

Article

Cross-Scale Modeling of Shallow Water Flows in Coastal Areas with an Improved Local Time-Stepping Method

Guilin Liu ¹, Tao Ji ¹ , Guoxiang Wu ^{1,*} , Hao Tian ^{2,*} and Pubing Yu ³¹ College of Engineering, Ocean University of China, Qingdao 266100, China² School of Mathematical Sciences, Ocean University of China, Qingdao 266100, China³ Zhejiang Institute of Hydraulics and Estuary, Hangzhou 310000, China

* Correspondence: guoxiang@ouc.edu.cn (G.W.); haot@ouc.edu.cn (H.T.)

Abstract: A shallow water equations-based model with an improved local time-stepping (LTS) scheme is developed for modeling coastal hydrodynamics across multiple scales, from large areas to detailed local regions. To enhance the stability of the shallow water model for long-duration simulations and at larger LTS gradings, a prediction-correction method using a single-layer interface that couples coarse and fine time discretizations is adopted. The proposed scheme improves computational efficiency with an acceptable additional computational burden and ensures accurate conservation of time truncation errors in a discrete sense. The model performance is verified with respect to conservation and computational efficiency through two idealized tests: the spreading of a drop of shallow water and a tidal flat/channel system. The results of both tests demonstrate that the improved LTS scheme maintains precision as the LTS grading increases, preserves conservation properties, and significantly improves computational efficiency with a speedup ratio of up to 2.615. Furthermore, we applied the LTS scheme to simulate tides at grid scales of 40,000 m to 200 m for a portion of the Northwest Pacific. The proposed model shows promise for modeling cross-scale hydrodynamics in complex coastal and ocean engineering problems.

Keywords: shallow water model; local time-stepping; cross-scale; computational efficiency



Citation: Liu, G.; Ji, T.; Wu, G.; Tian, H.; Yu, P. Cross-Scale Modeling of Shallow Water Flows in Coastal Areas with an Improved Local Time-Stepping Method. *J. Mar. Sci. Eng.* **2024**, *12*, 1065. <https://doi.org/10.3390/jmse12071065>

Received: 9 May 2024

Revised: 20 June 2024

Accepted: 21 June 2024

Published: 25 June 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

With the continuous advancements in numerical methods and parallel technologies, cross-scale ocean models [1–4] have undergone significant developments. These models offer the ability to simulate multiple resolutions by manipulating grid sizes on a large scale. In many applications of ocean modeling, the practice is to employ smaller cells in areas where higher precision is required, while larger cells are used in other regions to optimize computational efficiency and cost-effectiveness. Time-stepping methods typically favor explicit global time-stepping (GTS) schemes due to their inherent parallelism and lower programming complexities. However, GTS schemes have a drawback—they require determining time steps that must satisfy the Courant–Friedrichs–Lewy (CFL) condition, resulting in substantial discrepancies between the local time steps of large cells and the global minimum time step. These discrepancies result in wasted computational resources, thus affecting the computational efficiency of cross-scale ocean models. Local time-stepping (LTS) schemes overcome these constraints by allowing cells in different regions to use their own local time steps rather than the global minimum time step.

In pursuit of enhancing computational efficiency in solving one-dimensional (1D) scalar conservation equations, the LTS algorithm was originally proposed by Osher and Sanders [5]. The proposed scheme achieves first-order accuracy in both space and time. Kleb et al. [6] presented a temporal adaptive algorithm for the time integration of the 2D Euler or Navier–Stokes equations. Dawson [7] proposed an extension of the high-resolution scheme of the finite element method for the advection equations and conducted a detailed analysis of the

second-order temporal format of the 1D scalar conservation laws in Dawson and Kirby [8]. The results demonstrate that the LTS scheme is highly competitive regarding temporal accuracy. In addition, Fumeaux [9], Muller and Stiriba [10], and Tan et al. [11] investigated LTS schemes for 2D and 3D problems and corrected numerical fluxes by interpolating variable predictions during flux solving. Trahan and Dawson [12] also studied LTS schemes for shallow water equations, using a Runge–Kutta discontinuous Galerkin finite element method for spatial discretization. The LTS scheme was second-order accurate in time away from local time-stepping interfaces and first-order accurate at the interfaces. Furthermore, higher-order LTS schemes have also been studied, and Krivodonova [13] introduced an explicit LTS scheme based on higher-order Runge–Kutta (RK) for solving conservation laws and wave propagation. Hoang et al. [14] developed second and third-order accurate explicit LTS schemes for shallow water equations. The proposed LTS schemes are of predictor–corrector type, constructed using Taylor series expansion and the stability-preserving Runge–Kutta (SSP-RK) stepping algorithm for constructing predictors, enabling the algorithm to achieve second and third-order temporal accuracy across the entire domain. Dazzi et al. [15]; Hu et al. [16] also combined GPU (Graphics Processing Unit)-accelerated algorithms with LTS algorithms to make the model computation more efficient.

Kramer and Jozsa [17] have introduced a simple but robust LTS algorithm based on the finite volume method for the 2D shallow water equations (SWEs). And the interface fluxes are corrected according to the correction algorithm of Tan et al. [11]. Sanders [18] applied the LTS method to an explicit Godunov-type shallow water model to improve operational efficiency. Hu et al. [19,20] improved Sanders’ algorithm by modifying the LTS levels, which are the localized gradations of time steps, at the wet/dry front and dynamic/static front and applied the improved model to calculating tides and hydro-sediment-morphodynamic processes. This represents an enhancement for numerical simulations of 2D shallow water flow. Yang et al. [21] studied the implementation of the LTS algorithm on a 2D shallow hydraulic model and discussed the influencing factors and adaptive conditions of the algorithm. Dawson et al. [22,23], employing the Runge–Kutta discontinuous Galerkin finite element method, investigated the LTS algorithm of the 2D SWEs. These findings were subsequently applied to simulate tidal flows and storm surges. Lilly et al. [4] developed a fourth-order algorithm based on Hoang et al. and achieved good results in storm surge simulation using the Model for Prediction Across Scales–Ocean (MPAS–O). In ocean modeling, a simple but stable and efficient model with guaranteed accuracy is one of our main concerns. Kramer and Jozsa [17] recommended that the maximum LTS level should be less than three and two neighboring cells’ LTS level should not differ by more than one. Sanders [18] only recommended setting the LTS level to a maximum of four to ensure the stability of the calculation. Hu et al. [20] proposed a slight but useful modification to use larger LTS levels but uncorrected for interface layer fluxes with significant differences in LTS levels. Yang et al. [21] did not mention the method of correction in their article.

In this study, we develop a 2D shallow water model using a conservative, explicit LTS algorithm intended for cross-scale ocean hydrodynamic modeling applications. The time step is determined by the local CFL condition, allowing for a combination of local time and spatial refinement instead of relying on the global CFL condition. The model combined LTS scheme for the shallow water equations proposed by Hu et al. [20]; Kramer et al. [17]; Krivodonova [13]; and Sanders [18] with flux correction theory based on the Taylor series expansion proposed by Hoang et al. [14], and applies the newly designed LTS scheme to the calculation of cross-scale 2D shallow water equations model. It is important to note that our prediction scheme yields the same results as the first-order prediction scheme employed by Kramer et al. [17]. However, based on the specific characteristics of the spatial discretization scheme and the solver scheme used in this paper, we chose the corrector defined by Hoang et al. [14] to ensure the accurate conservation of the fluxes. Moreover, with respect to the equation-solving and time-stepping methods used in this paper, we introduce a single interface to couple the intermediate time increments of two distinct LTS levels. This streamlines the algorithm’s complexity but still maintains accuracy and

conservation. Additionally, the governing equations are solved on unstructured meshes by the finite volume method. Numerical fluxes between cells are evaluated using Roe's approximate Riemann solver [24,25]. And the bottom slope term and wet/dry boundaries treatment by using the method proposed by Yu et al. [26]. Numerical tests demonstrate that the model significantly improves computational efficiency and effectively balances fluxes between regions with varying LTS grading. We also attempted to establish an offshore tidal flow model to verify its practical performance in ocean engineering applications.

This paper is structured as follows: Section 2 introduces the governing equations, finite volume discretization method, and details model processing. We also present the mathematical formulas for the improved LTS algorithm. Section 3 offers two ideal tests to verify the model's performance by simulating the spreading of a drop of shallow water and tidal flats' overflow and drainage processes. Section 4 demonstrates the application of the established cross-scale 2D shallow water model to simulate the currents in the Bohai Sea, Yellow Sea, and East China Sea from the Zhoushan Archipelago to the external sea areas and compares the results with actual measurement data. Finally, Sections 5 and 6 conclude the paper and provide further discussions.

2. Model Formulations

2.1. Governing Equations

Shallow water models based on unstructured meshes and finite volume method [27,28] have been well developed and widely used to simulate coastal processes, including tides, storm surges, and sediment transport in estuaries and nearshore regions [26,29–32]. The 2D shallow water governing equations are written in matrix form as follows:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S} \quad (1)$$

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + gh^2/2 \\ huv \end{bmatrix}, \mathbf{G} = \begin{bmatrix} hv \\ huv \\ hu^2 + gh^2/2 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ -gh(S_{ox} + S_{fx}) + F_{bx} \\ -gh(S_{oy} + S_{fy}) + F_{by} \end{bmatrix} \quad (2)$$

$\mathbf{U}$ represents the conserved physical vector; $\mathbf{F}$, $\mathbf{G}$ denote the convective fluxes in the x and y directions, respectively; $\mathbf{S}$ stands for the source term; h is the water depth as depicted in Figure 1a; The variables u , v correspond to the depth-averaged velocities in the x and y directions, respectively; g is the gravitational acceleration; $S_{ox} = \partial z_b / \partial x$, $S_{oy} = \partial z_b / \partial y$ indicate the bottom slope terms in the x and y directions, respectively; $S_{fx} = n^2 u \sqrt{u^2 + v^2} / h^{4/3}$, $S_{fy} = n^2 v \sqrt{u^2 + v^2} / h^{4/3}$ are the bottom friction terms in the x and y directions, respectively, with n representing the Manning coefficient ($s/[m(1/3)]$); F_{bx} , F_{by} denote the Coriolis force terms, for the northern hemisphere $F_{bx} = 2\omega \sin \phi \cdot v$, $F_{by} = -2\omega \sin \phi \cdot u$ ($\omega = 7.29 \times 10^{-5} \text{ rad/s}$, ϕ is the latitude).

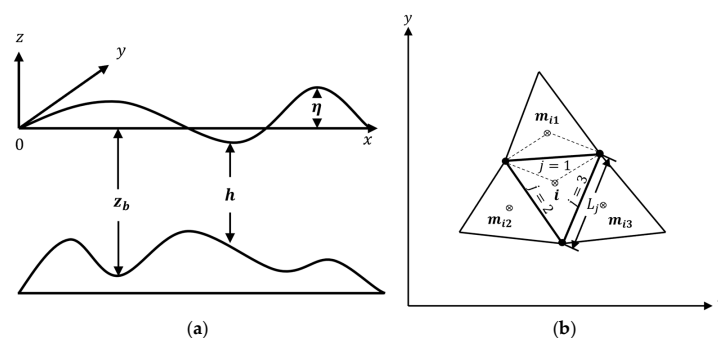


Figure 1. (a) Non-constant flow over complex topography. η is the water surface elevation, z_b is the bottom bed elevation, and $h = \eta - z_b$ is the water depth. (b) Unstructured triangular mesh cells. L is the length of the grid edge and m is the LTS level of neighboring cells (see Section 2.3) in counterclockwise order.

Equation (1) can be written in compact form:

$$\frac{\partial \mathbf{U}}{\partial t} + (\nabla \bullet \mathbf{E}) = \mathbf{S} \quad (3)$$

where $\mathbf{E} = (\mathbf{F}(\mathbf{U}), \mathbf{G}(\mathbf{U}))$ is the numerical flux.

2.2. Finite Volume Discretization and Flux Solving

To improve the adaptability to complex areas, this paper utilizes unstructured triangular meshes to partition the computational domain. A non-boundary triangular mesh has three nodes, three edges, and three adjacent cells (Figure 1b). An edge has two nodes and two adjacent cells. The elevation of the bottom bed is defined at the nodes and the other physical variables (h, u, v) are defined at the center of the cells.

Using the finite volume method, Equation (3) is integrated over the control volume:

$$\iint_{\Omega} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \iint_{\Omega} (\nabla \cdot \mathbf{E}) d\Omega = \iint_{\Omega} \mathbf{S} d\Omega \quad (4)$$

Utilizing the rotational invariance of the Euler equations, the governing equations are locally rotated within each computational cell to coordinate systems that are orthogonal to the cell boundaries for solving. According to the rotational invariance of Euler equations, the discretization of cell i can be derived as follows:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t^n}{A_i} \sum_{j=1}^M T^{-1} \bar{\mathbf{E}}_{nj}(\mathbf{U}_i^n) L_j + \mathbf{S}_i(\mathbf{U}_i^n) \quad (5)$$

where $\mathbf{U}_i^{n+1}, \mathbf{U}_i^n$ represent two consecutive time steps; i is the cell number, $i = 1, 2, 3, \dots, N$, N is the total number of cells; j is the cell edge number, $j = 1, 2, 3, \dots, M$, and M is the total number of cell edges; A_i is the area of the i -th control volume; T is the rotation matrix; $\bar{\mathbf{E}}_{nj}(\mathbf{U}_i^n)$ is the numerical flux at the j -th edge of the i -th cell; L_j is the edge length at the j -th edge of the i -th cell.

The solution of the numerical flux $\bar{\mathbf{E}}_{nj}$ is the core of solving Equation (5). This paper uses Roe's approximate Riemann solver [24] to compute the numerical fluxes at the cell interfaces:

$$\bar{\mathbf{E}}(\mathbf{U}) = \frac{1}{2}(\mathbf{E}(\mathbf{U}_L) + \mathbf{E}(\mathbf{U}_R)) - \frac{1}{2} \sum_{j=1}^3 \alpha'_j |\tilde{\lambda}_j| \gamma_j \quad (6)$$

where $\mathbf{E}(\mathbf{U}_L), \mathbf{E}(\mathbf{U}_R)$ are the fluxes at the left and right sides of the edge interface, and $\gamma_j, \tilde{\lambda}_j$ are the eigenvalues and eigenvectors of the modified Jacobian matrix [24], respectively. When encountering extreme situations such as the dry bed dam-break problem, entropy correction is necessary to ensure that the Roe scheme satisfies the entropy condition and produces accurate results. In addition, this paper adopts the bottom slope and wet/dry boundary treatments proposed by Yu et al. [26], ensuring the accuracy of the spatial scheme and conserving the interface flux.

2.3. Local Time-Stepping (LTS) Scheme

The explicit time discretization scheme is usually used for nonlinear discrete systems because of its simplicity and natural parallelism. However, the explicit method is inefficient for cross-scale models due to the global time step. Therefore, we propose an improved LTS scheme that enables the cells to advance based on their local time steps.

The first-order explicit time discretization format is given by Equation (5), which can be simplified to

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \Delta t^n \bar{\mathbf{F}}(\mathbf{U}_i^n) \quad (7)$$

for $i = 1, 2, 3, \dots, N$ and $n = 1, 2, 3, \dots, N_t$, where $\bar{\mathbf{F}}(\mathbf{U}_i^n) = \mathbf{S}_i(\mathbf{U}_i^n) - (\sum_{j=1}^M T^{-1} \bar{\mathbf{E}}_{nj}(\mathbf{U}_i^n) L_j) / A_i$.

For simplicity, we display the “coarse” and “fine” time increments of part of the computational domain as follows (Figure 2a).

$$[t^n, t^{n+1}) = \cup_{k=0}^{M_t-1} [t^{n,k}, t^{n,k+1}) \quad (8)$$

where $t^{n,0} = t^n$, and $t^{n,k+1} = t^{n,k} + \Delta t^n / M_t$, for all $k = 0, \dots, M_t - 1$.

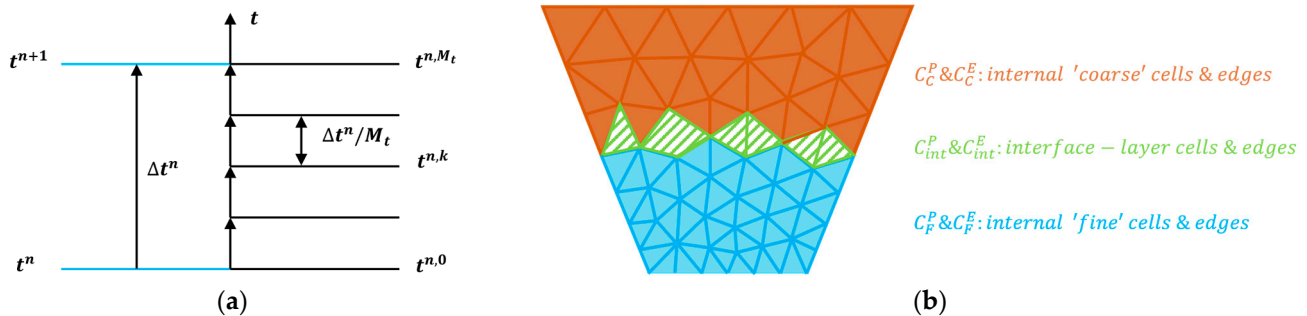


Figure 2. (a) Coarse and fine time increments, Δt^n is the local time step of the coarse time increments and $\Delta t^n / M_t$ is the local time step of the fine time increments. (b) An illustration of different types of cells and edges with either coarse or fine time increments.

To define the LTS scheme for the discrete system obtained from the shallow water model, we refer to Hoang et al. [14] and further partition the cells in the local computational domain, which contain different time increments, into three classes (Figure 2b).

The cells in this set belong to the following: C_C^E —“fine” cells, C_{int}^P —interface cells, C_C^P —“coarse” cells. The cell edges in this set belong to the following: C_C^E —“fine” cell edges; C_C^P —interface cell edges; C_C^E —“coarse” cell edges. Notably, for the cells sharing a boundary with the interface cells, the shared edges belong to the finer cells, i.e., the shared edges of C_C^E and C_{int}^P belong to C_C^E , and the shared edges of C_{int}^P and C_C^P belong to C_{int}^P . In this paper, we initially apply the algorithm to the first-order forward Euler method. Only one interface is introduced to couple the “fine” and “coarse” regions. Since the time-stepping method consists of only one prediction and one correction process, we expect the predicted value of the interface cells to affect only the “fine” cells.

The algorithm begins by hierarchizing all cells and identifying the interface layers, aiming to maximize the computational efficiency of each cell while ensuring compliance with the local CFL conditions. Then evaluate numerical flux and advance for different time increments until a complete increment ends. To obtain the advancement of intermediate time levels in the interface layers, we use the predictor proposed by Hoang et al. [14]. Once the predicted values are obtained, we can advance the values at the intermediate time levels for the coarse grids in the interface. Finally, a correction ensures the conservation of the flux. The following section presents the specific LTS algorithm based on this method.

Based on the local CFL conditions, calculate the maximum local time step Δt_i that each cell can be satisfied [20].

$$\begin{cases} \Delta t_i = Cr \min_{j=1,3} \left(\frac{r_{ij}}{\sqrt{u_{ij}^2 + v_{ij}^2} + \sqrt{gh_i}} \right) & (i = 1, 2, 3, \dots, N) \quad h_i \geq h_{cwd} \\ \Delta t_i = \min \left[\max_{h_i \geq h_{cwd}, i=1, N} (\Delta t_i), Cr \min_{j=1,3} \left(\frac{r_{ij}}{\sqrt{u_{ij}^2 + v_{ij}^2} + \sqrt{gh_i}} \right) \right] & 0 < h_i \leq h_{cwd} \end{cases} \quad (9)$$

for $i = 1, 2, 3, \dots, N$ and Cr is the Courant number, typically set to 0.9; r_{ij} is the distance from the center of the i -th cell to its j -th edge; u_{ij}, v_{ij} are the flow velocities in the normal local coordinate system to the j -th edge of the i -th cell; h_i is the water depth of the i -th cell; h_{cwd} is the critical water depth, which is determined according to the actual situation. It is worth noting that when the h of a cell is less than the h_{cwd} , the cell is considered a dry cell,

and its maximum local time step can be infinitely large. However, considering the actual situation, it is set to the maximum value of the local time step in wet cells.

The minimum global time step can be obtained from

$$\Delta t_{\min} = \min_{i=1,N}(\Delta t_i) \quad (10)$$

Next, calculate the LTS level m_i :

$$m_i = \min \left[\text{int} \left(\frac{\lg(\Delta t_i / \Delta t_{\min})}{\lg 2} \right), m_{user} \right] \quad i = 1, 2, 3, \dots, N \quad (11)$$

m_{user} is the upper limit of the manually set LTS level, intended to restrict the maximum LTS level of the model. When $m_{user} = 0$, the model is equivalent to the GTS model.

Then, we need to modify the LTS level of each cell and calculate the LTS level at the cell edges. Sanders [18] proposed that when the LTS levels change significantly in the interface region, the LTS level of adjacent cells in the coarse region should be modified to a smaller value. Based on numerical experiments, it is optimal to modify the interface LTS level according to the maximum LTS level of C_{int}^P . At the same time, when the interface is a dry/wet front or a dynamic/static front, i.e., when the flow velocity of one side of the interface is 0 or the water depth $h_i < h_{cud}$, the interface LTS level must also be modified. For conservatism, this paper adopts the method of Hu et al. [20] to correct the particular fronts. After the modification is completed, the C_{int}^P are identified, and the LTS level m_{ij} of the cell edges is calculated.

$$m_{ij} = \min(m_{ij}, m_i) \quad i = 1, 2, 3, \dots, N \quad j = 1, 2, 3, \dots, M \quad (12)$$

Depending on the obtained LTS level of each cell, the local time step Δt_{L-i} of the cells and the time interval of a complete cycle are calculated.

$$\Delta t_{L-i} = 2^{m_i} \Delta t_{\min} \quad (13)$$

$$m_{\max} = \max_{i=1,N}(m_i) \quad (14)$$

$$\Delta T = 2^{m_{\max}} \Delta t_{\min} \quad (15)$$

Once all the LTS parameters are calculated, the numerical fluxes and physical variables can be updated through the intermediate time steps s_n , $s_n = 1, 2, 3, \dots, 2^{m_{\max}}$. When $(s_n - 1)/2^{m_{ij}}$ is an integer, i.e.,

$$\text{mod} \left(\frac{s_n - 1}{2^{m_{ij}}} \right) = 0 \quad (16)$$

update edges numerical fluxes. After the physical variables of C_{int}^P are updated to the next time cycle with a “coarse” time step, physical variables of C_F^P have been synchronized to the first cycle. We use Taylor series expansion to predict the values of intermediate time levels in C_{int}^P .

Assuming that the solution at the moment t is known, we aim to find the second-order approximation of $\mathbf{U}^{n,k}$ at the intermediate time step $t^{n,k}$, for $k = 0, \dots, M_t - 1$. Performing Taylor series expansion of $\mathbf{U}$ at t^n yields

$$\mathbf{U}(t) = \mathbf{U}(t^n) + (t - t^n) \frac{d\mathbf{U}}{dt}(t^n) + \frac{1}{2!} (t - t^n)^2 \frac{d^2\mathbf{U}}{dt^2}(t^n) + \dots, \quad (17)$$

thus, we can approximate $\mathbf{U}^{n,k}$ at by truncating Equation (17) to the second term (the first-order Taylor expansion):

$$\mathbf{U}^{n,k} = \mathbf{U}^n + \frac{k\Delta t}{M_t} \frac{d\mathbf{U}^n}{dt} \quad (18)$$

which gives a truncation error of second order in time. By

$$\frac{d\mathbf{U}^n}{dt} = \frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} \quad (19)$$

we obtain a second-order approximation of $\mathbf{U}^{n,k}$

$$\mathbf{U}^{n,k} = (1 - \alpha_k)\mathbf{U}^{n,k} + \alpha_k\mathbf{U}^{n+1} \quad (20)$$

where $\alpha_k = k/M_t$, $k = 0, \dots, M_t - 1$.

After obtaining the values of the intermediate time levels, the numerical fluxes and physical variables of the “fine” cells on the subcycle can be further calculated. We note, however, that when using first-order Taylor expansion, the result is the same as the linear interpolation of the forward Eulerian solution. The need for Taylor expansion becomes apparent when extended to higher orders. When $(s_n - 1)/2^{m_i}$ is an integer, i.e.,

$$\text{mod}\left(\frac{s_n - 1}{2^{m_i}}\right) = 0 \quad (21)$$

update cells’ physical variables. And when

$$m_i - l_0(s_n) < 0 \quad (22)$$

$$l_0(s_n) = \sum_{k=0, m_{user}} [(k+1) \bullet \text{temp}], \text{ where temp} = \begin{cases} 1, \text{mod}(s_n, 2^k) = 0 \\ 0, \text{mod}(s_n, 2^k) \neq 0 \end{cases} \quad (23)$$

correct the physical variables of C_{int}^P using the interface fluxes, where l_0 is the threshold used for updating. Referring to the treatment described by Osher and Sanders [5] and Hoang et al. [14], the specific correction algorithm is:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t^n}{M_t} \sum_{k=0}^{M_t-1} \bar{\mathbf{F}}(\mathbf{U}_i^{n,k}), \forall i \in C_{\text{int}}^P \quad (24)$$

In order to minimize computation time, $\bar{\mathbf{F}}(\mathbf{U}_i^{n,k})$ does not need to be recalculated but is stored as shared information that is called upon when corrections are made. Consequently, the correction process incurs additional overhead compared to the original LTS method. We will delve into this in Section 3.2 for a detailed analysis. Figure 3 provides a flow chart of the numerical structure. Taking a complete cycle containing two LTS levels ($M_t = 2$) as an example, the LTS algorithm can be summarized as the following steps (Table 1).

Table 1. First-order LTS algorithm for a complete cycle containing two LTS levels.

Step 1	Calculate the LTS parameters for all cells.
Step 2	Start the first subcycle. Calculate the $\bar{\mathbf{F}}(\mathbf{U}^n)$ for all cells.
Step 3	Update solutions of the first subcycle using $\bar{\mathbf{F}}(\mathbf{U}^n)$ for all cells. End the first subcycle.
Step 4	Start the next subcycle. Calculate $\bar{\mathbf{F}}(\mathbf{U}^{n,k})$ for fine cells. Use Equation (20) to predict $\mathbf{U}_i^{n,k}$ for the intermediate time level.
Step 5	Update solutions for fine cells.
Step 6	Based on the fluxes calculated in the two subcycles ($\sum_{k=0}^{M_t-1} \bar{\mathbf{F}}(\mathbf{U}_i^{n,k})$), correct the $\mathbf{U}_i^{n+1}$ using Equation (24). End the complete cycle.

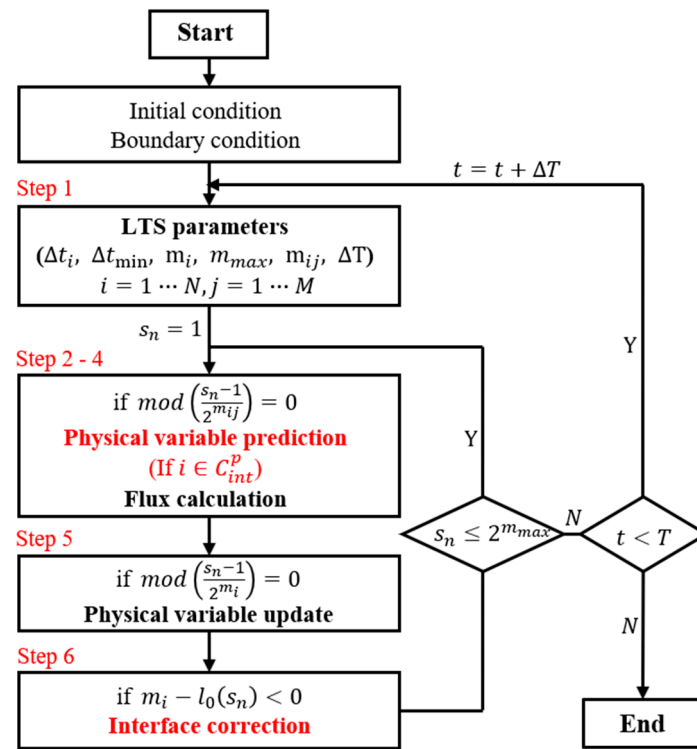


Figure 3. Numerical flowchart of the improved LTS, where “step” represents the algorithmic step in Table 1.

3. Idealized Test

To assess the stability and efficiency of the improved LTS scheme, two ideal numerical tests are conducted to demonstrate the model’s performance. In Section 3.1, we initially consider a simplified 2D model to simulate the 1D Spreading of a Drop of Shallow Water [33]. Contrasting the numerical results with the exact solution, we establish that the improved LTS scheme offers enhanced accuracy and conservation properties compared to the original scheme. In Section 3.2, a 2D test is presented, utilizing a complex scenario of a channel–flat system [34]. And we primarily delve into the advantages and limitations of the improved LTS scheme in terms of efficiency. To facilitate a more comprehensive efficiency comparison, all models are run on a PC with an Intel® Xeon® CPU E5-2640 v3 (2.60 GHz) and 64.0 GB RAM, without parallel processing.

3.1. Test 1: Spreading of a Drop of Shallow Water

Test 1 involved simulating the spreading of a parabolic-shaped water droplet to assess the capability of the LTS schemes in modeling dry bed dam-break scenarios, as well as the precision and conservation properties of the LTS models. In this test, we employed a simplified 2D model to simulate a 1D problem. The computational domain consisted of a rectangular, horizontal riverbed measuring 12 m in length and 1 m in width, with a parabolic water drop at $x = 0$ m. When the droplet suddenly breached, it led to the generation of dam-breaking water flows in two directions. This particular scenario has exact solutions. The gravitational acceleration $g = 1.0$ is taken and frictional resistance is neglected. The initial conditions are:

$$\begin{cases} h(x, t = 0) = \begin{cases} 1 - x^2 & |x| \leq 1 \\ 0 & |x| > 1 \end{cases} \\ u(x, t = 0) = 0 \end{cases} \quad (25)$$

The computational domain is discretized using 12,952 unstructured grids, with a minimum cell size of 0.02 m located at $x = 0$ m and a maximum cell size of 0.1 m at both ends of the river. A total of 6 levels of m_{user} from 0 to 5 were selected, and a total of 12 working conditions including the original LTS models and the improved LTS models were calculated. To facilitate error calculations, the minimum global time step for all working conditions is set to 0.002 s. This section primarily focuses on validating the advantages of the improved LTS scheme, while a discussion on computational efficiency is provided in Section 3.2.

Figure 4 provides comparisons between simulation results and exact solutions at $t = 1$ s, 2 s, and 3 s. In order to achieve 1D plotting, the 2D simulation results are projected onto various coordinates along the channel's centerline. Combined with the local zoomed-in images, it becomes evident that the improved LTS scheme performs exceptionally well in terms of water depth (h). The results align closely with the exact solutions and are notably consistent with the outcomes obtained using the GTS scheme. However, the results of the original LTS scheme have oscillated at $t = 1$ s and gradually deteriorated according to the advancement of time; in terms of the flow velocity (u), similarly, the improved scheme is also better than the original scheme. Figure 5 illustrates the LTS level and water depth distribution in each cell at $t = 0$ s. According to the comparison of the step-by-step output results, we observe that the regions where the oscillations begin all appear near the interface. This is due to the fact that in the initial setup of our model, each interface within the water droplet acts as a dynamic “front”. Without appropriate corrective measures, this leads to losses in mass and momentum conservation at the interfaces in coarse grid cells, thereby causing oscillations in the computations. In the improved LTS scheme, corrections made at the interface layers reduce these losses, thereby ensuring energy conservation and mitigating oscillations.

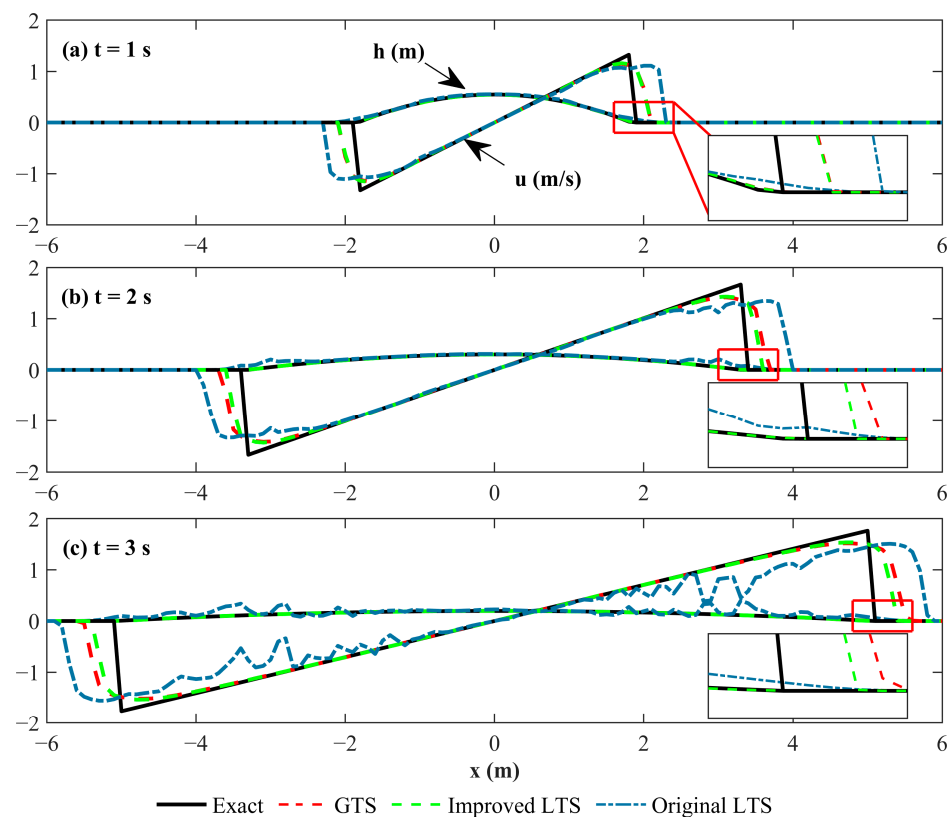


Figure 4. Comparison between the exact solutions and simulation results at $t = 1$ s, 2 s, 3 s. Each figure includes a detailed zoomed-in image, indicated by a red box in the lower right corner. Note that in the comparison of h , the improved LTS results override the GTS ($m_{user} = 0$) results.

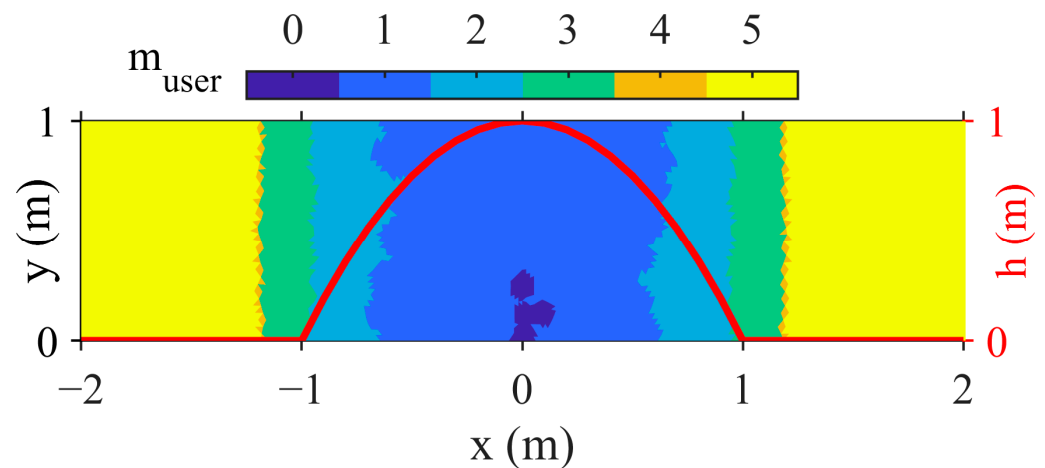


Figure 5. Distribution of LTS levels and water depths in each cell at $t = 0$ s. Note that the left scale is the river width, and the right is the initial water depth.

To further quantify the performance of the improved LTS scheme in terms of accuracy and conservation, the root mean square error (RMSE) between the results of each working condition and the exact solutions, as well as the mean global relative error (ϵ) in water mass, are calculated using the following formulas:

$$RMSE_i = \sqrt{\frac{1}{N} \sum_{j=1}^N [f_{ij} - f'_j]^2} \quad (26)$$

$$\epsilon = \frac{1}{K} \sum_{i=1}^K \frac{|Vol(t_i) - Vol(t_{i-1}) + \delta Vol|}{Vol(t_{i-1})} \quad (27)$$

where f_{ij} is the simulated value, f'_j is the exact solution, and N is the total number of comparison grids. $Vol(t_i)$ and $Vol(t_{i-1})$ represent the total volume of water within the computational domain at time instances t_i and t_{i-1} , respectively. δVol denotes the net volume passing through the boundaries between the time instances, and K represents the total number of time intervals considered. Notably, $Vol(t_0)$ corresponds to the total volume of water within the computational domain at the initial time instance.

The RMSEs in water depth and velocity are depicted in Figure 6. As previously analyzed, it is evident that the improved LTS scheme ($m_{user} = 1 - 5$) does not exhibit decreased computational accuracy compared to the GTS scheme ($m_{user} = 0$). The RMSEs in water depth (h) remain in the order of 10^{-5} to 10^{-4} . In terms of velocity (u), while accuracy experiences a slight decline with the progression of time, the improved LTS scheme continues to demonstrate its advantages. Conversely, the original LTS scheme shows a sharp decline in computational accuracy at $m_{user} = 2$, deteriorating further with higher values of m_{user} .

Figure 7 illustrates the mean global relative error in water mass for each working condition. The global water mass error is computed as an average across the total number of time steps, as opposed to assessing the error at individual time intervals. Notably, the original LTS scheme results in a substantial loss of mass, scaling up to four orders of magnitude with increasing m_{user} . In contrast, the improved scheme maintains a stable conservation of mass, with errors in the range of approximately 10^{-6} to 10^{-5} .

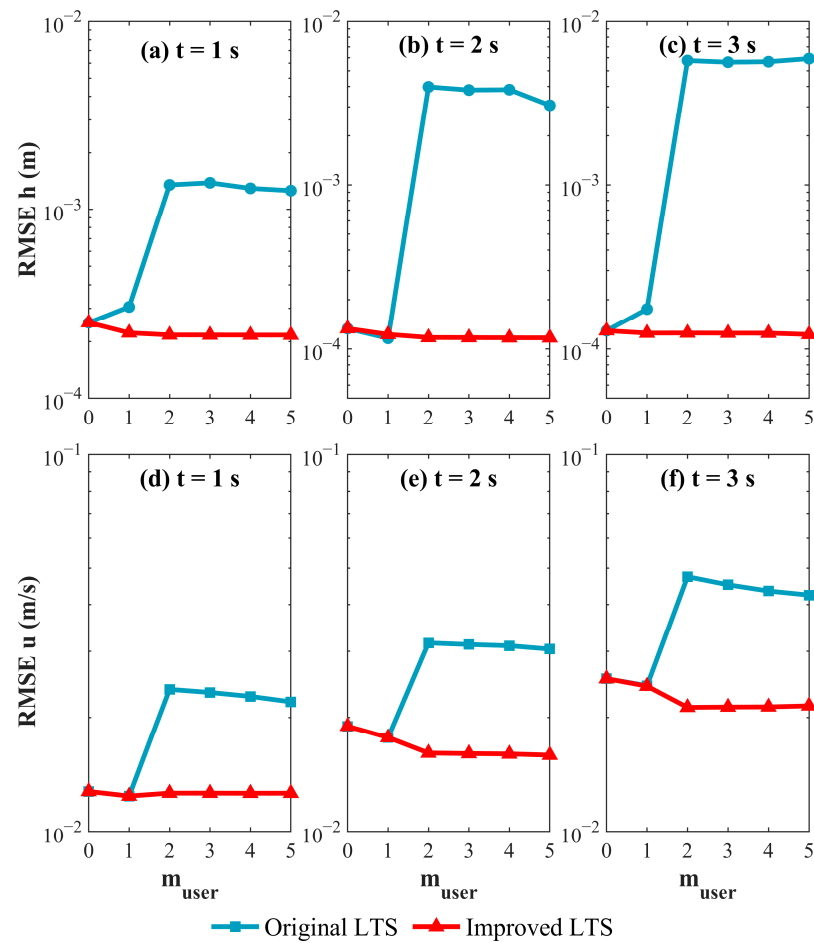


Figure 6. RMSE statistics for improved LTS and original LTS: (a–c) are the water depth RMSEs and (d–f) are the flow velocity RMSEs.

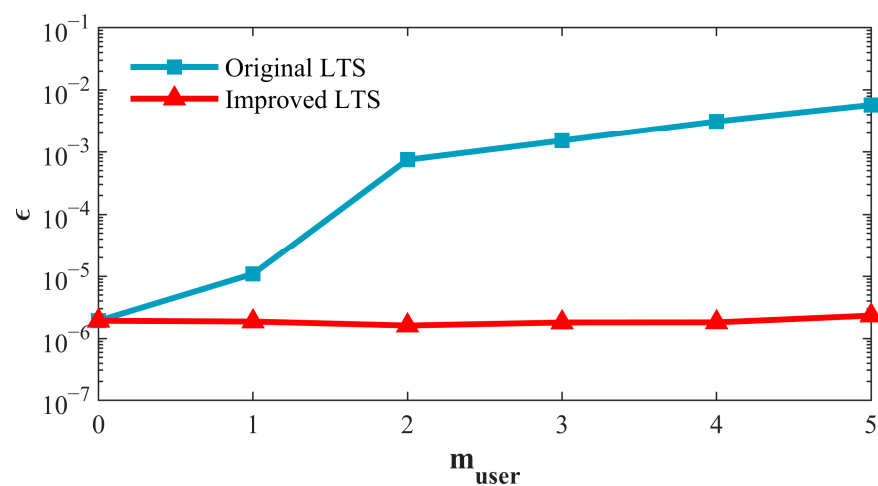


Figure 7. The m_{user} -series of the mean global relative error in water mass errors for original LTS and improved LTS.

3.2. Test 2: Tidal Flat/Channel System

In this test, we focus on the performance and limitations of the improved LTS scheme in terms of computational efficiency. The ideal test is a tidal process simulation on a conical tidal flat (Figure 8a). The simulation area is a rectangular area with a size of 4000 m $\times$ 4000 m, and a conical tidal flat is placed in the middle. The radius of the tidal flat is $r = 1000$ m, and the elevation of the inner surface of the tidal flat is proportional to the

tidal flat radius, set as $z = 0.001 r$. The outermost bed height of the tidal flat is 1 m, set as the land boundary, without flux exchange with the outer boundary cells. The tidal flat is cut by a straight channel connecting the inner and outer waters. The channel width is 20 m, and the elevation of the water in the channel and the outer area is -2 m. The model sets bottom friction, i.e., the global Manning coefficient $n = 0.001$. The water level boundary around the simulation area is set uniformly, with a water level amplitude of 0.8 m and a period of 12 h. The model runs for two days, and the last two tidal periods are used for comparison.

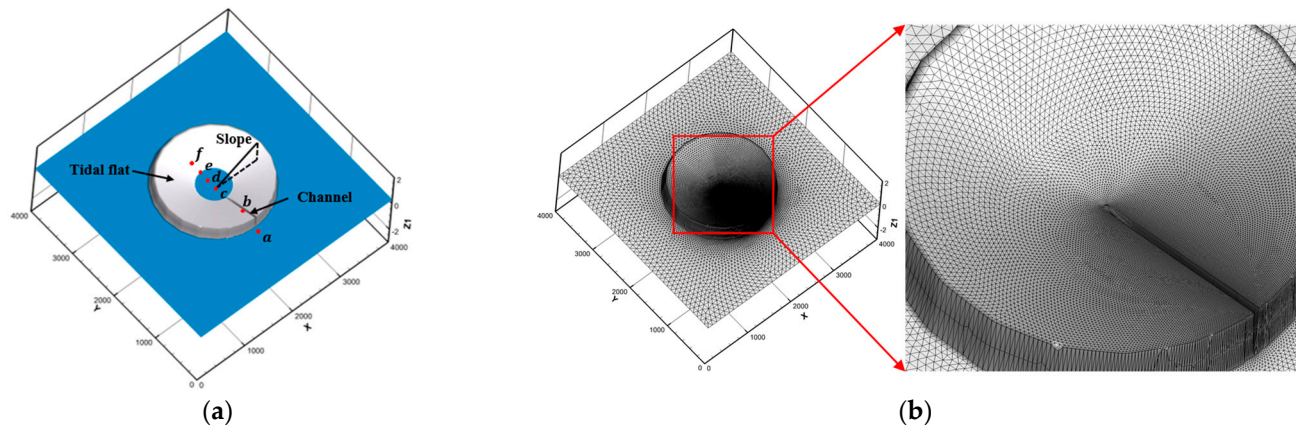


Figure 8. (a) Ideal tidal flat model topography. (b) Ideal tidal flat model calculation meshes.

The model consists of 19,257 nodes and 38,352 unstructured meshes (Figure 8b), with the minimum cell edge length being approximately 5 m and the maximum being approximately 100 m. Through a series of numerical tests, a total of six working conditions were adopted for m_{user} values from 0 to 5. Table 2 presents a comparison of the total simulation times and efficiency for various conditions in two tests. In Test 1, the minimum time step is restricted to 0.002 s, while Test 2 does not impose this limitation. Firstly, it can be seen that the total computational time decreases with the increase in the given m_{user} . Notably, the computational efficiency in Test 1 improves by a factor of 1.860 at $m_{user} = 5$ while in Test 2, it increases by 2.615 at $m_{user} = 5$. Although both LTS schemes exhibit similar acceleration efficiency across LTS levels, the original scheme requires less total computation time than the improved scheme. This difference arises due to the predictive–corrective algorithm. While the two schemes are consistent in calculating LTS parameters, and both involve almost identical memory and time overheads, the improved LTS scheme introduces additional computational costs when updating fluxes and advancing time steps.

Table 2. Ideal tests simulation time and efficiency for different working conditions. When $m_{user} = 0$, the scheme is equivalent to the GTS.

	m_{user}	0	1	2	3	4	5
Test 1	t_{min}/s	0.002	0.002	0.002	0.002	0.002	0.002
	t_{max}/s	0.002	0.004	0.008	0.016	0.032	0.064
	T/s (Original LTS)	27.563	26.531	22.438	17.688	15.594	14.469
	Speedup	-	1.039	1.228	1.558	1.768	1.905
	T/s (Improved LTS)	30.219	28.375	24.875	19.312	17.063	16.250
	Speed up	-	1.065	1.215	1.564	1.771	1.860
Test 2	t_{min}/s	0.073	0.074	0.074	0.075	0.077	0.072
	t_{max}/s	0.335	0.669	1.340	2.678	5.356	8.162
	T/h (Original LTS)	9.695	9.171	5.299	4.400	3.399	3.427
	Speedup	-	1.057	1.830	2.203	2.852	2.829
	T/h (Improved LTS)	10.204	9.636	5.899	4.742	4.011	3.902
	Speed up	-	1.059	1.730	2.152	2.544	2.615

Table 3 provides the overhead proportion (OP) of the correction module and the total number of cells proportion (CP) of each LTS level in the whole computation when $m_{user} = 5$. Since memory information from the prediction module can be shared and does not involve additional loops, we consider the cost associated with the prediction phase negligible. The 3.8% cost at $m_{user} = 0$ is a result of conditionals in the loop, and this cost increases gradually with m_{user} . Compared to the original LTS scheme, we find these costs acceptable. Additionally, it is noteworthy that computational efficiency does not linearly increase with m_{user} . Take Test 2 as an example, when m_{user} increases from 0 to 1, the computational acceleration is only 1.059, and when m_{user} increases from 4 to 5, it only improves by about 0.1. However, a significant improvement of about 0.7 occurs when m_{user} increases from 1 to 2. This discrepancy is determined by the total cell counts at various LTS levels, which is why, despite the possibility of increasing m_{user} infinitely, it is not advisable. Examining the proportion of cell counts for each LTS level in the entire computation when $m_{user} = 5$, we see that cells within LTS levels 4 and below account for approximately 96%. Therefore, increasing m_{user} further will not result in a significantly accelerated computation but would introduce unnecessary computational costs that reduce efficiency. Figure 9 shows the distribution of cells at each LTS level when $T = 3$ h, $T = 6$ h, and $T = 9$ h.

Table 3. Overhead proportion (OP) of the correction module and total number of cells proportion (CP) of each LTS level in the whole computation when $m_{user} = 5$.

Level	0	1	2	3	4	5
OP	3.80%	2.83%	3.66%	4.47%	5.19%	5.3%
CP	10.71%	5.75%	15.84%	39.82%	23.86%	4.02%

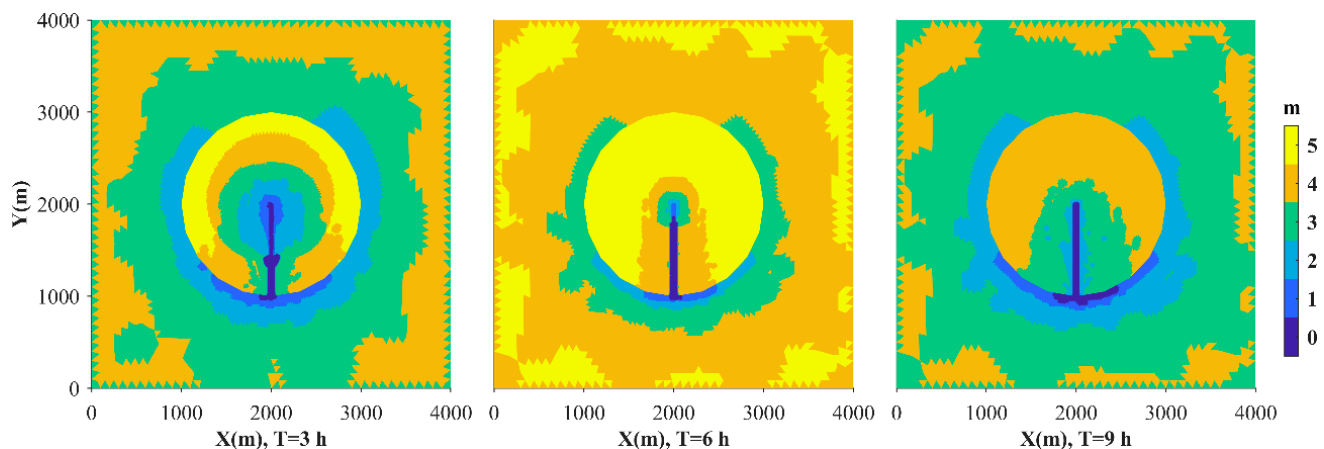


Figure 9. Distribution of cells LTS levels at $T = 0$ h, $T = 3$ h, and $T = 9$ h.

Figure 10 compares water level and v -velocity at six sites (denoted as a–f) in the model under six different LTS conditions. The selected sites are all located along the line at $x = 2000$. Site “a” is positioned at the entrance of the channel, site “b” is located midway between the entrance and the center of the channel, and site “c” is at the center of the channel. Sites “d”, “e”, and “f” are situated on the cone-shaped tidal flat at elevations of 0.2 m, 0.4 m, and 0.6 m, respectively. The results indicate that water level and velocity values for all LTS conditions match well with those of the GTS model. Tables 4 and 5 display the RMSEs of the simulated variables for each condition. The RMSEs for water level and velocity are relatively small, indicating a good consistency between the LTS and GTS models. Although the RMSE gradually increases as m_{user} increases, the intermediate error results are acceptable.

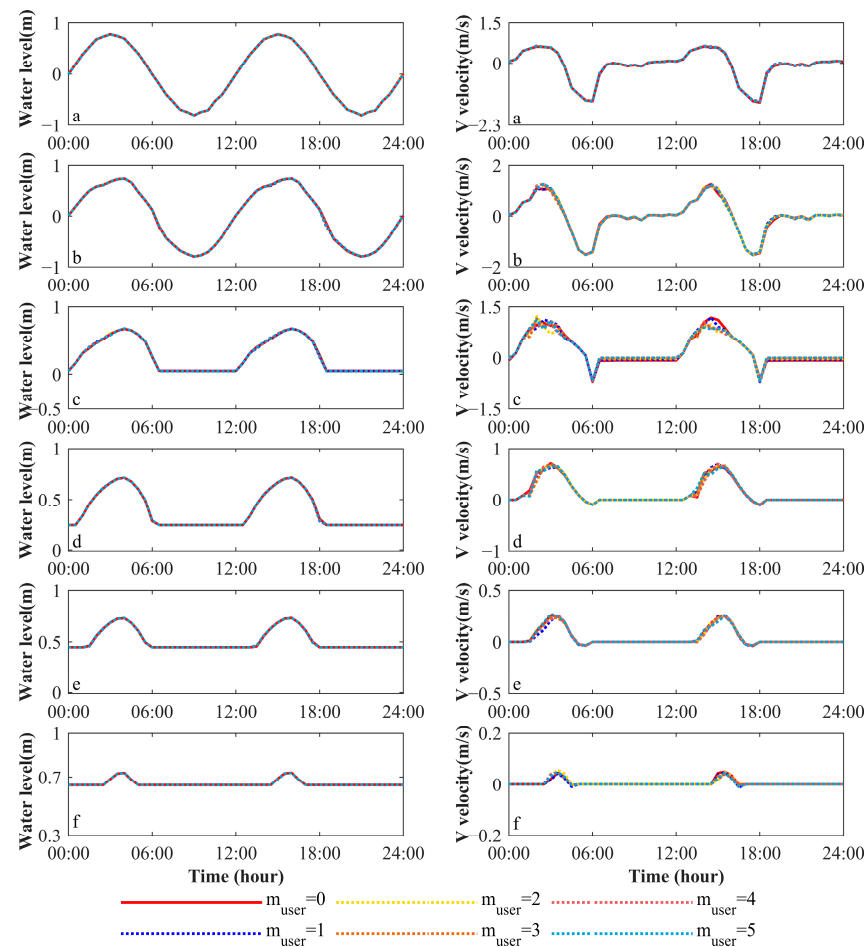


Figure 10. Time series of water level and v velocity of fringe at stations (a–f).

Table 4. RMSEs of water level for different LTS conditions at sites a–f with GTS model.

m_{user}	Output Stations					
	a	b	c	d	e	f
1	0.0031	0.0078	0.0086	0.0020	0.0018	0.0015
2	0.0043	0.0103	0.0123	0.0027	0.0025	0.0017
3	0.0052	0.0119	0.0145	0.0031	0.0030	0.0021
4	0.0064	0.0141	0.0168	0.0036	0.0034	0.0022
5	0.0082	0.0203	0.0190	0.0050	0.0042	0.0029

Table 5. RMSEs of v-velocity for different LTS conditions at sites a–f with the GTS model.

m_{user}	Output Stations					
	a	b	c	d	e	f
1	0.0350	0.0494	0.1100	0.0548	0.0271	0.0179
2	0.0466	0.0676	0.1565	0.0684	0.0313	0.0187
3	0.0542	0.0806	0.1886	0.0742	0.0349	0.0198
4	0.0645	0.0941	0.2210	0.0814	0.0375	0.0206
5	0.0890	0.1139	0.2481	0.0884	0.0403	0.0216

We also validate the conservation of the model. Figure 11 shows the mean global mass conservation relative error ε for each condition and compares it with the computational results of the original model. It can be observed that both models are similar in their

calculated errors, but the improved model demonstrates noticeably better conservation when m_{user} is increased.

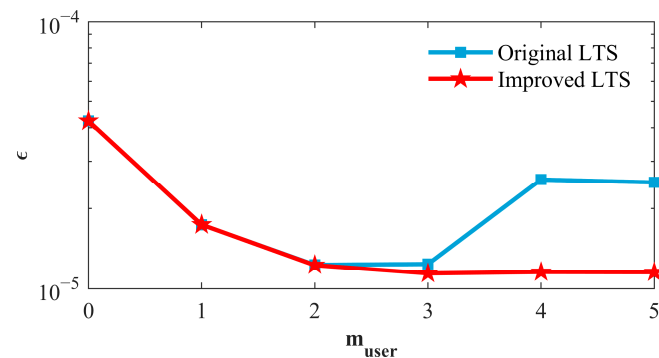


Figure 11. RMSE comparison of mean global relative error in water mass errors for original LTS and improved LTS.

4. Application to Simulating Tides in a Complex Coastal Area

In this section, to verify the capability of the cross-scale 2D shallow water model based on the improved LTS scheme for practical engineering applications, we established a selected NW Pacific tidal model containing Zhoushan Islands. Zhoushan Islands are a group of islands located in the northeastern part of Zhejiang Province, on the south side of the mouth of the Yangtze River and the outer edge of Hangzhou Bay in the East China Sea. Although there have been many studies on tides and currents in the Hangzhou Bay and Zhoushan Islands area in the past, most of them focused on the essential characteristics of tides and currents. Many numerical simulations had relatively low spatial resolution (1 km), which needs to be improved for such a complex terrain as the Zhoushan Islands area.

In this paper, the computational domain contains the entire Bohai Sea, Yellow Sea, and East China Sea, as well as the main streams of the Yangtze and Qiantang Rivers (Figure 12a). The computational domain has three open boundaries, including the Sanjiangying boundary of the Yangtze River, the Hangzhou boundary of the Qiantang River, and the outer sea boundary located near the continental shelf.

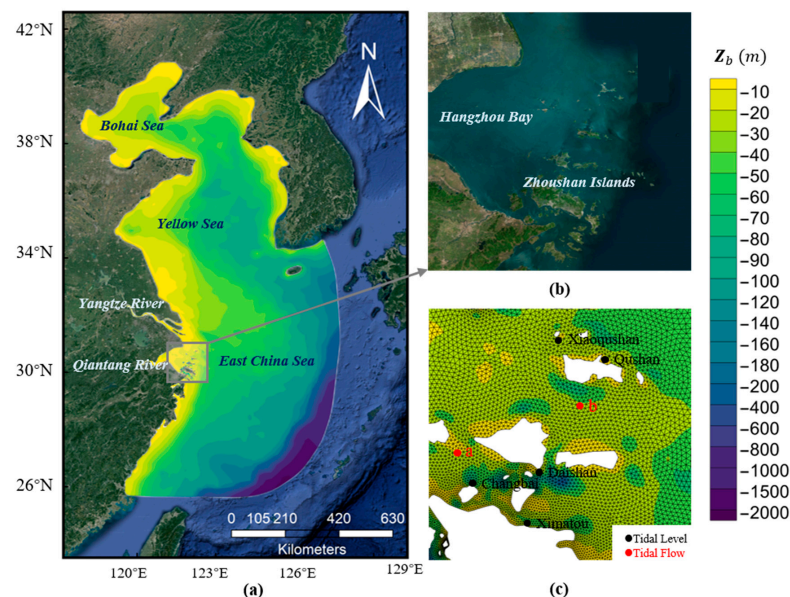


Figure 12. Calculation domain location, topography, and meshes. (a) the location of the calculation domain and the water depth topography; (b,c) the location of the validated sites and partial refined meshes.

Considering that the tidal variation in the computational domain during the simulation time is little affected by weather, the annual average runoff is taken for both open river boundaries. The outer open boundary is the hydrostatic level superimposed on the astronomical tide level, derived from the Chinese sea tidal wave model TPXO [35], which contains nine main sub-tides, Q1, P1, O1, K1, N2, M2, S2, K2, Sa. The computational domain is divided by triangular meshes, with denser meshes in the Zhejiang coastal area, especially in the Hangzhou Bay and Zhoushan Islands. A total of 75,492 cells and 39,936 nodes are arranged, the minimum cell edge length is approximately 150 m, and the maximum cell edge length is approximately 30,000 m. The coordinate projection uniformly adopts the Beijing 54 coordinate 3-degree zone (central meridian 120° E). The topography of the outer sea is based on the latest nautical charts released in China and data provided by the National Oceanic and Atmospheric Administration (NOAA) of the United States. Moreover, the recently measured terrain data are used for the Yangtze River estuary, Zhejiang coast, Hangzhou Bay, and Qiantang River estuary. Figure 12b,c show the topography and meshes of the model in areas of validation sites. The selection of the Manning coefficient affects the calculation results considerably. After several numerical experiments and debugging, the Manning coefficients are taken in the range of 0.016~0.018 for the outer sea, 0.010~0.016 for the Zhoushan Islands to Hangzhou Bay boundary, and 0.014~0.016 for the Yangtze River estuary to the open boundary. We selected eight working conditions with m_{user} ranging from 0 to 7 and simulated the tidal process from 20 May to 21 June 2019, for a total of 32 days. The minimum global time step for all conditions is set to 2 s to facilitate the calculation error.

Table 6 shows the comparison of the computational efficiency for different conditions. The LTS model can significantly improve the calculation efficiency, and the overall computation time decreases as m increases. When $m_{user} = 0$, the simulation time of the model is 36.15 h and when $m_{user} = 7$, the simulation time is only 9.23 h, which is about 3.9 times higher than the computational efficiency of the GTS model. However, it is worth noting that the simulation elapsed time of the model is 10.95 h when $m_{user} = 5$, which is longer than 10.44 h when $m_{user} = 4$. This is because the cell correction algorithm becomes more complicated with the local time step level increase. Combined with the ideal test in the previous section, this indicates that although the local time step level can be infinite, the number of cells at each time level still needs to be considered to obtain optimal computational efficiency.

Table 6. Comparison of simulation time for different working conditions of tidal simulation in Zhoushan Islands. When $m_{user} = 0$, the scheme is equivalent to the GTS.

m_{user}	0	1	2	3	4	5	6	7
t_{min}/s	2	2	2	2	2	2	2	2
t_{max}/s	2	4	8	16	32	64	128	256
Time/h	36.15	25.36	15.01	11.54	10.44	10.95	9.50	9.23
speedup	-	1.43	2.41	3.13	3.46	3.30	3.81	3.92

Figure 13 illustrates the distribution of LTS levels for cells at different time instances. At a global scale, the changes in LTS levels during the computation process remain relatively small. However, noticeable variations in LTS levels do occur in regions with grid refinement. This shows the tidal model, indicating that tidal fluctuations have minimal impact on LTS levels in deep water areas with larger grid cells. Conversely, in shallower water regions with denser grids, tidal effects trigger significant changes in LTS levels. Importantly, these changes tend to originate at the interfaces. Thus, for extended-duration simulations, the incorporation of the new LTS scheme proves essential to accurately capture these variations.

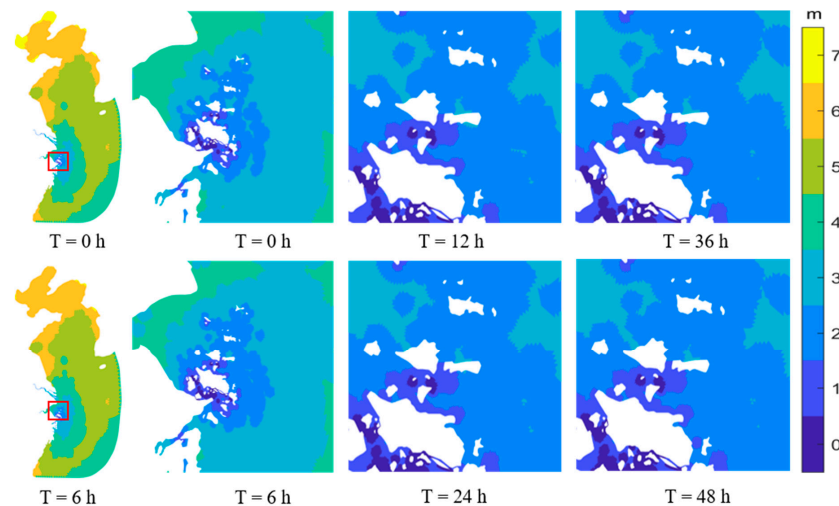


Figure 13. LTS levels distribution for the entire computational domain and the Zhoushan regional grids at these moments. Note that the location of the zoom-in localized map is marked by the red box.

Figures 14 and 15 show the time series of observed and simulated values of tide level and tidal current (velocity and direction) measurement stations in the Zhoushan Islands. Stations Daishan, Qushan, Ximatou, Xiaoqushan, and Changbai are tide level stations, while a and b are tidal current measurement stations, each measuring four sets of tidal data during spring and neap tides. To better demonstrate the agreement between the simulated and observed values while comparing the results of different time-step models, we compare the simulated results with $m_{user} = 0$ and $m_{user} = 7$ to the observed values. It can be seen that the simulated values of tide level and current match well with the observed values. Although the model achieves excellent agreement in tide level when $m_{user} = 0$ and $m_{user} = 7$ are selected, there are still differences in the current. While the correction method can reduce the effects of non-conservative algorithms, it cannot completely eliminate this effect.

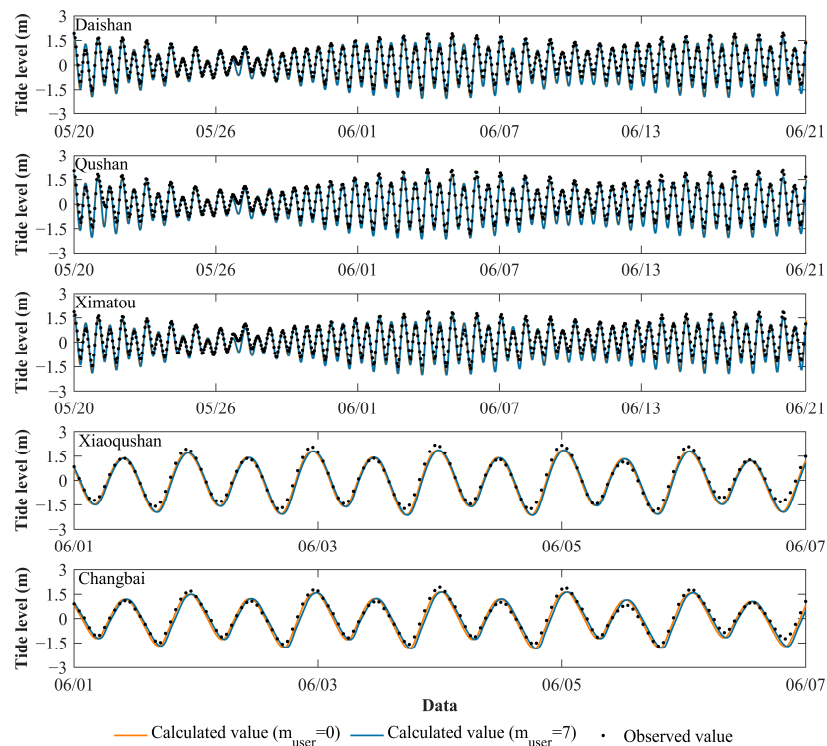


Figure 14. Comparison of simulated and observed water level time series. Stations, Xiaoqushan and Changbai, retained only 7 days of comparisons to show detail.

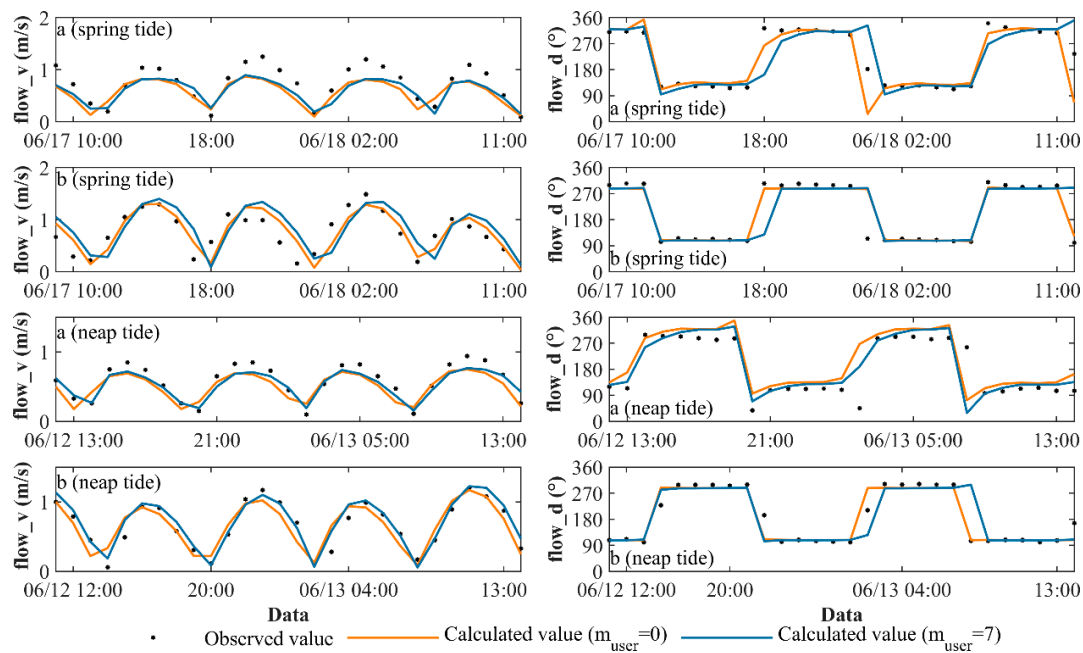


Figure 15. Comparison of simulated and measured flow velocity and flow direction time series. The left side of the figures shows the tidal flow velocity time series for sites a and b during spring and neap tides, while the right side displays the corresponding flow directions.

Model performance is assessed using the skill score given by

$$\text{Skill score} = 1 - \frac{\sum_{i=1}^N (X_i^{\text{mod}} - X^{\text{obs}})^2}{\sum_{i=1}^N (|X_i^{\text{mod}} - \overline{X^{\text{obs}}}| + |X^{\text{obs}} - \overline{X^{\text{obs}}}|)^2} \quad (28)$$

where X_i^{mod} and X^{obs} represent simulated and observed values, respectively, and $\overline{X^{\text{obs}}}$ represents the mean value of the observed data. Skill score = 1 indicates perfect agreement between the model and observed results. Tables 7 and 8 present the skill scores for the GTS and LTS models with m_{user} values ranging from 0 to 7. The skill scores for all tidal levels consistently reach above 0.97, with exceptional performances at the Xiaoqushan and Changbai stations achieving a score of 0.98. Although there is a marginal decline in scores with the increase in m_{user} , the overall results remain highly satisfactory. In terms of flow velocity, except for the flow velocity during the spring tides at station a, all other stations consistently score 0.90 or above. Remarkably, the results at $m_{\text{user}} = 7$ even outperform those at $m_{\text{user}} = 0$.

Table 7. Skill scores of water level (h) for tidal simulation in Zhoushan Islands.

m_{user}	Observation Sites				
	Daishan	Qushan	Ximatou	Xiaoqushan	Changbai
0	0.9702	0.9731	0.9656	0.9820	0.9805
1	0.9702	0.9731	0.9656	0.9819	0.9805
2	0.9701	0.9730	0.9655	0.9819	0.9805
3	0.9702	0.9730	0.9656	0.9818	0.9805
4	0.9704	0.9729	0.9660	0.9816	0.9807
5	0.9707	0.9730	0.9668	0.9811	0.9806
6	0.9715	0.9731	0.9687	0.9801	0.9800
7	0.9719	0.9724	0.9702	0.9762	0.9744

Table 8. Skill scores of flow velocity (m/s) and direction ($^{\circ}$) for tidal simulation in Zhoushan Islands.

m_{user}	a (Spring Tide)		b (Spring Tide)		a (Neap Tide)		b (Neap Tide)	
	Velocity	Direction	Velocity	Direction	Velocity	Direction	Velocity	Direction
0	0.8449	0.9393	0.9106	0.9660	0.9137	0.8856	0.9406	0.9749
1	0.8451	0.9392	0.9105	0.9660	0.9138	0.8857	0.9408	0.9751
2	0.8454	0.9387	0.9104	0.9660	0.9140	0.8859	0.9408	0.9751
3	0.8485	0.9372	0.9116	0.9657	0.9167	0.8870	0.9420	0.9751
4	0.8555	0.9327	0.9157	0.9652	0.9236	0.8894	0.9461	0.9751
5	0.8667	0.9108	0.9230	0.9481	0.9396	0.8915	0.9526	0.9754
6	0.8745	0.9045	0.9387	0.9264	0.9479	0.9023	0.9646	0.9758
7	0.8687	0.9259	0.9574	0.8946	0.9506	0.9096	0.9739	0.9306

Model performance is also evaluated using RMSE (Tables 9 and 10). LTS model results are in close agreement with the observed values.

Table 9. RMSEs of water level (h) for tidal simulation in Zhoushan Islands.

m_{user}	Observation Sites				
	Daishan	Qushan	Ximatou	Xiaoqushan	Changbai
0	0.3078	0.3082	0.3135	0.2664	0.2381
1	0.3081	0.3085	0.3138	0.2666	0.2382
2	0.3084	0.3090	0.3141	0.2670	0.2383
3	0.3085	0.3098	0.3140	0.2679	0.2382
4	0.3076	0.3105	0.3125	0.2697	0.2376
5	0.3067	0.3110	0.3097	0.2738	0.2382
6	0.3035	0.3114	0.3015	0.2814	0.2421
7	0.3003	0.3154	0.2931	0.3067	0.2715

Table 10. RMSEs of flow velocity (m/s) and direction ($^{\circ}$) of tidal simulation in Zhoushan Islands.

m_{user}	a (Spring Tide)		b (Spring Tide)		a (Neap Tide)		b (Neap tide)	
	Velocity	Direction	Velocity	Direction	Velocity	Direction	Velocity	Direction
0	0.2377	46.9905	0.2211	34.0394	0.1301	63.9208	0.1558	27.8543
1	0.2375	47.0254	0.2213	34.0368	0.1299	63.8982	0.1559	27.8552
2	0.2370	47.2495	0.2216	34.0416	0.1296	63.8141	0.1561	27.8529
3	0.2346	47.8250	0.2206	34.1677	0.1276	63.4787	0.1549	27.8452
4	0.2293	49.6210	0.2165	34.4367	0.1223	62.7404	0.1502	27.8440
5	0.2211	56.4806	0.2087	41.6385	0.1094	62.2197	0.1425	27.6574
6	0.2127	58.3850	0.1882	50.0988	0.1007	58.6215	0.1255	27.5439
7	0.2150	50.9924	0.1580	59.9253	0.0980	56.2860	0.1101	46.5289

5. Conclusions

A 2D cross-scale shallow water equations model based on an improved LTS scheme is developed in this paper, which is suitable for modeling cross-scale ocean hydrodynamic problems. The governing equations are numerically solved on unstructured triangular meshes using the finite volume method and Roe's approximate Riemann solver. A more stable and conservation-focused LTS scheme has been developed, employing a single-layer interface strategy and a prediction–correction method.

Model performance is validated in the spreading of a drop of shallow water and the tidal flat/channel system. The results indicate that the improved LTS scheme exhibits superior conservation properties, ensuring precision throughout the computational process without compromising water quality. Moreover, the novel LTS scheme significantly improves computational efficiency without incurring additional burdens compared to the original scheme.

Due to the efficiency and conservativeness of the improved LTS scheme, we successfully simulated the tidal processes in the Bohai Sea, Yellow Sea, East China Sea, and Zhoushan Islands using the established high-efficiency cross-scale 2D shallow water model. The simulation results are well matched with the observed data at the Zhoushan Islands. Furthermore, the computational efficiency is improved by a factor of 3.92. The new LTS scheme guarantees accuracy even at LTS level 7. This improvement paves the way for a promising future in a wide range of cross-scale ocean modeling applications.

6. Discussions

Note that, the speedup factor of the LTS scheme proposed in this paper to improve the efficiency of the model varies with the non-uniformity of the meshes and flow velocities but does not increase indefinitely. Its variations are contingent on the number of cells at different levels and are dictated by the LTS level encompassing the most cells. And the overhead of the algorithm increases as the LTS rank increases, which is also a tradeoff. Additionally, the time scheme used in this article is first-order, which has relatively low computational accuracy. In future research, we plan to explore the development of high-order accuracy LTS schemes. In conjunction with the study conducted by Hoang et al. [14], the explicit LTS scheme employed in this paper, based on prediction–correction, exhibits inherent parallelism. Therefore, it is also possible to combine the improved LTS algorithm with program parallel algorithms [15,16] to further improve the computational efficiency of the model. Moreover, the cross-scale coastal hydrodynamic model developed based on the improved LTS scheme in this study holds promise for simulating nearshore storm surges and tidal flats, offering broad applications.

Author Contributions: G.L.: supervision, funding acquisition, writing—review and editing. T.J.: investigation, software, validation, writing—original draft. G.W.: conceptualization, resources, supervision, funding acquisition, writing—review and editing. H.T.: methodology, formal analysis, funding acquisition, writing—review and editing. P.Y.: software, data curation, project administration, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (No. 52071305 supported by Guoxiang Wu, No. 52071306 supported by Guilin Liu, No. 11801533 and No. 11971482 both supported by Hao Tian), the Natural Science Foundation of Zhejiang Province (No. LZ22E090003 supported by Pubing Yu), and the Chinese Fundamental Research Funds for the Central Universities (No. 202264006 and No. 202042008 both supported by Hao Tian).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The datasets supporting the conclusions of this article are available upon reasonable request. Topographic data used in this study were derived from publicly accessible sources, including NOAA (<https://ngdc.noaa.gov>, accessed on 20 June 2024) and relevant nautical charts for coastal regions. Detailed simulation data and additional datasets generated during the current study are available from the corresponding author upon reasonable request, subject to compliance with applicable data privacy or confidentiality obligations.

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

References

1. Danilov, S. Ocean modeling on unstructured meshes. *Ocean Model.* **2013**, *69*, 195–210. [CrossRef]
2. Li, Q.; Van Roekel, L. Towards multiscale modeling of ocean surface turbulent mixing using coupled MPAS-Ocean v6.3 and PALM v5.0. *Geosci. Model. Dev.* **2021**, *14*, 2011–2028. [CrossRef]
3. Korn, P. Formulation of an unstructured grid model for global ocean dynamics. *J. Comput. Phys.* **2017**, *339*, 525–552. [CrossRef]
4. Lilly, J.R.; Capodaglio, G.; Petersen, M.R.; Brus, S.R.; Engwirda, D.; Higdon, R.L. Storm Surge Modeling as an Application of Local Time-Stepping in MPAS-Ocean. *J. Adv. Model. Earth Syst.* **2023**, *15*, e2022MS003327. [CrossRef]
5. Osher, S.; Sanders, R. Numerical approximations to nonlinear conservation laws with locally varying time and space grids. *Math. Comput.* **1983**, *41*, 321–336. [CrossRef]
6. Kleb, W.L.; Batina, J.T.; Williams, M.H. Temporal adaptive Euler/Navier-Stokes algorithm involving unstructured dynamic meshes. *AIAA J.* **1992**, *30*, 1980–1985. [CrossRef]

7. Dawson, C. High resolution upwind-mixed finite element methods for advection-diffusion equations with variable time-stepping. *Numer. Methods Part. Differ. Equ.* **1995**, *11*, 525–538. [\[CrossRef\]](#)
8. Dawson, C.; Kirby, R. High resolution schemes for conservation laws with locally varying time steps. *SIAM J. Sci. Comput.* **2001**, *22*, 2256–2281. [\[CrossRef\]](#)
9. Fumeaux, C.; Baumann, D.; Leuchtmann, P.; Vahldieck, R. A Generalized Local Time-Step Scheme for Efficient FVTD Simulations in Strongly Inhomogeneous Meshes. *IEEE Trans. Microw. Theory* **2004**, *52*, 1067–1076. [\[CrossRef\]](#)
10. Müller, S.; Stiriba, Y. Fully Adaptive Multiscale Schemes for Conservation Laws Employing Locally Varying Time Stepping. *J. Sci. Comput.* **2007**, *30*, 493–531. [\[CrossRef\]](#)
11. Tan, Z.J.; Zhang, Z.R.; Huang, Y.Q.; Tang, T. Moving mesh methods with locally varying time steps. *J. Comput. Phys.* **2004**, *200*, 347–367. [\[CrossRef\]](#)
12. Trahan, C.J.; Dawson, C. Local time-stepping in Runge–Kutta discontinuous Galerkin finite element methods applied to the shallow-water equations. *Comput. Methods Appl. Mech. Eng.* **2012**, *217–220*, 139–152. [\[CrossRef\]](#)
13. Krivodonova, L. An efficient local time-stepping scheme for solution of nonlinear conservation laws. *J. Comput. Phys.* **2010**, *229*, 8537–8551. [\[CrossRef\]](#)
14. Hoang, T.; Leng, W.; Ju, L.L.; Wang, Z.; Pieper, K. Conservative explicit local time-stepping schemes for the shallow water equations. *J. Comput. Phys.* **2019**, *382*, 152–176. [\[CrossRef\]](#)
15. Dazzi, S.; Vacondio, R.; Dal Palù, A.; Mignosa, P. A local time stepping algorithm for GPU-accelerated 2D shallow water models. *Adv. Water Resour.* **2018**, *111*, 274–288. [\[CrossRef\]](#)
16. Hu, P.; Zhao, Z.X.; Ji, A.F.; Li, W.; He, Z.G.; Liu, Q.F.; Li, Y.W.; Cao, Z.X. A GPU-Accelerated and LTS-Based Finite Volume Shallow Water Model. *Water* **2022**, *14*, 922. [\[CrossRef\]](#)
17. Krámer, T.; Józsa, J. Solution-adaptivity in modelling complex shallow flows. *Comput. Fluids* **2007**, *36*, 562–577. [\[CrossRef\]](#)
18. Sanders, B.F. Integration of a shallow water model with a local time step. *J. Hydraul. Res.* **2008**, *46*, 466–475. [\[CrossRef\]](#)
19. Hu, P.; Lei, Y.; Han, J.; Cao, Z.; Liu, H.; He, Z.; Yue, Z. Improved Local Time Step for 2D Shallow-Water Modeling Based on Unstructured Grids. *J. Hydraul. Eng.* **2019**, *145*, 6019011–6019017. [\[CrossRef\]](#)
20. Hu, P.; Lei, Y.L.; Han, J.J.; Cao, Z.X.; Liu, H.H.; He, Z.G. Computationally efficient modeling of hydro-sediment-morphodynamic processes using a hybrid local time step/global maximum time step. *Adv. Water Resour.* **2019**, *127*, 26–38. [\[CrossRef\]](#)
21. Yang, X.; An, W.; Li, W.; Zhang, S. Implementation of a Local Time Stepping Algorithm and Its Acceleration Effect on Two-Dimensional Hydrodynamic Models. *Water* **2020**, *12*, 1148. [\[CrossRef\]](#)
22. Dawson, C.; Trahan, C.J.; Kubatko, E.J.; Westerink, J.J. A parallel local timestepping Runge–Kutta discontinuous Galerkin method with applications to coastal ocean modeling. *Comput. Methods Appl. Mech. Eng.* **2013**, *259*, 154–165. [\[CrossRef\]](#)
23. Dawson, C. *A Local Timestepping Runge–Kutta Discontinuous Galerkin Method for Hurricane Storm Surge Modeling*; Feng, X., Karakashian, O., Xing, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2014; Volume 157, pp. 133–148.
24. Roe, P.L. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **1981**, *43*, 357–372. [\[CrossRef\]](#)
25. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 87–114.
26. Yu, P.B.; Pan, C.H.; Xie, Y.L. 2-dimensional real time forecasting model for storm tides and its application in Hangzhou Bay. *J. Hydrodynamics* **2011**, *26*, 747–756. [\[CrossRef\]](#)
27. Adamy, K.; Pham, D. A Finite Volume Implicit Euler Scheme for the Linearized Shallow Water Equations: Stability and Convergence. *Numer. Func. Anal. Opt.* **2006**, *27*, 757–783. [\[CrossRef\]](#)
28. Chippada, S.; Dawson, C.N.; Martinez, M.L.; Wheeler, M.F. A Godunov-type finite volume method for the system of Shallow water equations. *Comput. Methods Appl. Mech. Eng.* **1998**, *151*, 105–129. [\[CrossRef\]](#)
29. Benkhaldoun, F.; Sahmim, S.; Seaid, M. Solution of the Sediment Transport Equations Using a Finite Volume Method Based on Sign Matrix. *SIAM J. Sci. Comput.* **2009**, *31*, 2866–2889. [\[CrossRef\]](#)
30. Cea, L.; French, J.R.; Vázquez-Cendón, M.E. Numerical modelling of tidal flows in complex estuaries including turbulence: An unstructured finite volume solver and experimental validation. *Int. J. Numer. Methods Eng.* **2006**, *67*, 1909–1932. [\[CrossRef\]](#)
31. Kong, J.; Song, Z.Y.; Zhang, H.G. New non-structured numerical model for solving shallow water equation and its application. *J. Hohai Univ. (Nat. Sci.)* **2006**, *34*, 456–459. [\[CrossRef\]](#)
32. Pan, C.H.; Lu, H.Y. 2D numerical model for discontinuous shallow water flows and application to simulation of tidal bore. *J. Zhejiang Univ.* **2009**, *43*, 2107–2113. [\[CrossRef\]](#)
33. Schär, C.; Smolarkiewicz, P.K. A Synchronous and Iterative Flux-Correction Formalism for Coupled Transport Equations. *J. Comput. Phys.* **1996**, *128*, 101–120. [\[CrossRef\]](#)
34. Wu, G.X.; Shi, F.Y.; Kirby, J.T.; Mieras, R.; Liang, B.C.; Li, H.J.; Shi, J. A pre-storage, subgrid model for simulating flooding and draining processes in salt marshes. *Coast. Eng.* **2016**, *108*, 65–78. [\[CrossRef\]](#)
35. Egbert, G.D.; Erofeeva, S.Y. Efficient Inverse Modeling of Barotropic Ocean Tides. *J. Atmos. Ocean. Technol.* **2002**, *19*, 183–204. [\[CrossRef\]](#)

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.