

Article

Development of a Three-Dimensional Hydrodynamic Model Based on the Discontinuous Galerkin Method

Guoquan Ran, Qinghe Zhang  and Zereng Chen

State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin 300072, China

* Correspondence: qhzhzhang@tju.edu.cn

Abstract: Though the discontinuous Galerkin method is attracting more and more applications in many fields due to its local conservation, high-order accuracy, and flexibility for resolving complex geometries, only a few three-dimensional hydrodynamic models based on the discontinuous Galerkin method are present. In this study, a three-dimensional hydrodynamic model with a σ -coordinate system in the vertical direction is developed. This model is discretized in space using the discontinuous Galerkin method and advanced in time with the implicit-explicit Runge–Kutta method. Numerical tests indicate that the developed model is convergent and can obtain better results with a smaller computational time when a higher approximation order is adopted. Other tests with exact solutions also indicate that the developed model can well simulate the vertical circulation under the effect of surface wind stress and the flow under the combined effect of wind stress and Coriolis acceleration terms. The simulation results of tidal flow in part of Bohai Bay, China, indicate that the model can be used for the simulation of tidal wave motion in realistic situations.

Keywords: σ -coordinate system; three-dimensional hydrodynamic model; discontinuous Galerkin method; implicit-explicit Runge–Kutta method



Citation: Ran, G.; Zhang, Q.; Chen, Z. Development of a Three-Dimensional Hydrodynamic Model Based on the Discontinuous Galerkin Method. *Water* **2023**, *15*, 156. <https://doi.org/10.3390/w15010156>

Academic Editors: Gholamreza Kefayati and Hasan Sajjadi

Received: 30 November 2022

Revised: 24 December 2022

Accepted: 26 December 2022

Published: 30 December 2022



Copyright: © 2022 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 assumption of hydrostatic pressure distribution, the three-dimensional shallow water equations (3D SWEs) can be derived from the Reynolds-averaged three-dimensional Navier–Stokes equations. They can predict the vertical distribution of primitive variables and can be used together with advection-diffusion-type equations to simulate the transport and fate of substances, such as temperature, salt, and sediment.

In the last few decades, 3D SWEs have been intensively studied [1], and many hydrodynamic models have been developed and widely used. A brief categorization of these models, based on the numerical methods and the type of grids used in the horizontal direction, is listed in Table 1. For the models at an early stage, the conservative finite difference method on structured grids was usually used. They can be implemented straightforwardly and are computationally efficient, while they approximate the coastlines as staircases and prevent the flexible implementation of variable resolution. Later, unstructured, grid-based hydrodynamic models, such as the finite element and finite volume method, were brought into the main stream, since the coastlines could be accurately represented to give a reasonable simulation at the regional scale [2], and the area of most concern could be simulated using local refinement grids.

In addition to the numerical methods mentioned above, the discontinuous Galerkin (DG) method has received much attention in recent years. It can be viewed as a hybrid between the finite element method and the finite volume method and enjoys most of the strengths of both methods, such as local conservation, compactness, a high order of accuracy, and good application to unstructured meshes. It has been successfully applied in the numerical discretization of two-dimensional shallow water equations [3], two-dimensional non-hydrostatic shallow water equations [4], and Boussinesq-type equations [5].

Table 1. Categorization of the commonly used hydrodynamic model.

Grid Type	Numerical Method		
	Finite Difference Method	Finite Volume Method	Finite Element Method
Structured grid	ROMS [6], POM [7], MoM [8], GETM [9], TRIM [10], DELFT3D [11], EFDC [12], ECOMSED [13], COHERENS [14]	MITgcm [15], MOHID [16]	
Unstructured grid		FVCOM [17], UnTRIM [18], HydroInfo [19]	FESOM [20], ICOM, SELFE [21], ADCIRC [22], SCHISM [23], TELEMAC [24]

In the realm of 3D SWEs with DG solutions, Dawson and Aizinger [25] first developed a three-dimensional barotropic model with a z -coordinate system adopted in the vertical direction and presented the corresponding stability analysis [26]. The vertical eddy viscosity terms in the model are explicitly advanced in time. Due to the stiffness of these terms, the time step must be small enough, which limits the applicability of the model. Later, Aizinger et al. [27] made improvements by introducing the baroclinic terms and treating both the vertical eddy viscosity and the vertical diffusion terms implicitly. This model is called UTBEST3D and has been validated with realistic applications. In the same period, Blaise [28] and Comblen [29] developed the three-dimensional baroclinic model named SLIM and adopted it for the simulation of various baroclinic phenomena. Due to the discontinuous nature of the DG method, the water depth between the adjacent elements is generally not continuous such that the lateral boundaries are mismatched, and the post-process measures are needed to smooth the water depth such that the calculation of numerical flux will not be affected. To circumvent the problem of boundary mismatch between adjacent elements, Gandham [30] mapped the time-varying z -coordinate system to a fixed domain by invoking the σ -coordinate transformation and developed a three-dimensional barotropic model. However, only one layer is adopted in the vertical direction, which may lead to serious numerical errors when the water depth is large. Conroy and Kubatko [31] developed a three-dimensional barotropic model with a σ -coordinate system adopted in the vertical direction and developed a fast and accurate method for the calculation of depth-averaged velocity. An arbitrary number of layers can be used in the vertical direction. However, non-linear advective terms are not included, and the vertical eddy viscosity terms are explicitly advanced in time. To the authors' knowledge, there is currently no DG method-based three-dimensional hydrodynamic model that includes non-linear advective terms, treats the vertical eddy viscosity terms implicitly, and can use an arbitrary number of vertical layers in the σ -coordinate system simultaneously.

In this paper, a three-dimensional hydrodynamic model that includes nonlinear advective terms and can use an arbitrary number of vertical layers in the σ -coordinate system is developed based on the DG method. The paper is organized as follows. In Section 2, we provide the governing equations of the 3D SWEs in the σ -coordinate system. Then, we present the space discretization and time stepping of the model in Section 3. Numerical experiments are given in Section 4. Finally, conclusions are drawn in Section 5.

2. Governing Equations

Using the notations given in Table 2, the governing equations of the three-dimensional hydrodynamic model in the physical domain are as follows [13]:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \quad (1)$$

$$\frac{\partial u}{\partial t} + \frac{\partial(uu)}{\partial x} + \frac{\partial(uv)}{\partial y} + \frac{\partial(uw)}{\partial z} - fv = -g \frac{\partial \eta}{\partial x} + \frac{\partial}{\partial z} \left(K_m \frac{\partial u}{\partial z} \right) + \frac{\partial}{\partial x} \left(K_h \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h \frac{\partial u}{\partial y} \right), \quad (2)$$

$$\frac{\partial v}{\partial t} + \frac{\partial(uv)}{\partial x} + \frac{\partial(vv)}{\partial y} + \frac{\partial(vw)}{\partial z} + fu = -g \frac{\partial \eta}{\partial y} + \frac{\partial}{\partial z} \left(K_m \frac{\partial v}{\partial z} \right) + \frac{\partial}{\partial x} \left(K_h \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h \frac{\partial v}{\partial y} \right), \quad (3)$$

Table 2. Notations for the governing equations of the three-dimensional hydrodynamic model.

Variables	Explanation
$u(x, y, z, t)$	velocity of water in x direction
$v(x, y, z, t)$	velocity of water in y direction
$w(x, y, z, t)$	velocity of water in z direction
g	acceleration due to gravity
θ	angle of geographical latitude
ω	magnitude of the angular velocity of the Earth
$f = 2\omega \sin \theta$	Coriolis parameter
K_m	vertical eddy viscosity coefficient
K_h	horizontal eddy viscosity coefficient
ρ_0	density
τ_{sx}	wind stress in x direction
τ_{sy}	wind stress in y direction
$\eta(x, y, t)$	the surface elevation
$u_\eta(x, y, t)$	surface velocity of water in x direction
$v_\eta(x, y, t)$	surface velocity of water in y direction
$b(x, y)$	bottom elevation
$u_b(x, y, t)$	bottom velocity of water in x direction
$v_b(x, y, t)$	bottom velocity of water in y direction
$z_0(x, y, t)$	half height of the bottommost element
$u_c(x, y, t)$	velocity of water at z_0 in x direction
$v_c(x, y, t)$	velocity of water at z_0 in y direction
$C_f(x, y, t)$	drag coefficient
κ	von Karman constant
$z_0^b(x, y)$	bottom roughness parameter
$\omega(x, y, \sigma, t)$	vertical velocity in the computational domain
$\bar{U}(x, y, t)$	depth-averaged velocity in x direction
$\bar{V}(x, y, t)$	depth-averaged velocity in y direction
$D(x, y, t) = \eta(x, y, t) - b(x, y)$	water depth

The surface boundary conditions for u , v , and w are

$$\left(K_m \frac{\partial u}{\partial z}, K_m \frac{\partial v}{\partial z} \right) = \frac{1}{\rho_0} (\tau_{sx}, \tau_{sy}), \quad (4)$$

$$w = \frac{\partial \eta}{\partial t} + u_\eta \frac{\partial \eta}{\partial x} + v_\eta \frac{\partial \eta}{\partial y}. \quad (5)$$

Likewise, the bottom boundary conditions for u , v , and w are

$$w = u_b \frac{\partial b}{\partial x} + v_b \frac{\partial b}{\partial y}, \quad (6)$$

$$\left(K_m \frac{\partial u}{\partial z}, K_m \frac{\partial v}{\partial z}\right) = C_f \sqrt{u_c^2 + v_c^2} (u_c, v_c), \quad (7)$$

where C_f is determined by matching a logarithmic bottom layer to the model at height z_0 and calculated as [32]

$$C_f = \left(\frac{\kappa}{\ln((z_0 + z_0^b)/z_0^b)} \right)^2. \quad (8)$$

To accurately represent the bottom and surface geometry, the σ transformation developed by Philips [33] is adapted

$$t^* = t, x^* = x, y^* = y, \sigma = \frac{z - \eta(x, y, t)}{D(x, y, t)}. \quad (9)$$

According to the principle of chain differentiation, partial derivatives of the variable $\zeta = \zeta(x, y, z, t)$ in the physical domain can be written in the following form in the computational domain:

$$\begin{aligned} \frac{\partial \zeta}{\partial t} &= \frac{\partial \zeta}{\partial t^*} + \frac{\partial \zeta}{\partial \sigma} \frac{\partial \sigma}{\partial t}, \\ \frac{\partial \zeta}{\partial x} &= \frac{\partial \zeta}{\partial x^*} + \frac{\partial \zeta}{\partial \sigma} \frac{\partial \sigma}{\partial x}, \\ \frac{\partial \zeta}{\partial y} &= \frac{\partial \zeta}{\partial y^*} + \frac{\partial \zeta}{\partial \sigma} \frac{\partial \sigma}{\partial y}, \\ \frac{\partial \zeta}{\partial z} &= \frac{\partial \zeta}{\partial \sigma} \frac{\partial \sigma}{\partial z}. \end{aligned} \quad (10)$$

Substituting Equation (10) back into Equations (1)–(3), and ignoring the superscript “*”, we can obtain the governing equations in the computational domain as

$$\frac{\partial D}{\partial t} + \frac{\partial(Du)}{\partial x} + \frac{\partial(Dv)}{\partial y} + \frac{\partial \omega}{\partial \sigma} = 0, \quad (11)$$

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial \sigma} = \mathbf{S}_0 + \mathbf{S}_f + \mathbf{S}_{d,v} + \mathbf{S}_{d,h}. \quad (12)$$

with

$$\mathbf{U} = \begin{bmatrix} Du \\ Dv \end{bmatrix}, \mathbf{E} = \begin{bmatrix} Du^2 + 1/2(g(D^2 - z_b^2)) \\ Du v \end{bmatrix}, \mathbf{G} = \begin{bmatrix} Du v \\ Dv^2 + 1/2(g(D^2 - z_b^2)) \end{bmatrix}, \mathbf{H} = \begin{bmatrix} \omega u \\ \omega v \end{bmatrix}. \quad (13)$$

$$\mathbf{S}_0 = \begin{bmatrix} -g\eta \partial z_b / \partial x \\ -g\eta \partial z_b / \partial y \end{bmatrix}, \mathbf{S}_f = \begin{bmatrix} -f Dv \\ f Du \end{bmatrix}. \quad (14)$$

$$\mathbf{S}_{d,v} = \begin{bmatrix} \frac{\partial}{\partial \sigma} \left(\frac{K_m}{D^2} \frac{\partial(Du)}{\partial \sigma} \right) \\ \frac{\partial}{\partial \sigma} \left(\frac{K_m}{D^2} \frac{\partial(Dv)}{\partial \sigma} \right) \end{bmatrix}, \mathbf{S}_{d,h} = \begin{bmatrix} \frac{\partial}{\partial x} \left(K_h D \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h D \frac{\partial u}{\partial y} \right) \\ \frac{\partial}{\partial x} \left(K_h D \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(K_h D \frac{\partial v}{\partial y} \right) \end{bmatrix}. \quad (15)$$

where ω is related to w by

$$\omega = D \frac{d\sigma}{dt} = w - \left(\frac{\partial z_s}{\partial t} + \sigma \frac{\partial D}{\partial t} \right) - u \left(\frac{\partial z_s}{\partial x} + \sigma \frac{\partial D}{\partial x} \right) - v \left(\frac{\partial z_s}{\partial y} + \sigma \frac{\partial D}{\partial y} \right). \quad (16)$$

The surface boundary condition in the computational domain is given by

$$\begin{aligned} \omega &= 0, \\ \left(\frac{K_m}{D^2} \frac{\partial(Du)}{\partial \sigma}, \frac{K_m}{D^2} \frac{\partial(Dv)}{\partial \sigma} \right) &= \frac{1}{\rho_0} (\tau_{sx}, \tau_{sy}), \end{aligned} \quad (17)$$

and the bottom boundary condition is given by

$$\omega = 0, \quad \left(\frac{K_m}{D^2} \frac{\partial(Du)}{\partial\sigma}, \frac{K_m}{D^2} \frac{\partial(Dv)}{\partial\sigma} \right) = C_f \sqrt{u_b^2 + v_b^2} (u_b, v_b), \quad (18)$$

Integrating Equation (11) from bottom to surface and applying the boundary condition about ω leads to the primitive continuity equation as

$$\frac{\partial D}{\partial t} + \frac{\partial(D\bar{U})}{\partial x} + \frac{\partial(D\bar{V})}{\partial y} = 0 \quad (19)$$

3. Numerical Implementation

In this section, we will detail the numerical implementation of our model based on the DG method. The domain partition and the discontinuous polynomial space are first described. Then, the numerical discretization of each part is presented, followed by the time-stepping scheme.

3.1. Domain Partition and Polynomial Space

For an arbitrary three-dimensional domain Ω_{3d} , its horizontal projection is denoted as Ω_{2d} and is partitioned by $Ele2d$ non-overlapping triangular or quadrilateral elements; i.e., $\Omega_{2d,h} = \{\Omega_e\}, e \in [1, Ele2d]$. In the vertical direction, $\Omega_{2d,h}$ is extruded by N_L layers to obtain the three-dimensional computational domain $\Omega_{3d,h}$, and the computational domain consists of $Ele2d \times N_L$ triangular prisms or quadrangular prisms. In this study, the three-dimensional element obtained by the extrusion of Ω_e is denoted as $\Omega_{(e,L)}, L \in [1, N_L]$, with the bottommost element denoted as $\Omega_{(e,1)}$ and the upmost element denoted as $\Omega_{(e,N_L)}$. Figure 1 gives a schematic view of two columns of three-dimensional computational elements with Figure 1a showing the quadrangular prisms and Figure 1b showing the triangular prisms. To illustrate the discontinuous nature of the DG method, the discontinuities between the adjacent computational elements have been exaggerated, and no gap exists in reality.

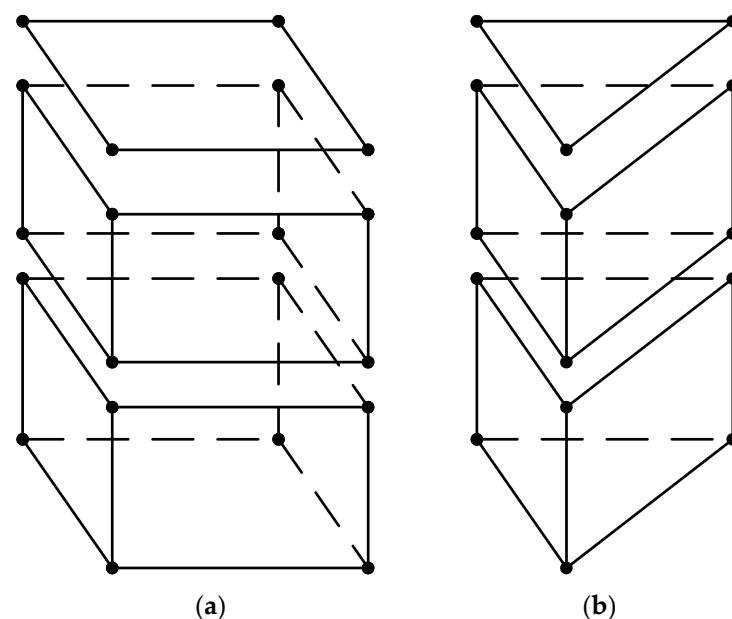


Figure 1. Schematic view of a column of three-dimensional computational elements. (a) quadrangular prisms; (b) triangular prisms.

In addition, we introduce the following finite dimensional space of polynomials:

$$V_{h,3d} = \left\{ s \in L^2(\Omega) : \forall \Omega_{(e,L)} \in \Omega_{3d,h}, s \in \mathcal{P}_{(N_h, N_v)}(\Omega_{(e,L)}) \right\}, \quad (20)$$

where $L^2(\Omega)$ is the space of the square-integrable functions and $\mathcal{P}_{(N_h, N_v)}(\Omega_{(e,L)})$ is the complete space of polynomials defined over $\Omega_{(e,L)}$, which is of order at most N_h in the horizontal direction and at most N_v in the vertical direction. In this study, $V_{h,3d}$ is spanned by N_p Lagrangian functions, $l_i(x)$, $i \in [1, N_p]$, and the dimension, N_p , depends on the approximation order and element type. For triangular prisms,

$$N_p = \frac{(N_h + 1)(N_h + 2)}{2} \times (N_v + 1), \quad (21)$$

while for quadrilateral prisms,

$$N_p = (N_h + 1)^2 \times (N_v + 1). \quad (22)$$

For any two adjacent elements Ω_{e1} and Ω_{e2} , let $\varepsilon = \partial\Omega_{e1} \cap \partial\Omega_{e2}$ be the unique interior interface between these two elements, and let $\mathbf{n}^-$ and $\mathbf{n}^+$ be the outward unit normal for these two elements on ε . For scalar field $c \in V_{h,3d}$, let $c_\varepsilon^\pm$ be the trace of c on interface ε from the interior of $\Omega_{e1(2)}$. We further define the mean value $\{c\}$ and the jump value $[c]$ on interface ε as

$$\{c\} = (c_\varepsilon^+ + c_\varepsilon^-)/2, [c] = c_\varepsilon^+ \mathbf{n}^+ + c_\varepsilon^- \mathbf{n}^-. \quad (23)$$

On the boundary of the computational domain, they are defined as

$$\{c\} = c, [c] = c\mathbf{n}, \quad (24)$$

where c is the single value and $\mathbf{n}$ is the outward unit normal on the boundary edges. We note that the jump of scalar c is a vector, and we further denote its horizontal and vertical components by $[c]_h$ and $[c]_\sigma$, respectively.

3.2. Numerical Discretization of Momentum Equations

For Equation (12), both the Coriolis acceleration terms and the vertical eddy viscosity terms are treated implicitly, while the others are treated explicitly.

3.2.1. Convective and Bottom Topography Terms

For the numerical discretization of Equation (12), we consider the following general formulation: find the local approximate solution vector $\mathbf{U}_h \in (V_{h,3d})^2$ such that for all test functions $\phi \in V_{h,3d}$ and $\Omega_{(e,L)} \in \Omega_{3d,h}$ we have

$$\int_{\Omega_{(e,L)}} \left(\frac{\partial \mathbf{U}_h}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{U}_h)}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U}_h)}{\partial y} + \frac{\partial \mathbf{H}(\mathbf{U}_h)}{\partial \sigma} \right) \phi d\mathbf{x} = \oint_{\partial\Omega_{(e,L)}} \phi \mathbf{n} \cdot (\mathbf{F}(\mathbf{U}_h^-) - \mathbf{F}^*(\mathbf{U}_h^-, \mathbf{U}_h^+)) d\mathbf{x} \\ + \int_{\Omega_{(e,L)}} \mathbf{S}_0(\mathbf{U}_h) \phi d\mathbf{x} + \mathbf{D}_{d,h}(D, K_h, \mathbf{U}_h, \phi) + \mathbf{D}_{d,fv,h}(D, K_m, \mathbf{U}_h, f, \phi). \quad (25)$$

where $\mathbf{U}_h^-$ and $\mathbf{U}_h^+$ are the values of the local approximate solution vector $\mathbf{U}_h$ on the interface of two adjacent elements, and $\mathbf{n} \cdot \mathbf{F}(\mathbf{U}_h^-) = n_x \mathbf{E}(\mathbf{U}_h^-) + n_y \mathbf{G}(\mathbf{U}_h^-) + n_\sigma \mathbf{H}(\mathbf{U}_h^-)$ and $\mathbf{n} \cdot \mathbf{F}^*(\mathbf{U}_h^-, \mathbf{U}_h^+)$ are the flux term and the numerical flux at the element boundary, respectively. For the three-dimensional computational mesh used in this study, the third component of the unit outward normal is zero, i.e., $n_\sigma = 0$, at the lateral boundary of the computational elements, the calculation of the numerical flux degenerates to that of the two-dimensional shallow water equations, and the Harten–Lax–van Leer with contact (HLLC) Riemann solver [34] is adopted. As $n_x = n_y = 0$ at the top and bottom boundaries of the computational elements, only the vertical convection needs to be considered, and the upwind flux is taken, i.e.,

$$\mathbf{n} \cdot \mathbf{F}^*(\mathbf{U}_h^-, \mathbf{U}_h^+) = 0.5n_\sigma^-(\omega^-\mathbf{U}_h^-/D(1 + \text{sgn}(\omega^-\mathbf{n}_\sigma^-)) + \omega^+\mathbf{U}_h^+/D(1 + \text{sgn}(\omega^+\mathbf{n}_\sigma^+))), \quad (26)$$

where sgn is the sign function. At the top boundary of the computational domain ($\sigma = 0$) and bottom boundary of the computational domain ($\sigma = -1$), the numerical flux is set to zero. Operators $\mathbf{D}_{d,h}(D, K_h, \mathbf{U}_h, f, \phi)$ and $\mathbf{D}_{d,fv,h}(D, K_m, \mathbf{U}_h, f, \phi)$ in Equation (25) are the discretization corresponding to the horizontal eddy viscosity terms and the vertical and Coriolis acceleration terms, which will be detailed in the forthcoming parts. The requirement that Equation (25) is valid for all $\phi \in V_{h,3d}$ and can be imposed by considering Equation (25) for all the basis functions that span the functional space $V_{h,3d}$, i.e., setting $\phi = l_i(\mathbf{x})$, $i \in [1, N_p]$.

For the discretization of both the volume and the surface terms, the quadrature-free manner of Hesthaven and Warburton [35] is adopted. We assume that these terms also lie in the finite dimensional space $(V_{h,3d})^2$ and can be expressed as

$$\begin{aligned} \mathbf{E}(\mathbf{U}_h) &\approx \mathbf{E}_h = \sum_{i=1}^{N_p} \mathbf{E}_i l_i, \mathbf{G}(\mathbf{U}_h) \approx \mathbf{G}_h = \sum_{i=1}^{N_p} \mathbf{G}_i l_i, \\ \mathbf{H}(\mathbf{U}_h) &\approx \mathbf{H}_h = \sum_{i=1}^{N_p} \mathbf{H}_i l_i, \mathbf{S}_0(\mathbf{U}_h) \approx \mathbf{S}_{0,h} = \sum_{i=1}^{N_p} \mathbf{S}_{0,i} l_i, \\ \mathbf{F}(\mathbf{U}_h^-) \cdot \mathbf{n} &\approx \mathbf{F}_h^- \cdot \mathbf{n} = \sum_{f=1}^{N_{fp}} (\mathbf{F}_{h,f}^- \cdot \mathbf{n}_f) l_f, \\ \mathbf{F}^*(\mathbf{U}_h^-, \mathbf{U}_h^+) \cdot \mathbf{n} &\approx \mathbf{F}_h^* \cdot \mathbf{n} = \sum_{f=1}^{N_{fp}} (\mathbf{F}_{h,f}^* \cdot \mathbf{n}_f) l_f, \end{aligned} \quad (27)$$

where $\{l_f\}_{f=1}^{N_{fp}}$ are the basis functions on the element boundaries and N_{fp} is the total number of basis functions along all the boundaries. According to this approximation, we can obtain the following semi-discrete equation in matrix form:

$$\begin{aligned} \mathcal{M}_{(e,L)} \frac{d\mathbf{U}_{h,(e,L)}}{dt} &= -\mathcal{S}_{x,(e,L)} \mathbf{E}_{h,(e,L)} - \mathcal{S}_{y,(e,L)} \mathbf{G}_{h,(e,L)} - \mathcal{S}_{\sigma,(e,L)} \mathbf{H}_{h,(e,L)} \\ &+ \mathcal{M}_{(e,L)}^{\partial f} (\mathbf{F}_{h,(e,L)}^- - \mathbf{F}_{h,(e,L)}^*) + \mathcal{M}_{(e,L)} \mathbf{S}_{0,h,(e,L)}, \end{aligned} \quad (28)$$

where $\mathbf{U}_{h,(e,L)} = [\mathbf{D}\mathbf{u}_{h,(e,L)}, \mathbf{D}\mathbf{v}_{h,(e,L)}]^T$ are the nodal values of the conservative variables at the N_p interpolation points; $\mathbf{E}_{h,(e,L)}$, $\mathbf{G}_{h,(e,L)}$ and $\mathbf{H}_{h,(e,L)}$ are the non-linear flux terms at the N_p interpolation points; $\mathbf{F}_{h,(e,L)}^- = \mathbf{E}_{h,(e,L)} \cdot \mathbf{n}_x + \mathbf{G}_{h,(e,L)} \cdot \mathbf{n}_y + \mathbf{H}_{h,(e,L)}^- \cdot \mathbf{n}_\sigma$ and $\mathbf{F}_{h,(e,L)}^*$ are the norm flux term and the numerical flux term at the N_{fp} facial interpolation points, respectively; $\mathbf{S}_{0,h,(e,L)}$ is the bottom topography source term at the N_p interpolation points; and $\mathcal{M}_{(e,L)}$, $\mathcal{S}_{(e,L)} = [\mathcal{S}_{x,(e,L)}, \mathcal{S}_{y,(e,L)}, \mathcal{S}_{\sigma,(e,L)}]$, and $\mathcal{M}_{(e,L)}^{\partial f}$ are the mass matrix, stiff matrix, and edge-mass matrix, respectively, and can be expressed as

$$\begin{aligned} [\mathcal{M}_{(e,L)}]_{i,j} &= \int_{\Omega_{(e,L)}} l_i l_j d\mathbf{x}, [\mathcal{M}_{(e,L)}^{\partial f}]_{i,f} = \oint_{\partial\Omega_{(e,L)}} l_i l_f d\mathbf{x}, \\ [\mathcal{S}_{x,(e,L)}]_{i,j} &= \int_{\Omega_{(e,L)}} l_i \frac{\partial l_j}{\partial x} d\mathbf{x}, [\mathcal{S}_{y,(e,L)}]_{i,j} = \int_{\Omega_{(e,L)}} l_i \frac{\partial l_j}{\partial y} d\mathbf{x}, \\ [\mathcal{S}_{\sigma,(e,L)}]_{i,j} &= \int_{\Omega_{(e,L)}} l_i \frac{\partial l_j}{\partial \sigma} d\mathbf{x}. \end{aligned} \quad (29)$$

3.2.2. Horizontal Eddy Viscosity Terms

For the discretization of the horizontal eddy viscosity terms in Equation (12), the local discontinuous Galerkin method by Cockburn and Shu [36] is adopted. To discretize the

horizontal eddy viscosity terms in the momentum equation about Du , we first express it in the following form:

$$r = \nabla_{xy} \cdot (K_h D \nabla_{xy} u), \quad (30)$$

where r is an auxiliary function. Following the philosophy of the local discontinuous Galerkin method, we introduce the auxiliary variables and cast Equation (30) into the following form:

$$\begin{aligned} \hat{q} &= (\hat{q}_x, \hat{q}_y) = \nabla_{xy} u, \\ q &= v_h \hat{q}, \\ r &= \nabla_{xy} \cdot q, \end{aligned} \quad (31)$$

with $v_h = K_h D$ and $\nabla_{xy} = (\partial/\partial x, \partial/\partial y)$. For the numerical discretization of Equation (31), we assume a local approximate vector solution $(u_h, \hat{q}_h) = (u_h, \hat{q}_{h,x}, \hat{q}_{h,y}) \in V_{h,3d} \times (V_{h,3d})^2$ and $q_h = (q_{h,x}, q_{h,y}) \in (V_{h,3d})^2$ such that for all $(\phi, \pi) \in V_{h,3d} \times (V_{h,3d})^2$, we have

$$\begin{aligned} \int_{\Omega_{(e,L)}} \pi \cdot (\hat{q}_h - \nabla_{xy} u_h) dx &= - \int_{\partial\Omega_{(e,L)}} \pi \cdot n_h (u_h^- - u_h^*) dx, \\ \int_{\Omega_{(e,L)}} \pi \cdot q_h dx &= \int_{\Omega_{(e,L)}} v_h \hat{q}_h \cdot \pi dx, \\ \int_{\Omega_{(e,L)}} \phi r_h dx &= \int_{\Omega_{(e,L)}} \phi (\nabla_{xy} \cdot q_h) dx - \int_{\partial\Omega_{(e,L)}} \phi n_h \cdot (q_h^- - q_h^*) dx, \end{aligned} \quad (32)$$

where the terms marked with superscript “*” are the flux terms and are given as

$$\begin{aligned} u_h^* &= \{u_h\}, \\ q_h^* &= \{(v_h \nabla_{xy} u_h)\} - \tau [u_h]_h, \end{aligned} \quad (33)$$

where τ is the penalty parameter defined as

$$\tau = \frac{(N+1)(N+3)}{3} \frac{n_0}{2} \frac{A}{V} \max(v_h^-, v_h^+), \quad (34)$$

where $N = \max(N_h, N_v)$, n_0 is the number of neighbors of the element, i.e., five for triangular prisms and six for quadrilateral prisms, A is the area of the interface, and V is the average volume of the two adjacent elements.

We further suppose $(\hat{q}_h, q_h, v_h, r_h, u_h) \in (V_{h,3d})^2 \times (V_{h,3d})^2 \times V_{h,3d} \times V_{h,3d} \times V_{h,3d}$ and express these terms as

$$\begin{aligned} \hat{q}_h &= \sum_{i=1}^{N_p} \hat{q}_{h,(e,L),i} l_i(x), \quad q_h = \sum_{i=1}^{N_p} q_{h,(e,L),i} l_i(x), \quad v_h = \sum_{i=1}^{N_p} v_{h,(e,L),i} l_i(x), \\ r_h &= \sum_{i=1}^{N_p} r_{h,(e,L),i} l_i(x), \quad u_h = \sum_{i=1}^{N_p} u_{h,(e,L),i} l_i(x), \end{aligned} \quad (35)$$

where $\hat{q}_{h,(e,L),i} = (\hat{q}_{h,(e,L),x,i}, \hat{q}_{h,(e,L),y,i})$ and $q_{h,i} = (q_{h,(e,L),x,i}, q_{h,(e,L),y,i})$ are the auxiliary variables at the i th interpolation point. Substituting Equation (35) back into Equation (32), we have

$$\begin{aligned} \mathcal{M}_{(e,L)} \hat{q}_{h,(e,L),x} &= \mathcal{S}_{x,(e,L)} u_{h,(e,L)} - \mathcal{M}_{(e,L),f} \left[(u_{h,(e,L)}^- - u_{h,(e,L)}^*) \cdot n_x \right], \\ \mathcal{M}_{(e,L)} \hat{q}_{h,(e,L),y} &= \mathcal{S}_{y,(e,L)} u_{h,(e,L)} - \mathcal{M}_{(e,L),f} \left[(u_{h,(e,L)}^- - u_{h,(e,L)}^*) \cdot n_y \right], \end{aligned} \quad (36)$$

$$q_{h,(e,L),x} = v_{h,(e,L)} \cdot \hat{q}_{h,(e,L),x}, \quad q_{h,(e,L),y} = v_{h,(e,L)} \cdot \hat{q}_{h,(e,L),y}, \quad (37)$$

$$\mathcal{M}_{(e,L)} r_h = \mathcal{S}_{x,(e,L)} q_{h,(e,L),x} + \mathcal{S}_{y,(e,L)} q_{h,(e,L),y} - \mathcal{M}_{(e,L),f} \left[(q_{h,(e,L)}^- - q_{h,(e,L)}^*) \cdot n_h \right], \quad (38)$$

where $\mathbf{n}'_h = [\mathbf{n}_x, \mathbf{n}_y]$ is the vector of the horizontal components of the outward unit normal outward at the facial interpolation points.

The right-hand side of Equation (38) corresponds to the numerical discretization of the second-order term for the horizontal momentum equation about Du , and the same procedure applies to that about Dv .

3.2.3. Vertical Eddy Viscosity and Coriolis Acceleration Terms

For the discretization of the vertical eddy viscosity and Coriolis acceleration terms in Equation (12), we first express it in the following form:

$$\begin{aligned} r &= \frac{\partial}{\partial \sigma} \left(\frac{K_m}{D^2} \frac{\partial(Du)}{\partial \sigma} \right) - fDv, \text{ in } \Omega \\ \frac{K_m}{D^2} \frac{\partial(Du)}{\partial \sigma} n_\sigma &= g_N^{Du}, \text{ on } \Omega_N \\ Du &= g_D^{Du}, \text{ on } \Omega_D \end{aligned} \quad (39)$$

where g_N^{Du} is the Neumann boundary condition about Du on Ω_N and is given in Equations (17) and (18); g_D^{Du} is the Dirichlet boundary condition about Du on Ω_D . The symmetric interior penalty discontinuous Galerkin (SIPG) method [37] is adopted, and the corresponding primal form is given as

$$\begin{aligned} &\int_{\Omega_{h,3d}} v_v \frac{\partial \phi}{\partial \sigma} \frac{\partial(Du)_h}{\partial \sigma} d\mathbf{x} - \int_{\varepsilon_I} [(Du)_h]_\sigma \left\{ v_v \frac{\partial \phi}{\partial \sigma} \right\} d\mathbf{x} - \\ &\int_{\varepsilon_I} \left(\left\{ v_v \frac{\partial(Du)_h}{\partial \sigma} \right\} - \tau[(Du)_h]_\sigma \right) [\phi]_\sigma d\mathbf{x} - \int_{\partial\Omega_D} \phi v_v \frac{\partial(Du)_h}{\partial \sigma} n_\sigma d\mathbf{x} \\ &- \int_{\partial\Omega_D} (Du)_h v_v \frac{\partial \phi}{\partial \sigma} n_\sigma d\mathbf{x} + \int_{\partial\Omega_D} \tau \phi n_\sigma^2 (Du)_h d\mathbf{x} + \int_{\Omega_{h,3d}} \phi f(Dv)_h d\mathbf{x} \\ &= - \int_{\Omega_{h,3d}} \phi r d\mathbf{x} + \int_{\partial\Omega_D} \tau \phi n_\sigma^2 g_D^{Du} d\mathbf{x} - \int_{\partial\Omega_D} g_D^{Du} v_v \frac{\partial \phi}{\partial \sigma} n_\sigma d\mathbf{x} + \int_{\partial\Omega_N} \phi n_\sigma^2 g_N^{Du} d\mathbf{x}. \end{aligned} \quad (40)$$

where $v_v = K_m/D^2$ is the vertical eddy viscosity coefficient in the computational domain. The primal form about Dv is similar to that about Du . In this study, the Neumann boundary conditions for Du and Dv given in Equation (18) are treated implicitly, and the bottom velocity module $\sqrt{u_c^2 + v_c^2}$ is linearized by using the old values to simplify the calculation. As Equation (40) is only solved in the vertical direction, this equation is independent for each column of prisms and can be solved independently.

3.3. Numerical Discretization of the Primitive Continuity Equation

The water depth is calculated according to Equation (19), and the semi-discrete form is given as

$$\begin{aligned} \mathcal{M}_e \frac{d\mathbf{D}_{h,e}}{dt} &= -\mathcal{S}_{x,e}(\mathbf{DU})_{h,e} - \mathcal{S}_{y,e}(\mathbf{DV})_{h,e} + \\ &\mathcal{M}_e^{\partial f} \left(\left((\mathbf{DU})_{h,e} - (\mathbf{DU})_{h,e}^* \right) \cdot \mathbf{n}_x \right) + \\ &\mathcal{M}_e^{\partial f} \left(\left((\mathbf{DV})_{h,e} - (\mathbf{DV})_{h,e}^* \right) \cdot \mathbf{n}_y \right), \end{aligned} \quad (41)$$

where $\mathcal{M}_e$, $\mathcal{S}_e = [\mathcal{S}_{x,e}, \mathcal{S}_{y,e}]$ and $\mathcal{M}_e^{\partial f}$ are the mass matrix, stiff matrix and edge-mass matrix defined over the two-dimensional computational element Ω_e , respectively. Their definitions are analogous to the three-dimensional counterpart defined in Equation (29). $\mathbf{D}_{h,e}$, $(\mathbf{DU})_{h,e}$, and $(\mathbf{DV})_{h,e}$ are the nodal values of the water depth and the depth-averaged velocities in the x and y directions. $(\mathbf{DU})_{h,e} \cdot \mathbf{n}_x + (\mathbf{DV})_{h,e} \cdot \mathbf{n}_y$ and $(\mathbf{DU})_{h,e}^* \cdot \mathbf{n}_x + (\mathbf{DV})_{h,e}^* \cdot \mathbf{n}_y$ are the normal flux term and the numerical flux term defined over the boundary of the

two-dimensional computational element Ω_e and are obtained through the depth integration of their three-dimensional counterpart to preserve the local mass conservation properties.

3.4. Calculation of Vertical Velocity

According to Equations (11) and (19), the governing equation for vertical velocity ω is given as

$$\frac{\partial \omega}{\partial \sigma} = \frac{\partial(DU - Du)}{\partial x} + \frac{\partial(DV - Dv)}{\partial y}. \quad (42)$$

For the space discretization of Equation (42), suppose that for all $\phi \in V_{h,3d}$ and $\Omega_{(e,L)} \in \Omega_{h,3d}$, we have

$$\begin{aligned} \int_{\Omega_{(e,L)}} \phi \frac{\partial \omega_h}{\partial \sigma} d\mathbf{x} + \int_{\Omega_{(e,L)}} \phi \left(\frac{\partial((Du)_h - (DU)_h)}{\partial x} + \frac{\partial((Dv)_h - (DV)_h)}{\partial y} \right) d\mathbf{x} = \\ \int_{\partial\Omega_{(e,L)}^{Lat}} \phi n_x [((Du)_h - (DU)_h) - ((Du)_h - (DU)_h)^*] d\mathbf{x} + \\ \int_{\partial\Omega_{(e,L)}^{Lat}} \phi n_y [((Dv)_h - (DV)_h) - ((Dv)_h - (DV)_h)^*] d\mathbf{x} + \\ \int_{\partial\Omega_{(e,L)}^{Top}} \phi n_\sigma (\omega_h - \omega_h^*) d\mathbf{x} + \int_{\partial\Omega_{(e,L)}^{Bot}} \phi n_\sigma (\omega_h - \omega_h^*) d\mathbf{x}, \end{aligned} \quad (43)$$

where a_h is the approximate solution to a and belongs to $V_{h,3d}$. In Equation (43), the boundary of $\Omega_{(e,L)}$ is split into three parts, i.e., the lateral boundary $\partial\Omega_{(e,L)}^{Lat}$, the top boundary $\partial\Omega_{(e,L)}^{Top}$, and the bottom boundary $\partial\Omega_{(e,L)}^{Bot}$, and terms marked with an asterisk are the numerical flux terms. Here, the numerical flux $n_x(Du)_h^* + n_y(Dv)_h^*$ defined on $\partial\Omega_{(e,L)}^{Lat}$ is calculated according to the HLLC Riemann solver, and $n_x(DU)_h^* + n_y(DV)_h^*$ is the depth-averaged counterpart. At the bottom boundary, $\partial\Omega_{(e,L)}^{Bot}$, ω_h^* is taken as follows:

$$\omega_h^* = \begin{cases} \omega_{h,(e,L-1)}^{Top}, & L > 1, \\ 0, & L = 1. \end{cases} \quad (44)$$

The numerical flux is taken to be the boundary condition at the bottom in the case of the bottommost element or using the solution from the element below. Thus, the integral defined over $\partial\Omega_{(e,L)}^{Top}$ diminishes. With such a definition of numerical flux about ω_h^* given in Equation (44), we can obtain the vertical velocity ω_h layer by layer in each water column starting at the bottom and using the solution from the element below to provide an initial condition.

3.5. Time Stepping

In the discretization presented above, the ordinary differential equations Equation (28) and Equation (41) are obtained for the momentum equations and the primitive continuity equation, and they can be formulated as

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}^{\text{expl}}(\mathbf{y}(t), t) + \mathbf{f}^{\text{impl}}(\mathbf{y}(t), t), \quad (45)$$

where $\mathbf{y}(t)$ denotes the vector of all discrete degrees of freedom of a step and $\mathbf{f}^{\text{expl}}(\mathbf{y}(t), t)$ and $\mathbf{f}^{\text{impl}}(\mathbf{y}(t), t)$ denote the terms treated explicitly and implicitly. For the momentum equations, the explicit terms consist of nonlinear advection, horizontal eddy viscosity, and bottom topography source terms, while the implicit terms consist of the vertical eddy viscosity and Coriolis acceleration terms. For the primitive continuity equation, all terms are treated explicitly.

In this study, the implicit-explicit Runge–Kutta method is adapted to integrate system (45) in time. For an s-stage implicit Runge–Kutta method defined by coefficients $a_{i,j}$, b_i , and

c_i and a $\sigma' = s + 1$ stage explicit Runge–Kutta method defined by coefficients $\hat{a}_{i,j}$, $\hat{b}_i$, and $\hat{c}_i$, this time-stepping method is formulated as

$$\begin{aligned} \mathbf{y}_i &= \mathbf{y}_n + \Delta t \sum_{j=1}^i \hat{a}_{i+1,j} \hat{K}_j + \Delta t \sum_{j=1}^s a_{i,j} K_j, \quad i \in [1, s], \\ \mathbf{y}_{n+1} &= \mathbf{y}_n + \Delta t \sum_{j=1}^{\sigma'} \hat{b}_j \hat{K}_j + \Delta t \sum_{j=1}^s b_j K_j, \end{aligned} \quad (46)$$

where $\mathbf{y}_n$ are the known values at time t_n , Δt is the time step, $\hat{K}_1 = \mathbf{f}^{expl}(\mathbf{y}_n, t_n)$, $\hat{K}_{i+1} = \mathbf{f}^{expl}(t_n + \hat{c}_{i+1}\Delta t, \mathbf{y}_i)$, $K_i = \mathbf{f}^{impl}(\mathbf{y}_i, t_n + c_i\Delta t)$, and $\mathbf{y}_{n+1}$ are the unknown values at $t_{n+1} = t_n + \Delta t$. The time-stepping method adopted is derived by Ascher [38], and the corresponding coefficients are

$$\left[\begin{array}{c|c} c_i & a_{i,j} \\ \hline & b_j \end{array} \right] = \left[\begin{array}{c|ccc} \gamma & \gamma & 0 & \\ \hline 1 & 1-\gamma & \gamma & \\ & 1-\gamma & \gamma & \end{array} \right], \quad (47)$$

$$\left[\begin{array}{c|c} \hat{c}_i & \hat{a}_{i,j} \\ \hline & \hat{b}_j \end{array} \right] = \left[\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \hline \gamma & \gamma & 0 & 0 \\ 1 & \delta & 1-\delta & 0 \\ \hline \delta & 1-\delta & 0 & \end{array} \right], \quad (48)$$

with $\gamma = (2 - \sqrt{2})/2$ and $\delta = 1 - 1/(2\gamma)$. The global time-stepping algorithm from t_n to $t_{n+1} = t_n + \Delta t$ is as follows:

1. Calculate the explicit terms $\mathbf{K}_1^{expl}$ and $\mathbf{K}_{D,1}^{expl}$ for the three-dimensional momentum equations and primitive continuity equation according to the variables at t_n .
2. Calculate the intermediate water depth as

$$D_i = D_n + \Delta t \sum_{j=1}^i \hat{a}_{i+1,j} \mathbf{K}_{D,j}^{expl}, \quad i \in [1, 2] \quad (49)$$

Likewise, the intermediate conservative variables $\mathbf{U}_i$ and $\mathbf{K}_i^{impl}$ are

$$\mathbf{U}_i = \mathbf{U}_n + \Delta t \sum_{j=1}^i \hat{a}_{i+1,j} \mathbf{K}_j^{expl} + \Delta t \sum_{j=1}^i a_{i,j} \mathbf{K}_j^{impl}, \quad i \in [1, 2] \quad (50)$$

With $\mathbf{U}_i$ available, the intermediate depth-averaged momenta $(DU)_i$ and $(DV)_i$ are calculated through the depth-integration of $\mathbf{U}_i$, followed by the calculation of the intermediate vertical velocity ω_i . Later, the explicit terms $\mathbf{K}_{i+1}^{expl}$ and $\mathbf{K}_{D,i+1}^{expl}$ for the three-dimensional momentum equations and primitive continuity equation, are calculated according to the intermediate variables.

3. Calculate the water depth D_{n+1} and the conservative variables $\mathbf{U}_{n+1}$ at t_{n+1} as

$$\begin{aligned} \mathbf{U}_{n+1} &= \mathbf{U}_n + \Delta t \sum_{j=1}^3 \hat{b}_j \mathbf{K}_j^{expl} + \Delta t \sum_{j=1}^2 b_j \mathbf{K}_j^{impl}, \\ D_{n+1} &= D_n + \Delta t \sum_{j=1}^3 \hat{b}_j \mathbf{K}_{D,j}^{expl}. \end{aligned} \quad (51)$$

4. Integrate $\mathbf{U}_{n+1}$ along the water depth to obtain the final depth-averaged momenta $(DU)_{n+1}$ and $(DV)_{n+1}$, followed by the calculation of the final vertical velocity ω_{n+1} .

4. Numerical Tests

In this section, several numerical experiments are conducted to verify the performance of the developed model. For all the experiments, the acceleration due to gravity, g , is set to 9.81 m/s^2 . In addition, all the experiments are run on an Intel Xeon E5-2620 processor with 16 GB of internal memory. Our program is parallelized using OpenMP to run on six cores.

4.1. Manufactured Solution

This test is used to verify the convergence property of the developed model. As the 3D SWEs are a non-linear system, designing a nontrivial function satisfying the equations and the boundary conditions at the same time is not a trivial task. Following the philosophy presented by Salari and Knupp [39], we manually give the solution that satisfies the continuity equation and solve a Dirichlet problem for a modified system that includes a forcing term.

For this test, the Coriolis acceleration terms are not included, the horizontal eddy viscosity coefficient K_h is taken as $0.1 \text{ m}^2 \text{ s}^{-1}$, the vertical eddy viscosity coefficient K_m is taken as $0.01 \text{ m}^2 \text{ s}^{-1}$, and the analytical solution is given as follows:

$$\begin{aligned} b(x, y, t) &= -(2 - 0.005(x + y)), \\ \eta(x, y, t) &= 0.01(\sin(0.01(x + t)) + \sin(0.01(y + t))), \\ u(x, y, \sigma, t) &= 0.1(D(1 + \sigma))^3 \sin(0.01(x + t)), \\ v(x, y, \sigma, t) &= 0.1(D(1 + \sigma))^3 \sin(0.01(x + t)). \end{aligned} \quad (52)$$

The vertical velocity at an arbitrary location is obtained through the integration of the continuity equation, Equation (11), from the bottom to the studied location. The horizontal projection of the horizontal domain is $\Omega_{2d} = [0, 100] \times [0, 100]$, and the whole simulation lasts for 86.4 s.

Tests are run with successively refined meshes. For the coarsest mesh, there is one element in the horizontal direction and one layer in the vertical direction. We refine the mesh up to four times by partitioning each quadrilateral prism into eight and compute the L_2 error of the numerical results against the analytical solutions as

$$L_2 = \frac{1}{\sum_e \int_{\Omega_e} dx} \left(\sum_e \int_{\Omega_e} (Q_e^S(x) - Q_e^{exact}(x))^2 dx \right)^{1/2}, \quad (53)$$

where $Q_e^S(x)$ is the numerical result and $Q_e^{exact}(x)$ is the analytical solution. To measure the convergence properties of the model, we further define the convergence rate (CR) as follows

$$CR = \log_2(L_2^m / L_2^{m-1}) / \log_2(L^m / L^{m-1}), \quad (54)$$

where L_2^m and L_2^{m-1} are the numerical errors on two successive refined meshes and L^m and L^{m-1} are the characteristic lengths of the meshes, which are set to be the longest edge length of the quadrilateral prisms. Tables 3 and 4 give the L_2 error for each variable and the corresponding CR for $N_h = N_v = 1$ and 2, respectively. N_e denotes the number of elements in the horizontal direction and N_L the number of vertical layers. We note that for water depth D , horizontal momenta Du and Dv converge at the optimal rate, i.e., $CR = 2$ for the former case and 3 for the latter, whereas the vertical velocity ω converges at the suboptimal rate for both cases. We owe this to the fact that the vertical velocity is computed from the numerical results for Du , Dv , and their depth-averaged counterparts using the continuity Equation (42), and the CR for vertical velocity would be influenced as the horizontal momenta are not exact, as pointed out by Dawson and Aizinger [25].

Table 3. $N_h = N_v = 1$, the L_2 error for each variable, and the corresponding CR.

N_e	N_L	$Er(D)$	$CR(D)$	$Er(Du)$	$CR(Du)$	$Er(Dv)$	$CR(Dv)$	$Er(\omega)$	$CR(\omega)$
1	1	0.00033		0.00169		0.00169		5.224×10^{-5}	
4	2	6.125×10^{-5}	2.42	0.00031	2.47	0.00031	2.47	1.277×10^{-5}	2.03
16	4	1.340×10^{-5}	2.19	6.086×10^{-5}	2.33	6.086×10^{-5}	2.33	3.748×10^{-6}	1.77
64	8	3.170×10^{-6}	2.08	1.350×10^{-5}	2.17	1.350×10^{-5}	2.17	1.239×10^{-6}	1.60
256	16	7.617×10^{-7}	2.06	3.083×10^{-6}	2.13	3.083×10^{-6}	2.13	4.527×10^{-7}	1.45

Table 4. $N_h = N_v = 2$, the L_2 error of each variable, and the corresponding CR.

N_e	N_L	$Er(D)$	$CR(D)$	$Er(Du)$	$CR(Du)$	$Er(Dv)$	$CR(Dv)$	$Er(\omega)$	$CR(\omega)$
1	1	9.997×10^{-6}		0.00015		0.00015		4.95×10^{-6}	
4	2	1.353×10^{-6}	3.10	1.804×10^{-5}	3.10	1.804×10^{-5}	3.10	5.71×10^{-7}	3.11
16	4	1.703×10^{-7}	3.04	2.187×10^{-6}	3.04	2.187×10^{-6}	3.04	1.13×10^{-7}	2.33
64	8	2.196×10^{-8}	3.05	2.649×10^{-7}	3.05	2.649×10^{-7}	3.05	3.00×10^{-8}	1.91
256	16	3.276×10^{-9}	3.02	3.264×10^{-8}	3.02	3.264×10^{-8}	3.02	6.99×10^{-9}	2.10

Figure 2 shows the relation between the L_2 error and DOFs and the normalized CPU time. Figure 2a indicates that the developed model is convergent, as the L_2 error for each variable diminishes in both cases as the computational mesh is refined. Figure 2b indicates that we can achieve a given error tolerance with fewer DOFs on a coarser mesh when a higher approximation order is adapted. For example, when the error tolerance about water depth D is taken as 1.0×10^{-6} , the normalized CPU time is about 0.033 when the approximation order is $N_h = N_v = 2$, whereas it is about 0.18 when the approximation order is $N_h = N_v = 1$.

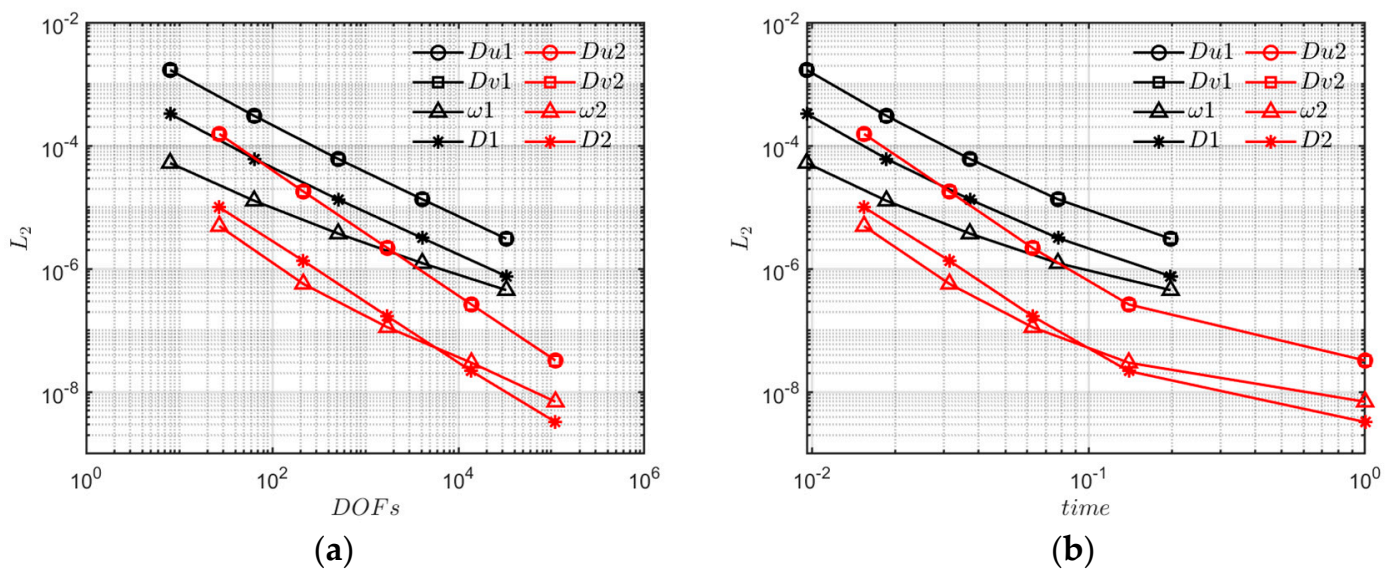


Figure 2. L^2 error versus DOFs and normalized CPU time for water depth D , vertical velocity ω , and horizontal momenta Du and Dv at different approximation orders ($N_h = N_v = 1$ (black), $N_h = N_v = 2$ (red)) on a sequence of meshes. (a) L^2 error versus DOFs; (b) L^2 error versus normalized CPU time.

4.2. Tide Wave Propagation in a Semi-Closed Bay

This test is used to indicate the accuracy of the developed model for the simulation of both the horizontal and vertical velocity fields. Figure 3 shows the horizontal locations of Gauge points A (2500, 5000), B (52,500, 5000), and C (92,500, 5000) and a sketch of the

horizontal domain. This domain is 100,000 m long and 10,000 m wide and is partitioned by a uniform quadrilateral element with a size of 1000 m in both directions. The still water depth is $h = 12$ m. A cosine tidal wave with amplitude $A = 0.25$ m and period $T = 12$ h is imposed at the left boundary, i.e., $x = 0$, whereas the other boundaries are all treated as slip walls. Both the eddy viscosity terms and the Coriolis acceleration terms are not included.

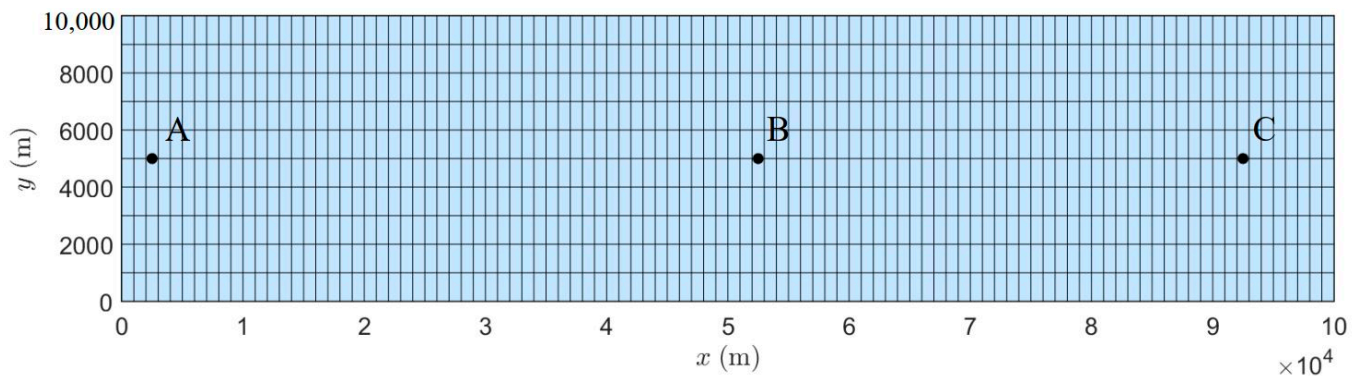


Figure 3. Location of the gauge points and sketch of the two-dimensional computational domain.

According to Neuman and Pierson [40], the analytical solution of the surface elevation η , the horizontal velocities u and v , and the vertical velocity w at a point $P(x, y, z)$ at time t are given as

$$\begin{aligned}
 \eta(x, t) &= A \left[\cos \left(\frac{\omega'}{\sqrt{gh}} (L - x) \right) / \cos \left(\frac{\omega'}{\sqrt{gh}} L \right) \right] \cos(\omega' t), \\
 u(x, t) &= - \left[\left(A \frac{\sqrt{gh}}{h} \right) / \cos \left(\frac{\omega'}{\sqrt{gh}} L \right) \right] \sin \left(\frac{\omega'}{\sqrt{gh}} (L - x) \right) \sin(\omega' t), \\
 v(x, t) &= 0, \\
 w(z, t) &= - \frac{h+z}{\eta+h} A \omega' \sin(\omega' t) \left[\cos \left(\frac{\omega'}{\sqrt{gh}} (L - x) \right) / \cos \left(\frac{\omega'}{\sqrt{gh}} L \right) \right] \\
 &\quad - \frac{h+z}{\eta+h} \frac{A^2 \omega'}{h} \frac{1}{\cos^2(\omega' L / \sqrt{gh})} \cos(\omega' t) \sin(\omega' t).
 \end{aligned} \tag{55}$$

where $\omega' = 2\pi/T$ is the frequency. For the initial condition, all the velocity components are set to zero, and the surface elevation is set according to the analytical solution. The number of vertical layers is set to 12, and we run this simulation for 24 h.

Figure 4 gives the comparison between the numerical result and the analytical solution for surface elevation η , horizontal velocity u , and vertical velocity w at Gauges A, B, and C, and the vertical coordinate for the three Gauges is $z = -2.5$ m. It can be seen that the numerical solution is generally in good agreement with the analytical solution, indicating that the three-dimensional model developed in this study has good accuracy in the simulation of water level and velocity.

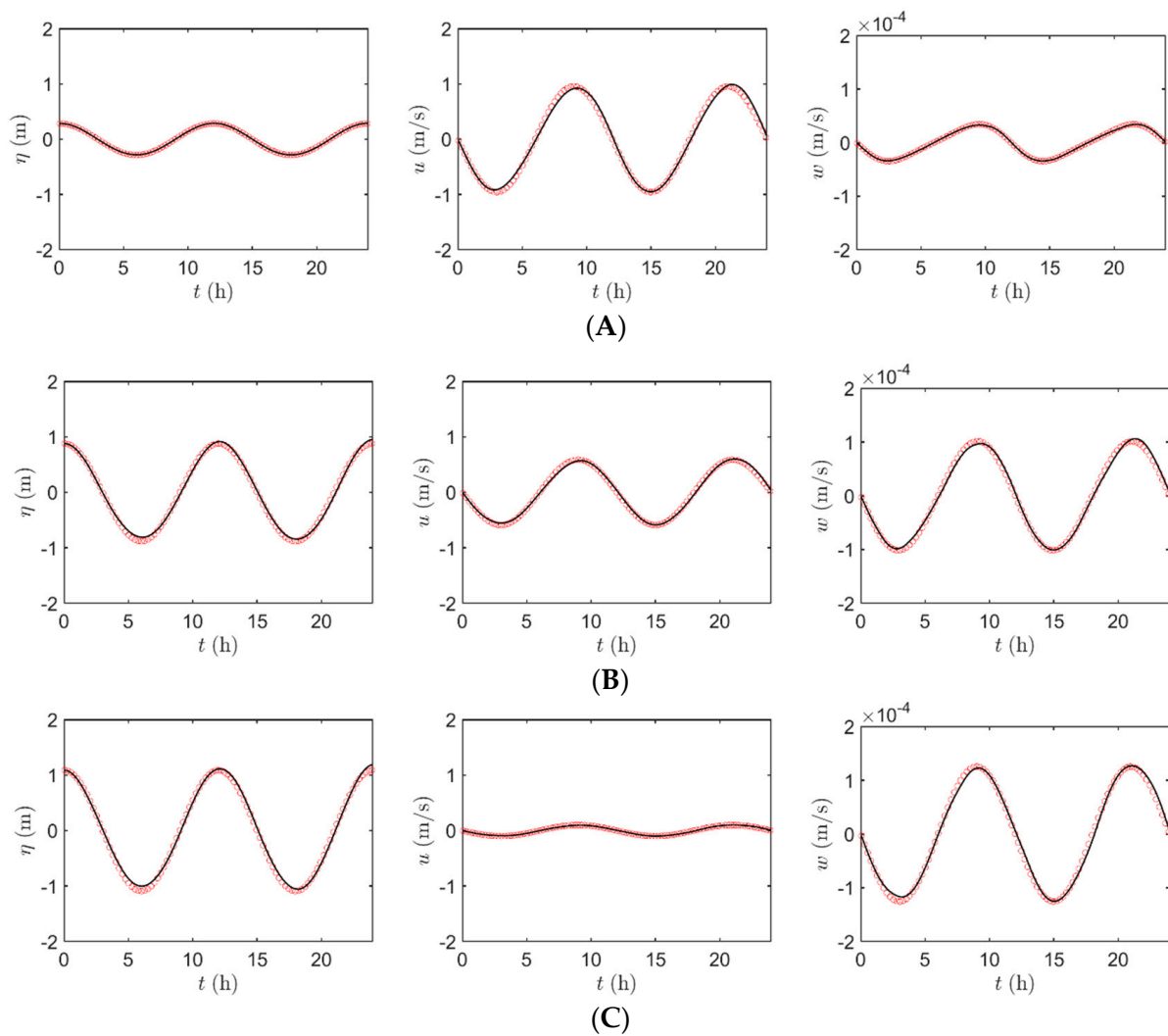


Figure 4. Comparison between the numerical result (black line) and the analytical solution (red circle) for surface elevation η , horizontal velocity u , and vertical velocity w at Gauges (A–C).

4.3. Wind-Induced Water Circulation

In this test, the wind-driven circulation in a rectangular closed basin is simulated. Following Liu et al. [41], we run this case to test the accuracy of the proposed numerical model in predicting the vertical stratified circulation.

With a homogeneous Dirichlet boundary condition adapted at the bottom boundary, Koutitas et al. [42] derived the analytical solution of the horizontal velocity in the center of the basin as

$$u(\sigma) = \frac{\tau_{sx}(\sigma + 1)h(3\sigma + 1)}{4K_v\rho_0}, \sigma \in [-1, 0] \quad (56)$$

where h is the still water depth of the basin. Likewise, Huang [43] also derived that when the Neumann boundary condition is adapted, the result reads

$$u(\sigma) = \frac{1}{6K_v}g\frac{\partial\eta}{\partial x}h^2(3\sigma^2 - 1) + \frac{\tau_{sx}h}{2\rho_0K_v}(2\sigma + 1), \sigma \in [-1, 0] \quad (57)$$

$$\frac{\partial\eta}{\partial x} = \frac{3}{2}\frac{\tau_{sx}}{\rho_0gh}\frac{2K_v + C_f|u_c|h}{3K_v + C_f|u_c|h'}$$

where $|u_c|$ is the velocity magnitude at the center of the bottommost element, and $C_f|u_c| = 0.005 \text{ m/s}$ in this study. In this part, these two situations are simulated.

This case is set as follows: the basin is 2000 m long and 800 m wide, $h = 10$ m, and $\tau_{sx} = 1.5$ Pa. The Coriolis acceleration and horizontal eddy viscosity terms are ignored, and the vertical eddy viscosity coefficient K_m is set to be $0.001 \text{ m}^2 \text{ s}^{-1}$; the whole simulation lasts for 3600 s.

Figure 5a,b shows the comparison between the numerical results and the analytical solution for the vertical distribution of horizontal velocity at the center of the basin. The comparison indicates that the numerical results match well with the analytical solution for both cases.

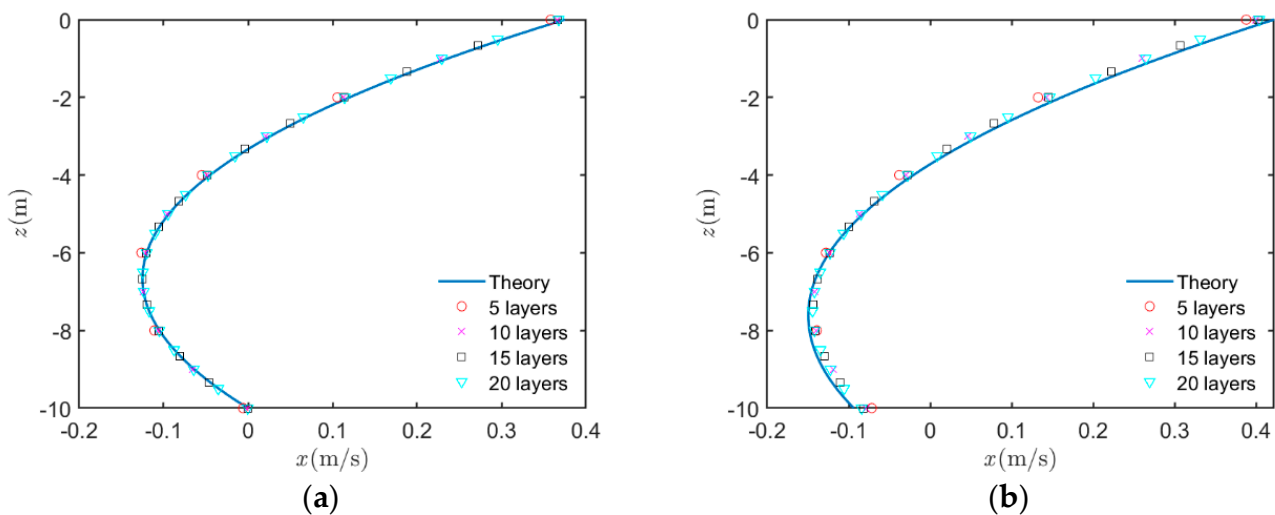


Figure 5. Comparison between the numerical results and the analytical solution for the vertical distribution of horizontal velocity at the center of the basin. (a) Homogeneous Dirichlet boundary condition; (b) Neumann boundary condition.

We also run the simulations with different numbers of vertical layers (N_L). To quantitatively measure the difference between different simulations, we define the root mean square error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^{N_T} (u_{a,j} - u_j)^2}{N_T}}, \quad (58)$$

where N_T is the number of simulated results, u_j is the numerical result at point j , and $u_{a,j}$ is the analytical solution at the same point. Table 5 gives the RMSE for different simulations. We find that the more vertical layers there are, the better the result will be for each case.

Table 5. RMSE for simulations with different vertical layers and different boundary conditions.

N_L	5	10	15	20
Homogeneous Dirichlet boundary condition				
RMSE	0.0107	0.0051	0.0041	0.0037
Neumann boundary condition				
RMSE	0.0219	0.0138	0.0114	0.0103

4.4. Generation of the Ekman Profile

Following Trahan et al. [44], we run this test to check whether the developed model can generate the Ekman layer, where the Coriolis acceleration terms, the pressure gradient, and the turbulent drag are in balance. The analytic solution for the Ekman velocity profile was developed by Ekman [45] and is expressed as

$$\begin{aligned}
 u &= u_0 e^{(-z/D_f)} \cos\left(\frac{\pi}{4} - \frac{z}{D_f}\right), \\
 v &= v_0 e^{(-z/D_f)} \sin\left(\frac{\pi}{4} - \frac{z}{D_f}\right), \\
 v_0 &= \frac{\tau_{sx}}{\rho(K_v f)^{0.5}}, \\
 D_f &= \left(\frac{2K_v}{f}\right)^{0.5}, \\
 f &= 2\omega \sin \theta.
 \end{aligned} \tag{59}$$

where D_f is the folding depth, $K_v = 0.1 \text{ m}^2/\text{s}$, $\tau_{sx} = 0.1 \text{ Pa}$, and θ is the geophysical latitude (45°). Theoretically, the flow direction at the surface would rotate counterclockwise 45 degrees to that of the wind stress.

The computational domain is 40,000 km long and 40,000 km wide and is partitioned by a uniform quadrilateral element with a size of 2000 km in both directions. A constant depth of 200 m is set, and 50 vertical layers are used. The bottom drag coefficient C_f is set to 0.0025. We run this simulation for 200 days. Figure 6 gives the comparison of the analytic solution and the computed Ekman profile. The numerical results accurately match the analytic solution and individual velocity components with an average error of less than 0.001 m/s.

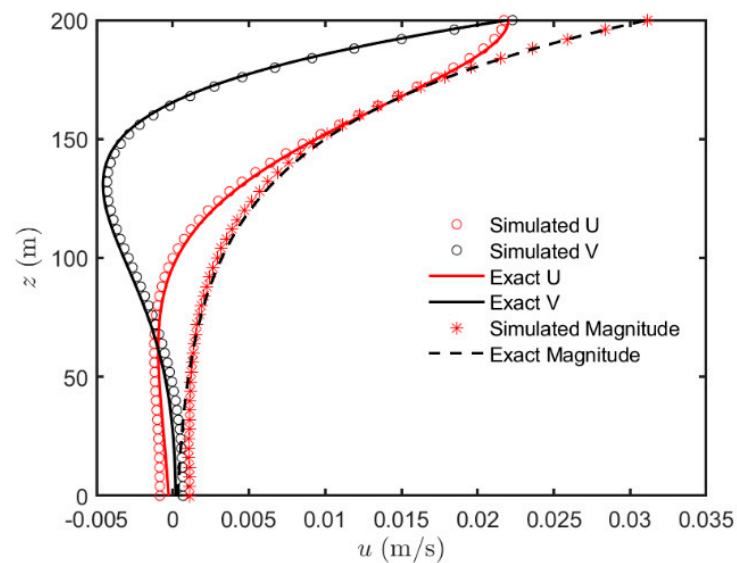


Figure 6. Comparison between the numerical results and the analytical solution for individual velocity components and velocity magnitude for the Ekman profile.

4.5. Tidal Flow in Bohai Bay

For the last case, we simulate tidal flow in the western region of Bohai Bay to test the ability of the developed model for the simulation of actual tidal wave motion problems. The Bohai Bay is one of the three major bays of the Bohai Sea in China. It is located in the west of the Bohai Sea and connected with Hebei and Shandong Province and Tianjin City. It's a typical semi-closed sea area. The average tidal range in Bohai Bay is 2.5 m, and its tidal current is irregularly semi-diurnal. In recent years, a lot of port and reclamation projects in Bohai Bay have been carried out, which has led to a large number of large-scale engineering structures. The accurate simulation of the tidal current field is of great significance for the convection–diffusion process of temperature, salt, pollutants, etc., as well as the sediment transport in Bohai Bay.

The domain geometry for this case is shown in Figure 7a; the still water depth ranges from 38 m at the east open boundary to 2 m at the coast. There are five measuring gauges

in the computational domain, and the distribution of these points is shown in Figure 7b. The computational domain is partitioned by 14,627 triangle grids, and the edge length of the finite element mesh ranges from 2 km to 100 m. Five equidistant layers are used in the vertical direction. Wetting/drying is not accounted in the present study.

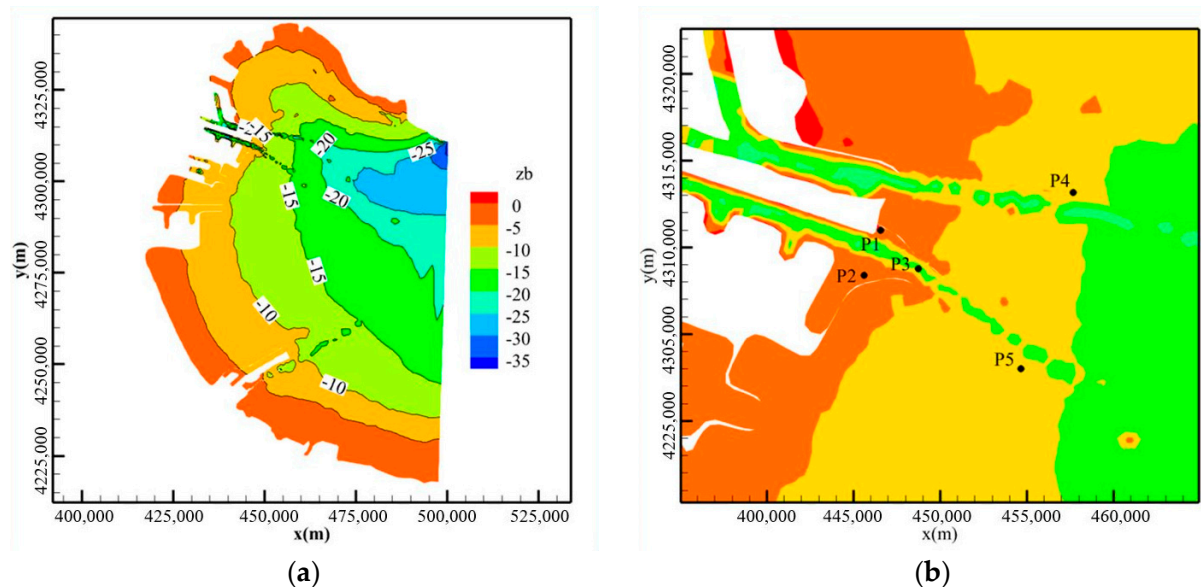


Figure 7. Computational domain geometry for tidal flow in the west region of Bohai Bay. (a) Computational domain geometry; (b) Gauge points.

The tidal elevations at the east open boundaries were provided by Chinatide [46], which is a tidal forecast system through nine harmonic constants of Q_1 , P_1 , O_1 , K_1 , N_2 , M_2 , S_2 , K_2 , and S_a . The Coriolis parameter was taken as $9.1557 \times 10^{-5} \text{ s}^{-1}$, and the horizontal eddy viscosity coefficient was set to $70 \text{ m}^2 \text{ s}^{-1}$. The vertical eddy viscosity coefficient was modelled using the $k - \epsilon$ turbulence closures provided by GOTM [47], and the coupling between GOTM and the hydrodynamic model is established following the same philosophy presented by Tuomas et al. [32]. The bottom roughness length is set as 0.1 mm. Field measurements from 12/08/2018 to 12/09/2018 are used to verify the developed model.

Figure 8 shows the flow field at the surface and the middle and bottom layers at the time of flood tide (left) and ebb tide (right), respectively. As we can see, there is little difference between the velocity fields of the surface and the middle layers. Because of the bottom friction, the velocity of the bottom layer is smaller than that of the surface and the middle layers, which is about 60% of the magnitude of the latter two. The results of the three-dimensional velocity field show that the model can reflect the three-dimensional characteristics of tidal flow.

Figure 9 shows the comparison between the simulated surfaced elevation and the field measurements at Gauges P1 and P2. The simulated data are quantitatively very close to the field measurements, and the tidal phasing is virtually identical. The left column of Figure 10 gives the comparison between the simulated velocity and the field measurements at gauges P3, P4, and P5. Small discrepancies exist between the simulated velocity and the field measurements. This is especially the case for P5, where the maximum simulated velocity is about 15% smaller than that of the field measurements. It can be attributed to that the computational mesh used in this simulation is relatively coarse, and the wetting/drying is not considered. Although differences exist, the simulated data are in phase with the field measurements as the comparison between the simulated flow direction and the field measurements shows in the right column of Figure 10. Generally, the developed hydrodynamic model can be used for the simulation of tidal wave motion in realistic situations.

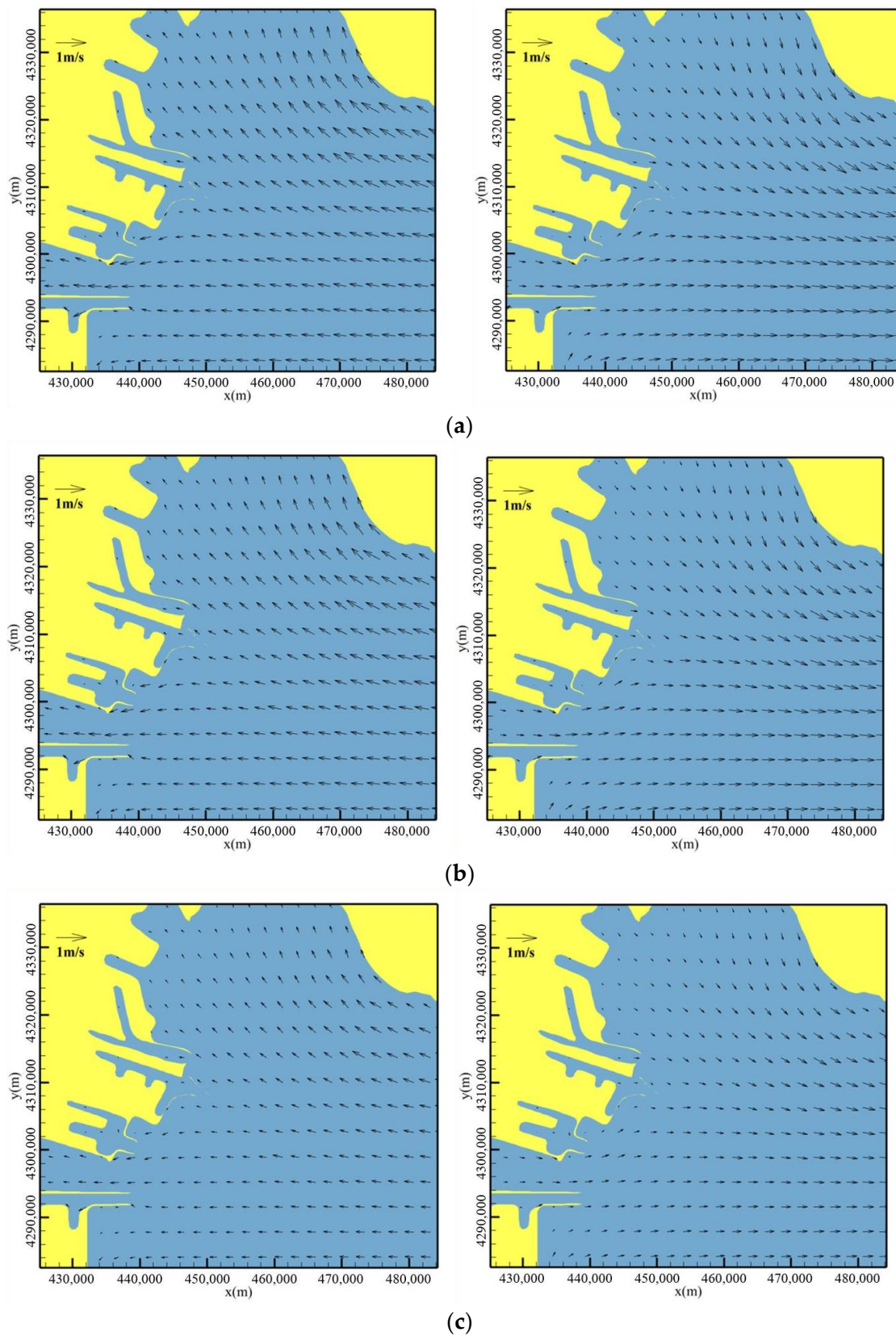


Figure 8. Flow field at different levels at the flood tide (left column) and the ebb tide (right column). (a) surface ($\sigma = 0$); (b) middle ($\sigma = -0.5$); (c) bottom ($\sigma = -1.0$).

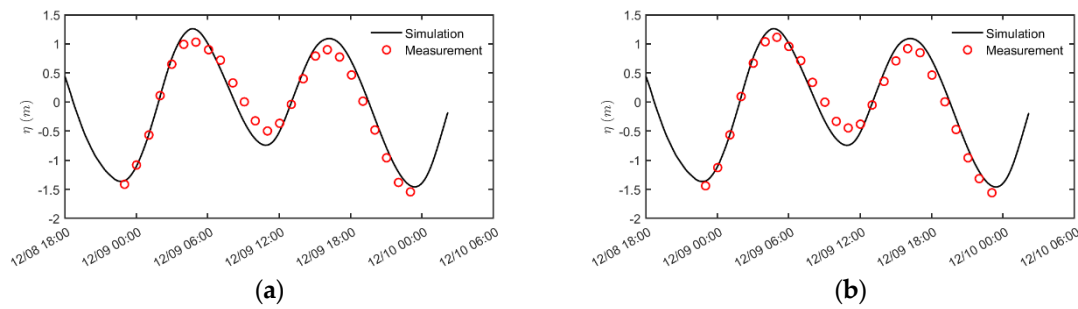


Figure 9. Comparison between the numerical results and the measured surface elevation at Gauges P1 and P2 for the Bohai Bay case. (a) P1; (b) P2.

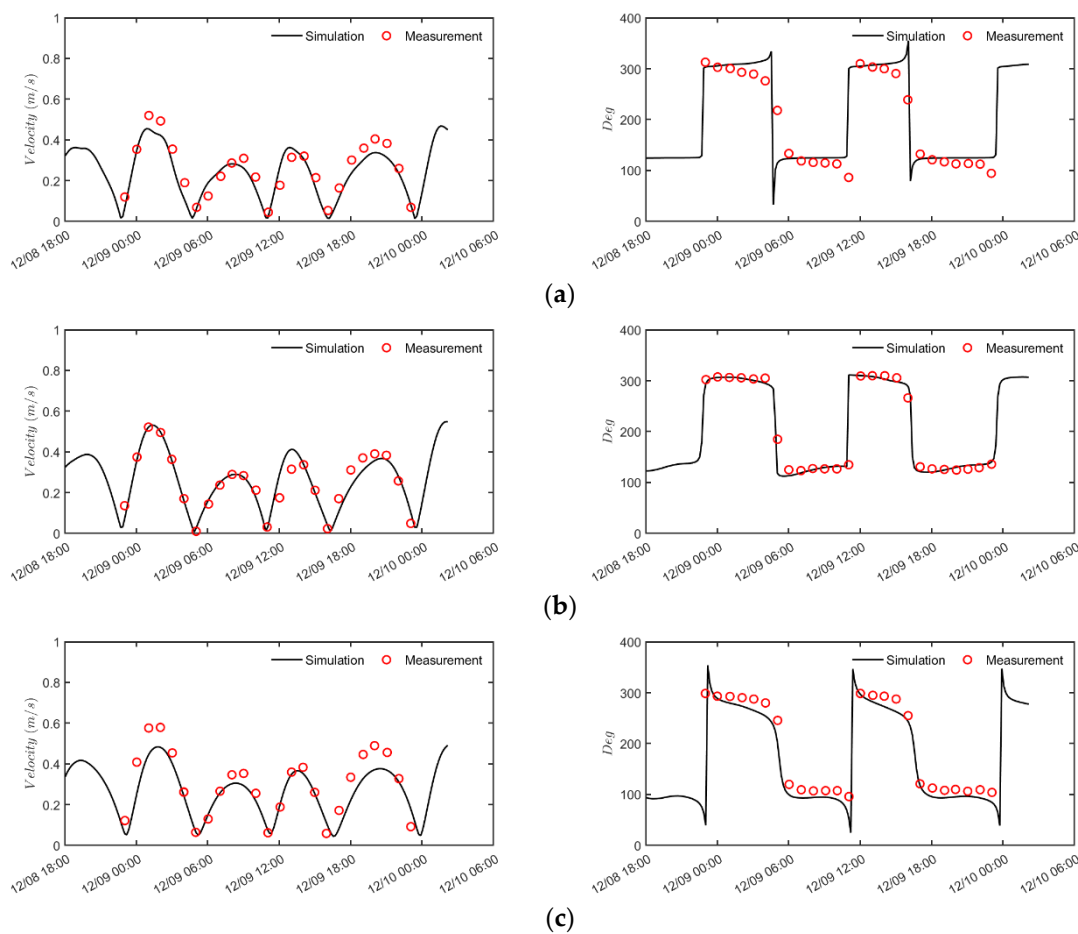


Figure 10. Comparison between the simulated results and the measured data for velocity (left column) and flow direction (right column) at Gauges P3, P4, and P5 for the Bohai Bay case. (a) P3; (b) P4; (c) P5.

5. Conclusions

In this study, a three-dimensional hydrodynamic model based on the Discontinuous Galerkin method is developed. The σ -coordinate system is used in the vertical direction to circumvent the mismatch of lateral faces due to the discontinuous nature of the solution between different elements. Non-linear advective terms are included, and the model is marched in time with the implicit-explicit Runge–Kutta method. The vertical eddy viscosity and the Coriolis acceleration terms are discretized with the symmetric interior penalty discontinuous Galerkin method and are treated implicitly, while others are treated explicitly.

The developed model is validated with a series of numerical tests. The case with the manufactured solution indicates that the model is convergent, and the L_2 error about the variables diminishes as we increase the approximation order or refine the computational mesh. Calculating the CR indicates that water depth D and horizontal momenta Du and Dv can all converge at the optimal order, whereas the vertical velocity converges in a suboptimal order. This case also indicates that we can obtain a better numerical result on a computational mesh with fewer DOFs when a higher approximation is adapted. Cases of tide-induced three-dimensional flow in a semi-closed bay, wind-induced water circulation, and the generation of the Ekman profile indicate that the developed model can well simulate the velocity field in both horizontal and vertical directions, can reasonably simulate the vertical stratified circulation, and can simulate water motion under the combined effect of wind stress and Coriolis acceleration terms. For the Bohai Bay case, the simulated surface elevation, velocity, and flow direction match reasonably well with the field measurements, indicating the model can be used for the simulation of tidal wave motion in realistic situations.

More applications of the developed model and development of the wetting/drying treatment and the baroclinic model will be presented in forthcoming work.

Author Contributions: Formal analysis, G.R.; Funding acquisition, Q.Z.; Methodology, G.R.; Software, G.R. and Z.C.; Supervision, Q.Z.; Validation, G.R.; Visualization, G.R.; Writing—original draft, G.R.; Writing—review and editing, Q.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This study was financially supported by the Joint Funds of the National Natural Science Foundation of China (Grant No. U1906231), the Natural Science Foundation of Tianjin (Grant No. 19JCZDJC40200).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Lee, J.; Lee, J.; Yun, S.-L.; Kim, S.-K. Three-Dimensional Unstructured Grid Finite-Volume Model for Coastal and Estuarine Circulation and Its Application. *Water* **2020**, *12*, 2752. [\[CrossRef\]](#)
2. Blaise, S.; Comblen, R.; Legat, V.; Remacle, J.-F.; Deleersnijder, E.; Lambrechts, J. A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. *Ocean. Dyn.* **2010**, *60*, 1371–1393. [\[CrossRef\]](#)
3. Li, L.; Zhang, Q. Development of an efficient wetting and drying treatment for shallow-water modeling using the quadrature-free Runge-Kutta discontinuous Galerkin method. *Int. J. Numer. Methods Fluids* **2021**, *93*, 314–338. [\[CrossRef\]](#)
4. Ran, G.; Zhang, Q.; Li, L. A depth-integrated non-hydrostatic model for nearshore wave modelling based on the discontinuous Galerkin method. *Ocean. Eng.* **2021**, *232*, 108661. [\[CrossRef\]](#)
5. Di Pietro, D.A.; Marche, F. Weighted interior penalty discretization of fully nonlinear and weakly dispersive free surface shallow water flows. *J. Comput. Phys.* **2018**, *355*, 285–309. [\[CrossRef\]](#)
6. Shchepetkin, A.F.; McWilliams, J.C. The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean. Model.* **2005**, *9*, 347–404. [\[CrossRef\]](#)
7. Mellor, G. *Users Guide for a Three-Dimensional, Primitive Equation, Numerical Ocean Model*; Princeton University: Princeton, NJ, USA, 2004.
8. Griffies, S.M. *Elements of the Modular Ocean Model (MOM)*; GFDL Ocean Group Technical Report; NOAA: Princeton, NJ, USA, 2012.
9. Burchard, H.; Bolding, K. *GETM: A General Estuarine Transport Model. Scientific Documentation*; European Commission Technical Report EUR 20253 EN; European Commission: Brussels, Belgium, 2002; p. 157.
10. Casulli, V.; Cheng, R.T. Semi-implicit finite difference methods for three-dimensional shallow water flow. *Int. J. Numer. Methods Fluids* **1992**, *15*, 629–648. [\[CrossRef\]](#)
11. Deltares Systems. *User Manual of Delft3D-FLOW—Simulation of Multi-Dimensional Hydrodynamic Flows and Transport Phenomena, Including Sediments*; Report of Delft Hydraulics; Deltares: Delft, The Netherlands, 2003; p. 497.
12. Hamrick, J.; Wu, T. Computational Design and Optimization of the EFDC/HEM3D Surface Water Hydrodynamic and Eutrophication. In *Next Generation Environmental Models and Computational Methods*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1997; pp. 143–161.
13. Blumberg, A.F.; Mellor, G.L. *A Description of a Three-Dimensional Coastal Ocean Circulation Model*; AGU: Washington, DC, USA, 1987; pp. 1–16.

14. Luyten, P.J.; Jones, J.E.; Proctor, R. A numerical study of the long-and short-term temperature variability and thermal circulation in the North Sea. *J. Phys. Oceanogr.* **2003**, *33*, 37–56. [\[CrossRef\]](#)
15. Alistair, A.; Jean-Michel, C.; Stephanie, D.; Constantinos, E.; David, F.; Gael, F.; Baylor, F.-K.; Patrick, H.; Chris, H.; Ed, H. *Mitgcm User Manual*; MIT: Cambridge, MA, USA, 2018.
16. Martins, F.; Leitão, P.; Silva, A.; Neves, R. 3D modelling in the Sado estuary using a new generic vertical discretization approach. *Oceanol. Acta* **2001**, *24*, 51–62. [\[CrossRef\]](#)
17. Chen, C.; Beardsley, R.; Cowles, G.; Qi, J.; Lai, Z.; Gao, G.; Stuebe, D.; Xu, Q.; Xue, P.; Ge, J. *An Unstructured-Grid, Finite-Volume Community Ocean Model: FVCOM User Manual*; SMAST/UMASSD Technical Report-13-0701; MIT: Cambridge, MA, USA, 2012; p. 404.
18. Casulli, V.; Walters, R.A. An unstructured grid, three-dimensional model based on the shallow water equations. *Int. J. Numer. Methods Fluids* **2000**, *32*, 331–348. [\[CrossRef\]](#)
19. Nan, Z.; Sheng, J.; Congfang, A.; Weiye, D. An integrated water-conveyance system based on Web GIS. *J. Hydroinform.* **2018**, *20*, 668–686. [\[CrossRef\]](#)
20. Wang, Q. The Finite Element Ocean Model and Its Aspect of Vertical Discretization. Ph.D. Thesis, Universität Bremen, Bremen, Germany, 2007.
21. Zhang, Y.; Baptista, A.M. SELFE: A semi-implicit Eulerian–Lagrangian finite-element model for cross-scale ocean circulation. *Ocean. Model.* **2008**, *21*, 71–96. [\[CrossRef\]](#)
22. Luettich, R.A.; Westerink, J.J.; Scheffner, N.W. ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves, Coasts, and Estuaries. Report 1, Theory and methodology of ADCIRC-2DD1 and ADCIRC-3DL; Department of the Army, US Army Corps of Engineers Waterways Experiment Station, Coastal Engineering Research Center: Vicksburg, MS, USA, 1992.
23. Zhang, Y.J.; Ye, F.; Stanev, E.V.; Grashorn, S. Seamless cross-scale modeling with SCHISM. *Ocean. Model.* **2016**, *102*, 64–81. [\[CrossRef\]](#)
24. Hervouet, J.-M. *Hydrodynamics of Free Surface Flows: Modelling with the Finite Element Method*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
25. Dawson, C.; Aizinger, V. A discontinuous Galerkin method for three-dimensional shallow water equations. *J. Sci. Comput.* **2005**, *22*, 245–267. [\[CrossRef\]](#)
26. Aizinger, V.; Dawson, C. The local discontinuous Galerkin method for three-dimensional shallow water flow. *Comput. Methods Appl. Mech. Eng.* **2007**, *196*, 734–746. [\[CrossRef\]](#)
27. Aizinger, V.; Proft, J.; Dawson, C.; Pothina, D.; Negusse, S. A three-dimensional discontinuous Galerkin model applied to the baroclinic simulation of Corpus Christi Bay. *Ocean. Dyn.* **2013**, *63*, 89–113. [\[CrossRef\]](#)
28. Blaise, S. Development of a Finite Element Marine Model. Ph.D. Thesis, UCL-Université Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, 2009.
29. Comblen, R. Discontinuous Finite-Element Methods for Two-and Three-Dimensional Marine Flows. Ph.D. Thesis, UCL-Université Catholique de Louvain, Ottignies-Louvain-la-Neuve, Belgium, 2010.
30. Gandham, R. High Performance High-Order Numerical Methods: Applications in Ocean Modeling. Ph.D. Thesis, Rice University, Houston, TX, USA, 2015.
31. Conroy, C.J.; Kubatko, E.J. Hp discontinuous Galerkin methods for the vertical extent of the water column in coastal settings part I: Barotropic forcing. *J. Comput. Phys.* **2016**, *305*, 1147–1171. [\[CrossRef\]](#)
32. Kärnä, T.; Legat, V.; Deleersnijder, E.; Burchard, H. Coupling of a discontinuous Galerkin finite element marine model with a finite difference turbulence closure model. *Ocean. Model.* **2012**, *47*, 55–64. [\[CrossRef\]](#)
33. Phillips, N.A. A coordinate system having some special advantages for numerical forecasting. *J. Atmos. Sci.* **1957**, *14*, 184–195. [\[CrossRef\]](#)
34. Toro, E.F. *Shock-Capturing Methods for Free-Surface Shallow Flows*; John Wiley and Sons: Hoboken, NJ, USA, 2001.
35. Hesthaven, J.S.; Warburton, T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
36. Cockburn, B.; Shu, C.-W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **1998**, *35*, 2440–2463. [\[CrossRef\]](#)
37. Di Pietro, D.A.; Ern, A. *Mathematical Aspects of Discontinuous Galerkin Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 69.
38. Ascher, U.M.; Ruuth, S.J.; Spiteri, R.J. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* **1997**, *25*, 151–167. [\[CrossRef\]](#)
39. Salari, K.; Knupp, P. *Code Verification by the Method of Manufactured Solutions*; SAND2000-1444; Sandia National Laboratories: Albuquerque, NM, USA, 2000.
40. Neumann, G.; Pierson, W.J., Jr. *Principles of Physical Oceanography*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1966.
41. Liu, X.; Mohammadian, A.; Sedano, J.Á.I.; Kurganov, A. Three-dimensional shallow water system: A relaxation approach. *J. Comput. Phys.* **2017**, *333*, 160–179. [\[CrossRef\]](#)
42. Koutitas, C.; O'Connor, B. Modeling three-dimensional wind-induced flows. *J. Hydraul. Div. ASCE* **1980**, *106*, 1843–1865. [\[CrossRef\]](#)

43. Huang, W. Three Dimensional Numerical Modeling Of Circulation And Water Quality Induced By Combined Sewage Overflow Discharges. Ph.D. Thesis, Department of Ocean Engineering, University of Rhode Island, Kingston, RI, USA, 1993.
44. Trahan, C.; Savant, G.; Berger, R.; Farthing, M.; McAlpin, T.; Pettey, L.; Choudhary, G.; Dawson, C.N. Formulation and application of the adaptive hydraulics three-dimensional shallow water and transport models. *J. Comput. Phys.* **2018**, *374*, 47–90. [[CrossRef](#)]
45. Price, J.F.; Weller, R.A.; Schudlich, R.R. Wind-driven ocean currents and Ekman transport. *Science* **1987**, *238*, 1534–1538. [[CrossRef](#)]
46. Li, M.; Zheng, J. Introduction to Chinatide software for tide prediction in China seas. *J. Waterw. Harb.* **2007**, *28*, 65–68. (In Chinese)
47. Burchard, H.; Bolding, K.; Villarreal, M.R. *GOTM, a General Ocean Turbulence Model: Theory, Implementation and Test Cases*; Technical Report EUR 18745 EN; European Commission: Brussels, Belgium, 1999.

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.