta\bar{U}) = f_s - \tilde{f}$, the PDE in (8) can be expressed as follows:

$$\Delta\bar{U}_t + F(\Delta\bar{U})_x = S(\Delta\bar{U}). \tag{24}$$

Now, as explained in [28], the Jacobian matrix $J(F) = \dfrac{\partial F}{\partial \Delta\bar{U}}$ in equation (24) can be replaced by the Jacobian matrix $J(f) = \dfrac{\partial f}{\partial \bar{U}}$ given in equation (3) of system (2) because they are equal. Accordingly the time step for each iteration of the scheme can be defined as follows:

$$\Delta t = \mathrm{CFL}\frac{\Delta x}{\max_i |\lambda_i|}$$

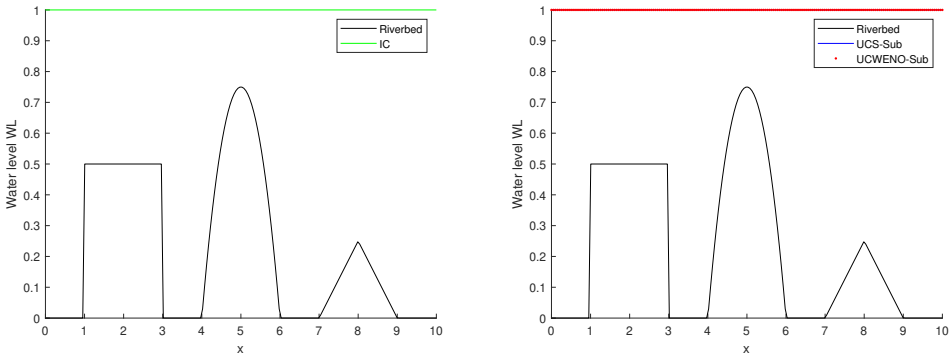

Fig. 6: Lake at rest: WL_0 at $t_0 = 0$ (left) and WL acquired using UCS-Sub versus UCWENO-Sub at $t_f = 1$ (right)

where CFL= 0.485 and where the λ_i's denote the eigenvalues of the Jacobian matrix $J_i(f_i) = \dfrac{\partial f_i}{\partial \bar{U}_i}$ over C_i as mentioned in equation (4).
In the following section, we shall address a few classical SWEs problems raised in the literature. Our main objective is to compare the final results we get by employing our method to ones acquired by applying UCS-Sub described in [5]. In a few other problems, we will compare our results to ones obtained by other higher-order schemes. As previously mentioned, the lake at rest steady state is the stationary solution $\tilde{U}$ to be used:

$$\tilde{U}_i^n = \begin{pmatrix} \tilde{h}_i^n \\ \tilde{h}_i^n \tilde{u}_i^n \end{pmatrix} \quad \text{such that} \quad \tilde{h}_i^n + b_i = \text{constant and} \quad \tilde{u}_i^n = 0.$$

3.2.1 Lake at rest problem

We examine the problem of a water flow in one dimension over varied bottom topography. This test pertains to the C-property. The computational domain $0 \leq x \leq 10$ is divided using 200 equally spaced grid points. The riverbed function is defined according to the below formula:

$$b(x) = \begin{cases} 0.5, & 1 \leq x \leq 3, \\ -0.75x^2 + 7.5x - 18, & 4 \leq x \leq 6, \\ -|0.25x - 2| + 0.25, & 7 < x \leq 9, \\ 0, & \text{otherwise.} \end{cases}$$

The ICs feature a lake at rest configuration with $WL_0 = 1$ and $u_0 = 0$. The numerical solution is generated using UCWENO-Sub scheme at $t_f = 1$.

Figure (6) shows WL_0 at $t_0 = 0$ (left) and shows WL acquired at $t_f = 1$ using UCS-Sub versus UCWENO-Sub scheme (right). The steady-state requirement is entirely resolved by the numerical solution obtained by the UCWENO-Sub approach. This is a

compelling sign of the potential of the procedure to manage and capture steady-state problems at the discrete level. The outcomes produced using UCS-Sub agree perfectly with the numerical solution generated using UCWENO-Sub scheme.

3.2.2 Steady flow problem, variable bathymetry

We now examine steady flow across variable bathymetry that was previously examined in [29]. We want to study the time convergence of steady flow over variable bathymetry. We use 201 grid points to mesh the computational domain $[0, 25]$. The riverbed is defined by the following function

$$b(x) = \begin{cases} \frac{1}{5} - \frac{1}{20}(x-10)^2, & 8 \leq x \leq 12, \\ 0, & \text{otherwise.} \end{cases}$$

We set u_0 to be null and the numerical solution is generated at $t_f = 200$. Three distinct instances are possible: transcritical without shock, transcritical with shock, and subcritical. Based on the BCs, we observe for the:

- Subcritical flow, upstream discharge hu is 4.42 and downstream water level is 2.
- Transcritical flow without shock, discharge of inflow water from the left is 1.53 and downstream water level is imposed at 0.66 if the motion is subcritical.
- Transcritical flow with shock, upstream discharge is 0.18 and downstream water level is 0.33.

The profile of WL_0 at time $t_0 = 0$ is depicted in figure (7). Using UCS-Sub and

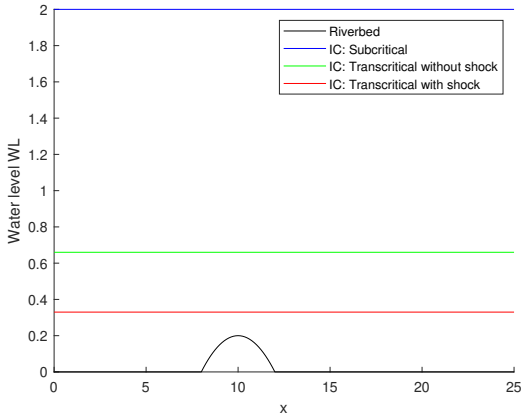

Fig. 7: Steady flow, variable bathymetry: WL_0 at $t_0 = 0$

UCWENO-Sub scheme, we generate the numerical solution at $t_f = 200$, and we report the results in figures (8,9) displaying the WL component in subcritical and transcritical with shock cases. Both schemes show excellent results. Again, the small difference

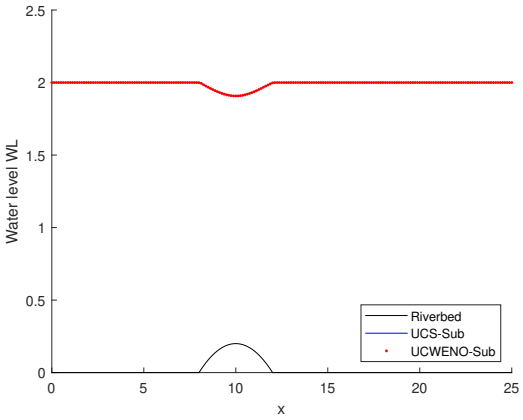

Fig. 8: Steady flow, variable bathymetry; subcritical flow: WL acquired using UCS-Sub versus UCWENO-Sub at $t_f = 200$

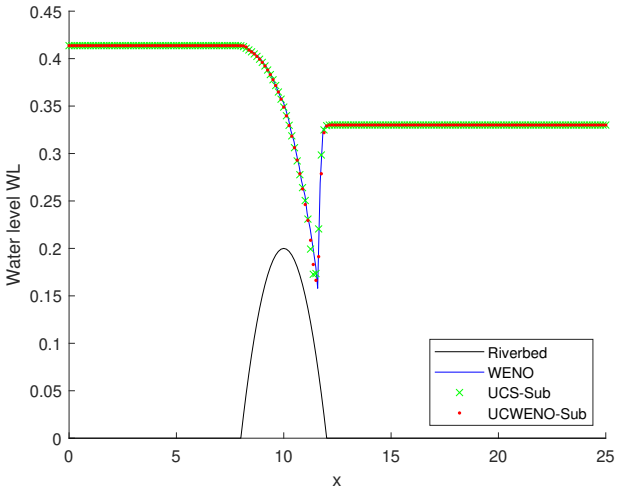

Fig. 9: Steady flow, variable bathymetry; transcritical flow with shock: WL acquired using WENO (4^{th}-order) versus UCS-Sub versus UCWENO-Sub at $t_f = 200$

in the inward propagating wave in figure (9) is because UCWENO-Sub is a third-order accurate scheme as opposed to UCS-Sub which is only second-order accurate. In figure (9), a comparison with results from WENO scheme displayed in [30] is also shown. Those results were obtained using WebPlotDigitizer. The presented comparisons are generated using a fourth-order WENO scheme and third-order UCWENO-Sub scheme and are in great agreement. In addition, we illustrate in table (4) the order of convergence of the proposed method generated following mesh refinement. The reported

results confirm the third order of convergence of the proposed numerical method. The results shown in table (4) validate our scheme's high-order accuracy, and the figure shows how third-order UCWENO-Sub is scheme in better agreement with fourth-order WENO scheme than the second-order UCS-Sub is. Fourth-order UCWENO-Sub scheme can be easily obtained and is expected to give better approximations to analytical solutions than what we can see here with the advantage of its being easier to implement, as we saw in the C-property possession that no further treatment for the source term was needed.

Table 4: Steady flow, variable bathymetry; subcritical case: L_1 error and order of convergence

N	L_1 error h	L_1 order
200	$7.62e-08$	
400	$6.93e-09$	3.46
800	$6.56e-10$	3.40

3.3 1-shock and 2-shock problem

We study now a shock problem previously mentioned in [6] where they worked with a high-order CWENO scheme. The computational domain is $[-10, 10]$ and it features a step bottom topography defined by

$$b(x) = \begin{cases} 0, & x \leq 0, \\ 1, & \text{otherwise}, \end{cases} \tag{25}$$

while the ICs for this problem are defined

$$h_0 = \begin{cases} 4, & x \leq 0, \\ 1, & \text{otherwise} \end{cases} \quad \text{and} \quad u_0 = \begin{cases} 5, & x \leq 0, \\ -0.9, & \text{otherwise}. \end{cases} \tag{26}$$

Two shocks are generated by this test case, the first traveling to the left and the second to the right. The numerical solution is resolved at $t_f = 1$.

WL_0 at $t_0 = 0$ is depicted in figure (10) (left). In Figure (10) (right), WL acquired using CWENO scheme versus UCWENO-Sub scheme at t_f. The results using CWENO scheme were devised using WebPlotDigitizer from [6]. One can clearly see that the generated numerical results using UCWENO-Sub scheme are in great agreement with the analytical solution of the problem and are a better approximation to it than the results using CWENO scheme as the figure shows that the points relating to UCWENO-Sub scheme are actually closer to the analytical solution than the crosses that relate to CWENO scheme.

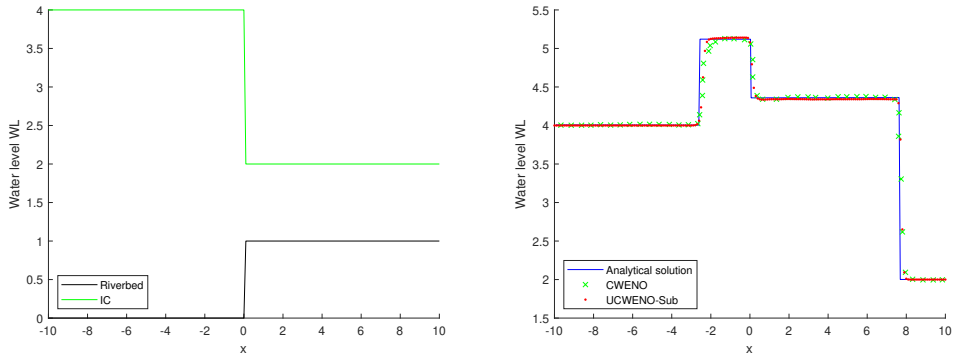

Fig. 10: 1-shock and 2-shock problem: WL_0 at $t_0 = 0$ (left) and WL acquired using CWENO versus UCWENO-Sub at $t_f = 1$ (right)

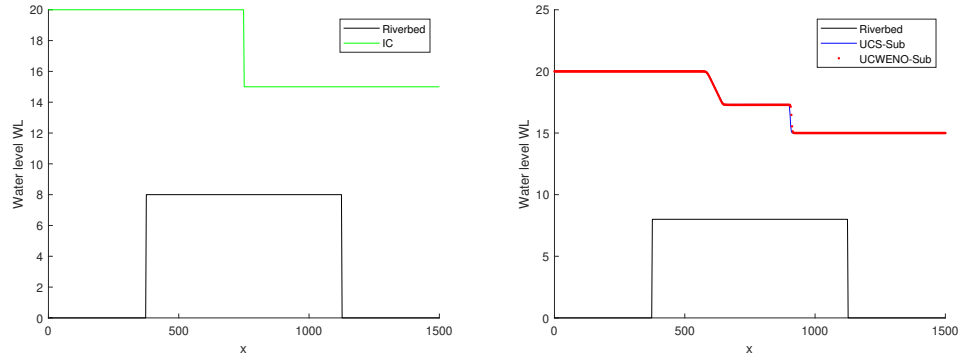

Fig. 11: Dam break, discontinuous bathymetry: WL_0 at $t_0 = 0$ (left) and WL acquired using UCS-Sub versus UCWENO-Sub at $t_f = 15$ (right)

3.3.1 Dam break problem, discontinuous bathymetry

We study the dam break over discontinuous bathymetry. The computational domain is $[0, 1500]$. 600 grid points are used to partition the domain into equal-sized segments. The numerical solution is resolved at $t_f = 15$.
The riverbed function:

$$b(x) = \begin{cases} 8, & 375 < x < 1125, \\ 0, & \text{otherwise}, \end{cases} \tag{27}$$

and the ICs are

$$WL_0 = \begin{cases} 20, & x < 750, \\ 15, & \text{otherwise}, \end{cases} \tag{28}$$

and $u_0 = 0$.

WL_0 at $t_0 = 0$ is depicted in figure (11) to the left. In Figure (11) to the right, WL acquired using UCS-Sub versus UCWENO-Sub scheme at various times and at t_f, respectively. The outcomes demonstrate the two plans' outstanding agreement.

4 Conclusion

In this paper, we proposed a new scheme for approximating solutions of hyperbolic balance laws by combining the UCWENO scheme with the subtraction method. The proposed scheme is third-order accurate that shapes a new unstaggered version of the original CWENO reconstruction, while preserving a non-oscillatory characteristic. The UCWENO scheme blends effectively with the subtraction method and yields a procedure that satisfies the well-balanced property at the discrete level. In particular the resulting method ensures the lake at rest property of shallow water equations without any additional treatment. The proposed scheme strikes a balance between simplicity, efficacy, and accuracy. It avoids the heavy process of solving Riemann problems at cell interfaces, and employs the subtraction method to achieve a well-balanced property without the need for any particular discretization of the source term. Comparisons with other schemes on benchmark problems demonstrate that the proposed scheme reduces effectively numerical oscillations while successfully capturing steady states at the discrete level. Future work will concentrate on extending the scheme to account for frictional effects, dry and wet phases on irregular topographies, and also Coriolis forces. Two-dimensional extensions of the scheme are currently under investigations.

Declarations

- Funding: The second author received funds from LAU-PIRF project number I0015.

References

[1] Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes iii. Journal of Computational Physics **71**, 231–303 (1987)

[2] Liu, Y., Lu, J., Shu, C.W.: An essentially oscillation-free discontinuous galerkin method for hyperbolic systems. SIAM Journal on Scientific Computing **44**(1), 230–259 (2022)

[3] Delis, A.I., Katsaounis, T.: Relaxation schemes for the shallow water equations. Int. J. Numer. Meth. Fluids **41**, 695–719 (2003)

[4] Jiang, G.S., Levy, D., Lin, C.T., Osher, S., Tadmor, E.: High-resolution non-oscillatory central schemes with non-staggered grids for hyperbolic conservation laws. SIAM J. Numer. Anal. **35**(6), 2147–2168 (1998)

[5] Touma, R., Malaeb, E., Klingenberg, C.: Combining a central scheme with the subtraction method for shallow water equations. International Journal of Computational Methods, 2350035 (2023)

[6] Li, G., Caleffi, V., Gao, J.: High-order well-balanced central weno scheme for pre-balanced shallow water equations. Computers & Fluids **99**, 182–189 (2014)

[7] Touma, R., Klingenberg, C.: Well-balanced central finite volume methods for the ripa system. Applied Numerical Mathematics **97**, 42–68 (2015)

[8] Nessyahu, H., Tadmor, E.: Non-oscillatory central differencing for hyperbolic conservation laws. Journal of Computational Physics **87**(2), 408–463 (1990)

[9] Kanbar, F., Touma, R., Klingenberg, C.: Well-balanced central schemes for the one and two-dimensional euler systems with gravity. Applied Numerical Mathematics **156**, 608–626 (2020)

[10] Shu, C.W.: Numerical experiments on the accuracy of eno and modified eno schemes. Journal of Scientific Computing **5**(2), 127–149 (1990)

[11] Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. Journal of Computational Physics **115**, 200–212 (1994)

[12] Jiang, G.S., Shu, C.W.: Efficient implementation of weighted eno schemes. Journal of Computational Physics **126**(1), 202–228 (1996)

[13] Bianco, F., Puppo, G., Russo, G.: High-order central schemes for hyperbolic systems of conservation laws. SIAM Journal on Scientific Computing **21**(1), 294–322 (1999)

[14] Levy, D., Puppo, G., Russo, G.: Central weno schemes for hyperbolic systems of conservation laws. ESAIM **33**(3), 547–571 (1999)

[15] Levy, D., Puppo, G., Russo, G.: Compact central weno schemes for multidimensional conservation laws. SIAM Journal on Scientific Computing **22**(2), 656–672 (2000)

[16] Levy, D., Puppo, G., Russo, G.: A third order central weno scheme for 2d conservation laws. Applied Numerical Mathematics **33**(1-4), 415–421 (2000)

[17] Adebayo, E.M., Tsoutsanis, P., Jenkins, K.W.: Application of central-weighted essentially non-oscillatory finite-volume interface-capturing schemes for modeling cavitation induced by an underwater explosion. Fluids **9**(2), 33 (2024)

[18] Guo, W., Chen, Z., Qian, S., Li, G., Niu, Q.: A new well-balanced finite volume cweno scheme for shallow water equations over bottom topography. Advances in Applied Mathematics and Mechanics **15**(6), 1515–1539 (2023)

[19] Tsoutsanis, P., Dumbser, M.: Arbitrary high order central non-oscillatory schemes on mixed-element unstructured meshes. Computers & Fluids **225**, 104961 (2021)

[20] Caleffi, V., Valiani, A., Bernini, A.: High-order balanced cweno scheme for movable bed shallow water equations. Advances in Water Resources **30**(4), 730–741 (2007)

[21] Berberich, J.P., Chandrashekar, P., Klingenberg, C.: High order well-balanced finite volume methods for multi-dimensional systems of hyperbolic balance laws. Computers & Fluids **219**, 104858 (2021)

[22] Geyer, A., Quirchmayr, R.: Shallow water equations for equatorial tsunami waves. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **376**(2111), 20170100 (2018)

[23] Courtier, P., Geleyn, J.F.: A global numerical weather prediction model with variable resolution: Application to the shallow-water equations. Quarterly Journal of the Royal Meteorological Society **114**(483), 1321–1346 (1988)

[24] Cioffi, F., Gallerno, F.: Multi-objective analysis of dam release flows in rivers downstream from hydropower reservoirs. Applied Mathematical Modelling **36**(7), 2868–2889 (2012)

[25] Toro, E.F.: Shock-Capturing Methods for Free-Surface Shallow Flow. Wiley, New York (2001)

[26] Touma, R.: Central unstaggered finite volume schemes for hyperbolic systems: Applications to unsteady shallow water equations. Applied Mathematics and Computation **213**, 47–59 (2009)

[27] Caleffi, V., Valiani, A., Bernini, A.: Fourth-order balanced source term treatment in central weno schemes for shallow water equations. Journal of Computational Physics **218**(1), 228–245 (2006)

[28] Rogers, B.D., Borthwick, A.G.L., Taylor, P.H.: Mathematical balancing of flux gradient and source terms prior to using roe's approximate riemann solver. Journal of Computational Physics **192**, 422–451 (2003)

[29] Xing, Y., Shu, C.W.: High order well-balnced finite volume weno schemes and discontinuous galerkin methods for a class of hyperbolic systems with source terms. Journal of Computational Physics **214**, 567–598 (2006)

[30] Caleffi, V., Valiani, A.: Well-balanced bottom discontinuities treatment for high-order shallow water equations weno scheme. Journal of Engineering Mechanics **135**(7), 684–696 (2009)