



The discontinuous Galerkin coastal and estuarine modelling system (DGCEMS v1.0.0): a three-dimensional, nonsplit-mode, implicit–explicit Runge–Kutta hydrostatic model

Zereng Chen^{1,2,3}, Qinghe Zhang¹, Guoquan Ran⁴, and Yang Nie¹

¹State Key Laboratory of Hydraulic Engineering Intelligent Construction and Operation, Tianjin University, Tianjin, 300350, China

²The National Key Laboratory of Water Disaster Prevention, Hohai University, Nanjing, 210024, China

³College of Harbour, Coastal and Offshore Engineering, Hohai University, Nanjing, 210024, China

⁴School of River and Ocean Engineering, Chongqing Jiaotong University, Chongqing, 400074, China

Correspondence: Qinghe Zhang (qhzhzhang@tju.edu.cn) and Guoquan Ran (2015205064@tju.edu.cn)

Received: 17 December 2024 – Discussion started: 12 February 2025

Revised: 30 July 2025 – Accepted: 30 July 2025 – Published: 14 October 2025

Abstract. Numerical methods of discontinuous Galerkin (DG) discretisation for coastal ocean modelling have advanced significantly, but there are still challenges in accurately simulating phenomena such as wetting and drying processes and baroclinic flows in coastal and estuarine regions. This study develops a novel three-dimensional (3D) coastal and estuarine modelling system named DGCEMS, using a quadrature-free nodal DG method. The model adopts σ coordinates, employs a nonsplit-mode framework, and integrates a semi-implicit Runge–Kutta scheme. A series of numerical experiments demonstrate the model's second-order convergence, low spurious mixing, and capability to simulate salt–freshwater interactions in the presence of wetting and drying boundaries.

1 Introduction

During the past few decades, numerical methods have evolved dramatically, and coastal ocean models can benefit from these advanced numerical methods (Blaise et al., 2010). Classical three-dimensional (3D) hydrodynamic models, e.g. ROMS (Shchepetkin and McWilliams, 2005), Delft3D (Lesser et al., 2004), FVCOM (Chen et al., 2003), TELEMAC (Moulinet et al., 2011), and SCHISM (Zhang et al., 2016), can be utilised to predict storm surges, tsunamis, and floods and assess environmental qualities by adding

additional equations (e.g. transport and reaction) to model oil slicks, contaminant plume propagation, temperature, and salinity transport, among other problems (Bertsch et al., 2022; Zhou et al., 2024; Sanz-Ramos et al., 2023).

In coastal and estuary waters, the presence of large gradients and strong discontinuity processes (e.g. wetting and drying processes and saline flow) may pose challenges for these classical circulation models, such as maintaining high numerical accuracy, minimising numerical mixing, and preventing excessive diffusion. From the perspective of numerical discretisation methods in the above models, the finite-volume (FV) method is known to have limitations in developing high-order accuracy schemes and incorporating so-called *hp* adaptivity (Lee, 2019). High-order FV models can increase the order by wide stencils, but they have to perform complex numerical reconstruction, which significantly increases computational complexity (Cheng et al., 2016). Models based on the finite-element (FE) method can also achieve high order, but they do not have local conservation. In the wet–dry (WD) fronts and in small-scale dynamics like baroclinic eddies, where local conservation of mass and momentum is critical, the property of local conservation is a preferred characteristic of a numerical scheme (Lee, 2020).

As a combination of both FV and FE methods, the discontinuous Galerkin (DG) method, with a relatively small number of elements, is well suited to modelling of 3D flows exhibiting strong velocity or density gradients. Dawson and

Aizinger (2005) first developed a 3D shallow-water equation (SWE) model based on DG methods called UTBEST3D. They used z coordinates in a vertical direction and a local discontinuous Galerkin (LDG) method for vertical diffusion (Aizinger and Dawson, 2007). To reduce the degrees of stiffness, Blaise et al. (2010) and Comblen et al. (2010) developed a 3D baroclinic marine model called SLIM3D and used the symmetric interior penalty Galerkin method (SIPG) to discretize the vertical diffusion. In addition, the mode-splitting procedure, which splits the 2D external mode and the 3D internal mode, is used to save computational time (Kowalik and Murty, 1993). Later, Delandmeter et al. (2018) added a fully consistent and conservative vertically adaptive coordinate system in SLIM3D. At the same time, Kärnä et al. (2018) also developed a 3D DG baroclinic model called Thetis. Thetis includes a more accurate mode-splitting method, revised viscosity formulation, and new second-order time integration scheme, which has lower or comparable numerical dissipation. The above three models all choose z coordinates, considering that σ coordinates enable a smooth representation of the bottom topography, which is particularly appropriate for coastal applications. Using σ coordinates, Conroy and Kubatko (2016) explicitly treated the vertical diffusion term to study the water column concerning barotropic forcing by building a 3D DG SWE model. The model of Conroy and Kubatko (2016) does not consider nonlinear advective terms and can only investigate 3D linear problems. The mentioned models use the quadrature-based nodal DG method. In explicit and semi-implicit time stepping schemes commonly used in connection with the SWE models, the most computationally expensive parts of a DG algorithm are the element and edge integrals computed via loops over quadrature points (Faghih-Naini et al., 2020). One criticism of the original DG method is that it requires expensive quadrature rules to reduce the aliasing error arising in the nonlinear system. The essence of the quadrature-free approach is to convert the nonlinear fluxes to polynomials or polynomial operations, which can be integrated analytically into the element or element boundaries (Li, 2024). The quadrature-free DG scheme has a comparable order and L_2 errors even at the 7th order of accuracy compared to the quadrature-based DG scheme for the isentropic vortex problem (Nair, 2015). In the MATLAB–C hybrid framework, Ran et al. (2022) developed a 3D barotropic model based on the quadrature-free nodal DG method. They choose the nonsplit-mode scheme considering that this method does not need to adjust the differences in the momentum calculation between the 2D external mode and the 3D internal mode and has fewer excessive errors (Chen et al., 2022; Conroy and Kubatko, 2016). A series of numerical experiments have proven the accuracy and potential application value of the barotropic model, but its computational efficiency is low, which limits the applicability of the model.

In addition to the utilisation of different numerical discretisation schemes, wetting and drying (WD) processes should

be included in coastal and estuarine modelling, which is a challenge for 3D DG models. At present, within existing 3D DG models, Vallaey (2018) noted that the WD processes can be considered in the external mode of the mode-splitting SLIM3D model, but this kind of wetting and drying module cannot be operated together with the baroclinic module. Chen et al. (2024) developed a WD treatment in a 3D DG model with the nonsplit-mode scheme, but there is still no description and verification of the coexistence of WD processes and thermohaline transport.

To adapt to the complex terrain of estuary and coastal areas and have a comparable calculation accuracy to existing 3D DG baroclinic models, this work aims to develop a new share-code discontinuous Galerkin coastal and estuarine modelling system (DGCEMS) for a 3D nonsplit-mode, implicit–explicit Runge–Kutta baroclinic model using σ coordinates with a WD module based on a quadrature-free nodal DG scheme. The arrangement of this paper is as follows. Section 2 describes the model with the discretisation of each term in the model, while Sect. 3 presents numerical tests and applications to show the main functionality of the model. Conclusions are given in Sect. 4.

2 Model and discretisation

2.1 Governing equations

The conservative and pre-balanced scheme of 3D shallow-water equations in the σ coordinates, with advection–diffusion equations of temperature T and salinity S , can be written as

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}). \quad (1)$$

Here $\mathbf{U} = [D, Du, Dv, DT, DS]^T$, where D is the water depth, and u and v are the velocity components along the x and y directions, respectively. The convective term $\mathbf{F}(\mathbf{U}) = [\mathbf{E}(\mathbf{U}), \mathbf{G}(\mathbf{U}), \mathbf{H}(\mathbf{U})]$, and each part in this term can be written as

$$\mathbf{E}(\mathbf{U}) = \begin{bmatrix} Du \\ Du^2 + 1/2(g(D^2 - z_b^2)) \\ Duv \\ DuT \\ DuS \end{bmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{bmatrix} Dv \\ Duv \\ Dv^2 + 1/2(g(D^2 - z_b^2)) \\ DvT \\ DvS \end{bmatrix}, \quad \mathbf{H}(\mathbf{U}) = \begin{bmatrix} \omega \\ \omega u \\ \omega v \\ \omega T \\ \omega S \end{bmatrix}, \quad (2)$$

where g is gravitational acceleration; z_b is the bottom elevation, and ω is the vertical velocity along the σ direction. The right side of Eq. (1) contains the terms of

$$\mathbf{S}(\mathbf{U}) = \mathbf{S}_b + \mathbf{S}_f + \mathbf{S}_{d,h} + \mathbf{S}_{d,v} + \mathbf{S}_{baro}, \quad (3)$$

where S_b , S_f , $S_{d,h}$, $S_{d,v}$, and S_{baro} represent the bottom topography term, Coriolis acceleration term, horizontal diffusion term, vertical diffusion term, and baroclinic term, respectively. The five terms are given as follows:

$$S_b = \begin{bmatrix} 0 \\ -g\eta\partial z_b/\partial x \\ -g\eta\partial z_b/\partial y \\ 0 \\ 0 \end{bmatrix}, S_f = \begin{bmatrix} 0 \\ -fDv \\ fDu \\ 0 \\ 0 \end{bmatrix}, \quad (4)$$

$$S_{d,h} = \begin{bmatrix} 0 \\ \frac{\partial}{\partial x}(K_h D \frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(K_h D \frac{\partial u}{\partial y}) \\ \frac{\partial}{\partial x}(K_h D \frac{\partial v}{\partial x}) + \frac{\partial}{\partial y}(K_h D \frac{\partial v}{\partial y}) \\ \frac{\partial}{\partial x}(K_H D \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(K_H D \frac{\partial T}{\partial y}) \\ \frac{\partial}{\partial x}(K_H D \frac{\partial S}{\partial x}) + \frac{\partial}{\partial y}(K_H D \frac{\partial S}{\partial y}) \end{bmatrix},$$

$$S_{d,v} = \begin{bmatrix} 0 \\ \frac{\partial}{\partial \sigma}(K_v \frac{\partial Du}{D^2 \partial \sigma}) \\ \frac{\partial}{\partial \sigma}(K_v \frac{\partial Dv}{D^2 \partial \sigma}) \\ \frac{\partial}{\partial \sigma}(K_v \frac{\partial DT}{D^2 \partial \sigma}) \\ \frac{\partial}{\partial \sigma}(K_v \frac{\partial DS}{D^2 \partial \sigma}) \end{bmatrix}, \quad (5)$$

$$S_{baro} = \begin{bmatrix} 0 \\ -\frac{gD}{\rho_0} \left[\int_{\sigma}^0 D \frac{\partial \rho}{\partial x} d\sigma - \frac{\partial D}{\partial x} \int_{\sigma}^0 \sigma \frac{\partial \rho}{\partial \sigma} d\sigma \right] \\ -\frac{gD}{\rho_0} \left[\int_{\sigma}^0 D \frac{\partial \rho}{\partial y} d\sigma - \frac{\partial D}{\partial y} \int_{\sigma}^0 \sigma \frac{\partial \rho}{\partial \sigma} d\sigma \right] \\ 0 \\ 0 \end{bmatrix}, \quad (6)$$

where $\eta = D + z_b$ is the surface elevation; f is the Coriolis coefficient; K_h and K_H are the horizontal (eddy) diffusion coefficients using Smagorinsky's method (Smagorinsky, 1963); K_v and K_V are the vertical (eddy) diffusion coefficients calculated by the MY-2.5 turbulent closure scheme in the General Ocean Turbulence Model (GOTM; Burchard et al., 1999); ρ_0 is the reference density; and the water density ρ is defined as $\rho = \rho(T, S)$, which is a function of T and S . In the model, two state equations can be chosen: linear, shown in Eq. (7), and full nonlinear, from Jackett et al. (2006):

$$\rho(T, S) = \rho_0 - \alpha_T(T - T_0) + \beta_S(S - S_0). \quad (7)$$

In Eq. (7), α_T and β_S are coefficients. T_0 and S_0 are reference temperature and salinity, respectively. It can be seen that the above system has seven prognostic variables, namely D (or η), T , u , v , ω , ρ , T , and S , but there are a total of six equations (Eqs. 1 and 7) to be solved. To this end, the depth-averaged continuity equation is applied to calculate the water depth (or free surface) for the nonsplit-mode model, which can be written as

$$\frac{\partial D}{\partial t} + \frac{\partial DU}{\partial x} + \frac{\partial DV}{\partial y} = 0, \quad (8)$$

where $U = \int_{-1}^0 u d\sigma$ and $V = \int_{-1}^0 v d\sigma$ are depth-averaged horizontal velocities. The vertical velocity is solved by combining 2D and 3D continuity equations, which can be written as

$$\frac{\partial \omega}{\partial \sigma} = \frac{\partial(DU - Du)}{\partial x} + \frac{\partial(DV - Dv)}{\partial y} \quad (9)$$

and then be integrated layer by layer along the vertical direction.

2.2 Discontinuous Galerkin discretisation

In this section, we describe the numerical implementation of DGCEMS. The DG function spaces are defined in Sect. 2.2.1, followed by the quadrature-free nodal DG discretisation (Hesthaven and Warburton, 2007) of the above governing equations.

2.2.1 Function spaces and notations

For a 3D computational domain, Ω_{3d} , there is a corresponding projection at the horizontal plane, which is denoted as Ω_{2d} , and it can be divided by an unstructured triangular mesh. These triangular elements extend vertically and form triangular prisms, which are computational elements in Ω_{3d} . The DG discretisation of this model is based on the linear function space, which is denoted as P_{N_h} in the 2D space and $P_{(N_h, N_v)}$ in the 3D space (in Fig. 1, both N_h and N_v are equal to 1, representing the stage of the function of horizontal and vertical space). It should be noted that the number of vertical layers in the model is flexible and user-defined. The vertical discretisation shown in Fig. 1 employs two layers for illustrative purposes only, to demonstrate the distribution of interpolation nodes in the σ -coordinate direction. In practical simulations, more vertical layers (e.g. 10 to 20 layers) are typically used to improve the resolution of vertical velocity profiles and stratification. Increasing the number of layers enhances vertical accuracy but does not change the order of numerical convergence and may increase computational cost. We set $\phi_{2d} \in P_{N_h}$, $\varphi_{2d} \in P_{N_h}^2$, $\phi_{3d} \in P_{(N_h, N_v)}$, and $\varphi_{3d} \in P_{(N_h, N_v)}^2$ as the test functions in the 2D and 3D function spaces.

Next, we use the following notations to represent the volume and surface integrals:

$$\langle \cdot \rangle_{\Omega} = \int_{\Omega} (\cdot) d\mathbf{x}, \quad (10)$$

$$\langle \langle \cdot \rangle \rangle_{\partial\Omega} = \int_{\partial\Omega} (\cdot) ds. \quad (11)$$

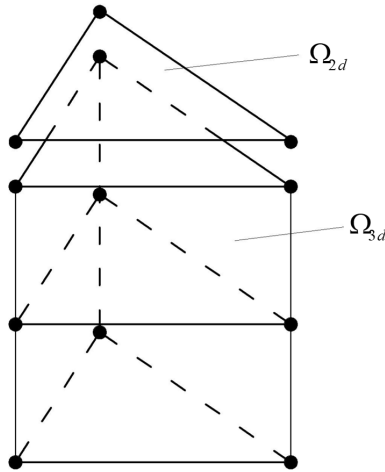


Figure 1. Schematic of vertical computational element distribution (example with two layers). Black dots represent the interpolation nodes corresponding to the horizontal first-order and vertical first-order basis functions.

In surface terms, we additionally define the concept of the average, $\{\{ \cdot \} \}$, and the jump, $[[\cdot]]$, to simplify the expression of subsequent formulas:

$$\{\{a\}\} = \frac{1}{2}(a^+ + a^-), \quad (12)$$

$$\{\{b\}\} = \frac{1}{2}(b^+ + b^-), \quad (13)$$

$$[[ab]] = a^+b^+ + a^-b^-, \quad (14)$$

$$[[b \cdot n]] = b^+ \cdot n^+ + b^- \cdot n^-, \quad (15)$$

$$[[bn]] = b^+n^+ + b^-n^-, \quad (16)$$

where a represents a scalar field and b represents a vector field, $-$ and $+$ are the local side and adjacent side, and $n = (n_x, n_y, n_\sigma)$ is the outward unit normal on the edges. Let $\mathcal{T}$ and $\mathcal{P}$ stand for the partition of the 2D domain Ω_{2d} and 3D domain Ω_{3d} , respectively. Element interfaces are denoted by $\mathcal{I}_{2d} = \{k_i \cap k_j | k_i, k_j \in \mathcal{T}, i \neq j\}$, while $\mathcal{I}_h$ and $\mathcal{I}_v$ are notations of horizontal and vertical interfaces in $\mathcal{P}$.

2.2.2 Convection terms and source terms

We multiply Eq. (1) by a test function $\varphi \in \phi_{3d}$ and integrate the face terms by parts twice:

$$\begin{aligned} & \langle U_t \cdot \varphi \rangle_{\Omega_{3d}} - \langle F(U) \cdot \nabla \varphi \rangle_{\Omega_{3d}} \\ & + \left\langle \left((E(U), G(U))^{*,HLLC} \cdot [[\varphi n_h]] \right) \right\rangle_{\mathcal{I}_v} \\ & - \langle \langle (E(U), G(U))^- \cdot [[\varphi n_h]] \rangle \rangle_{\mathcal{I}_v} \\ & + \langle \langle H(U)^{*,up} \cdot [[\varphi n_\sigma]] \rangle \rangle_{\mathcal{I}_h} \\ & - \langle \langle H(U)^- \cdot [[\varphi n_\sigma]] \rangle \rangle_{\mathcal{I}_h} \\ & = \langle S_b \cdot \varphi \rangle_{\Omega_{3d}} + \langle S_f \cdot \varphi \rangle_{\Omega_{3d}} + \langle S_{baro} \cdot \varphi \rangle_{\Omega_{3d}} \\ & - D_h(u, v, T, S, \varphi) + D_v(u, v, T, S, \varphi). \end{aligned} \quad (17)$$

Here, $(*)^{*,HLLC}$ is calculated by the Harten–Lax–van Leer–Contact (HLLC) method (Toro, 2001); $(*)^{*,up}$ means the upwind numerical flux at the interface; $n_h = (n_x, n_y)$ is the horizontal outward unit normal vector; n_σ represents the vertical outward unit normal vector; and D_h and D_v are the horizontal and vertical diffusion operators, which are introduced in the next section. In computational elements, the bottom and top interfaces are horizontal ($n_x, n_y = 0$), while the side interfaces are vertical ($n_\sigma = 0$). For convection terms, the process of surface integration is divided into a horizontal part and a vertical part, which need to use the upwind and the HLLC approximate Riemann solver, respectively. This model uses the quadrature-free nodal DG method to avoid the complex calculation processes of the Gaussian integral for nonlinear convection terms. The DG discretisation of the source terms also contains the process of the surface integral, but it seems to be omitted in Eq. (17) because the central numerical flux is used, resulting in the surface integral being 0 as a whole.

2.2.3 Horizontal and vertical diffusion terms

The discretisation of diffusion terms differs from that of convection because they have elliptic operators, which need additional stabilisation. Thus, the symmetric interior penalty Galerkin (SIPG) method (Epshteyn and Rivière, 2007) is used, where D_h and D_v read

$$\begin{aligned} D_h(u, \varphi) = & - \langle \nabla_h \varphi \cdot (D v_h \cdot \nabla_h u)^T \rangle_{\Omega_{3d}} \\ & + \langle \langle \{ \{ D v_h \cdot \nabla_h u \} \} \cdot [[\varphi n_h]] \rangle \rangle_{\mathcal{I}_v} \\ & + \langle \langle \{ \{ D v_h \cdot \nabla_h u \} \} \cdot [[un_h]] \rangle \rangle_{\mathcal{I}_v} \\ & - \langle \langle \{ \tau \} \{ \{ D v_h \} \} [[un_h]] [[\varphi n_h]] \rangle \rangle_{\mathcal{I}_v}, \end{aligned} \quad (18)$$

$$\begin{aligned} D_v(u, \varphi) = & - \left\langle \frac{\partial \varphi}{\partial \sigma} \cdot \left(\frac{v_v}{D^2} \cdot \frac{\partial Du}{\partial \sigma} \right) \right\rangle_{\Omega_{3d}} \\ & + \left\langle \left\langle \left\{ \left\{ \frac{v_v}{D^2} \cdot \frac{\partial Du}{\partial \sigma} \right\} \right\} \cdot [[\varphi n_\sigma]] \right\rangle \right\rangle_{\mathcal{I}_h} \\ & + \left\langle \left\langle \left\{ \left\{ \frac{v_v}{D^2} \cdot \frac{\partial \varphi}{\partial \sigma} \right\} \right\} \cdot [[Dun_\sigma]] \right\rangle \right\rangle_{\mathcal{I}_h} \\ & - \left\langle \left\langle \{ \tau \} \left\{ \left\{ \frac{v_v}{D^2} \right\} \right\} [[un_\sigma]] [[\varphi n_\sigma]] \right\rangle \right\rangle_{\mathcal{I}_h}. \end{aligned} \quad (19)$$

Here, $v_h = (0, K_h, K_h, K_h, K_h)^T$ and $v_v = (0, K_v, K_v, K_v, K_v)^T$, and τ is called the penalty factor. Following the study by Shahbazi (2005), it can be defined as

$$\tau = \frac{\gamma}{2} \frac{(N+3)(N+1)}{3L}, \quad (20)$$

in which N represents the order value of the horizontal or vertical basic function, γ is the number of surfaces in an element, and L is the length scale of the local element in the normal direction of the surface.

2.2.4 Primitive continuity and vertical velocity equations

We also multiply Eq. (8) by the test function and integrate the face terms by parts twice. The strong-form formulation of the primitive continuity equation is shown in Eq. (20):

$$\begin{aligned} \left\langle \frac{\partial D}{\partial t} \varphi \right\rangle_{\Omega_{2d}} &- \langle DU \cdot \nabla_h \varphi \rangle_{\Omega_{2d}} - \langle DV \cdot \nabla_h \varphi \rangle_{\Omega_{2d}} \\ &+ \left\langle (DU)^{*,\text{HLLC}} \cdot [[\varphi \mathbf{n}_h]] \right\rangle_{\mathcal{I}_{2d}} \\ &- \langle (DU)^- \cdot [[\varphi \mathbf{n}_h]] \rangle_{\mathcal{I}_{2d}} \\ &+ \left\langle (DV)^{*,\text{HLLC}} \cdot [[\varphi \mathbf{n}_h]] \right\rangle_{\mathcal{I}_{2d}} \\ &- \langle (DV)^- \cdot [[\varphi \mathbf{n}_h]] \rangle_{\mathcal{I}_{2d}} = 0. \end{aligned} \quad (21)$$

Then, expanding the function space of Eq. (21) to the 3D system and subtracting that from the first line of Eq. (17), we have

$$\begin{aligned} \left\langle \frac{\partial \omega}{\partial \sigma} \varphi \right\rangle_{\Omega_{3d}} &= \langle (Du - DU) \cdot \nabla_h \varphi \rangle_{\Omega_{3d}} \\ &+ \langle (Dv - DV) \cdot \nabla_h \varphi \rangle_{\Omega_{3d}} \\ &- \left\langle (Du - DU)^{*,\text{HLLC}} \cdot [[\varphi \mathbf{n}_h]] \right\rangle_{\mathcal{I}_v} \\ &+ \langle (Du - DU)^- \cdot [[\varphi \mathbf{n}_h]] \rangle_{\mathcal{I}_v} \\ &- \left\langle (Dv - DV)^{*,\text{HLLC}} \cdot [[\varphi \mathbf{n}_h]] \right\rangle_{\mathcal{I}_v} \\ &+ \langle (Dv - DV)^- \cdot [[\varphi \mathbf{n}_h]] \rangle_{\mathcal{I}_v}. \end{aligned} \quad (22)$$

Finally, the vertical integral can achieve vertical velocity from the bottom to the top.

2.3 Slope limiters and wetting/drying treatment

The slope limiters are necessary for the stabilisation of the model. Here we use a 3D anisotropic limiter to control the large gradient of the horizontal momentum and the tracers. The limited solution at the j th nodal number in the k th triangular prism element is denoted as $\tilde{T}_{k,j}$, which is defined as

$$\tilde{T}_{k,j} = \lambda T_{k,j} + (1 - \lambda) \bar{T}_k, \quad (23)$$

where $T_{k,j} = [(Du)_{k,j}, (Dv)_{k,j}, (DT)_{k,j}, (DS)_{k,j}]^T$ represents the original solution at the k th triangular prism element in $\mathcal{P}$; $\bar{T}_k$ is the volume-averaged solution at the k th triangular prism element; and λ is defined by Delandmeter (2017), which ensures the anisotropy of the 3D limiter.

Considering the WD processes in the nearshore regions, the strong discontinuity at the WD fronts also frequently causes difficulties in the model stability (Le et al., 2020; Medeiros and Hagen, 2013). To avoid the WD problems that may occur in the computational processes, we follow the WD treatment method by Chen et al. (2024), where they combine

the vertex-based limiter by Li and Zhang (2017), the well-balanced positivity-preserving limiter proposed by Xing and Zhang (2013), and Eq. (23) to prevent the blow-up of the water level and momentum solutions. In this model, the stencil of the 2D vertex-based limiter has been adjusted to align with the 3D limiter. This modification was identified through extensive numerical testing and has been found to enhance computational stability to a certain extent.

2.4 Time stepping

In the DG framework, the explicit Runge–Kutta time stepping method is popular. By selecting different numbers of stages, different levels of temporal accuracy can be achieved. Although the explicit scheme has a relatively low computational cost per time step, it imposes strict limitations on time steps. Specifically, for high-stiffness terms like the vertical diffusion term, the required time step should be sufficiently small. Thus, we use the implicit–explicit Runge–Kutta (IMEXRK) scheme. For clarity, the above DG discretisation processes are summarised as

$$\frac{dy}{dt} = f^{\text{EX}}(y) + f^{\text{IM}}(y), \quad (24)$$

where y denotes the degree of freedom over all elements; $f^{\text{IM}}(y)$ is the implicit system for the vertical diffusion term, and $f^{\text{EX}}(y)$ represents the explicit system that contains all other terms. The IMEXRK system reads

$$y^{(1)} = y^{(n)} + \Delta t f^{\text{EX}}(y^{(n)}), \quad (25)$$

$$y^{(2)} = y^{(n)} + \frac{1}{2} \Delta t (f^{\text{EX}}(y^{(n)}) + f^{\text{EX}}(y^{(1)})), \quad (26)$$

$$y^{(n+1)} = y^{(2)} + \Delta t f^{\text{IM}}(y^{(n+1)}). \quad (27)$$

Here, when the model calculates from steps n to $n + 1$, it has two middle steps, shown in Eqs. (25) and (26). Finally, the whole algorithm from time steps t_n to t_{n+1} is shown in Algorithm 1. In the workflow, after the physical variables are computed at each time step, 3D slope limiters are applied to the reconstructed solution to suppress spurious oscillations and prevent the generation of non-physical values, particularly near steep gradients or discontinuities. These limiters ensure numerical stability. Subsequently, the physical fields are vertically averaged to derive depth-integrated variables. The WD treatment is then performed on these depth-averaged variables to accurately capture shoreline movement while preserving the conservation of water surface elevation and depth-averaged momentum. This sequential process improves robustness and physical consistency in simulations involving complex wetting and drying processes.

Algorithm 1: the calculation process of the model in a computational time step

First stage

1. Calculate the convection term, the horizontal diffusion term, the source terms, and the primitive continuity equation (Eqs. 17, 18, and 21) explicitly according to $D^{(n)}$ (or $\eta^{(n)}$), $Du^{(n)}$, $Dv^{(n)}$, $DT^{(n)}$, and $DS^{(n)}$ at time t_n .
2. Achieve the values of $D^{(1)}$ (or $\eta^{(1)}$), $Du^{(1)}$, $Dv^{(1)}$, $DT^{(1)}$, and $DS^{(1)}$ at the first middle step (Eq. 25).
3. Apply slope limiters to $Du^{(1)}$, $Dv^{(1)}$, $DT^{(1)}$, and $DS^{(1)}$ (Eq. 23).
4. Calculate vertically averaged horizontal velocities by integration and update the WD statement, the density $\rho^{(n)}$, and the vertical velocity $\omega^{(n)}$ (Eqs. 7 and 9).

Second stage

1. Calculate the convection term, the horizontal diffusion term, the source terms, and the primitive continuity equation (Eqs. 17, 18, and 21) explicitly according to $D^{(1)}$ (or $\eta^{(1)}$), $Du^{(1)}$, $Dv^{(1)}$, $DT^{(1)}$, and $DS^{(1)}$.
2. Achieve the values of $D^{(n+1)}$ (or $\eta^{(n+1)}$), $Du^{(2)}$, $Dv^{(2)}$, $DT^{(2)}$, and $DS^{(2)}$ at the second middle step using the values at time t_n and the first middle step (Eq. 26).
3. Apply slope limiters to $Du^{(2)}$, $Dv^{(2)}$, $DT^{(2)}$, and $DS^{(2)}$ (Eq. 23).

Final stage

1. Calculate the vertical viscosity and vertical diffusion implicitly.
2. Achieve the values of $Du^{(n+1)}$, $Dv^{(n+1)}$, $DT^{(n+1)}$, and $DS^{(n+1)}$ at time t_{n+1} (Eq. 27).
3. Calculate vertically averaged horizontal velocities by integration and update the WD statement, the density $\rho^{(n+1)}$, and the vertical velocity $\omega^{(n+1)}$ (Eqs. 7 and 9).

3 Tests and analysis

In this section, we first validate the baroclinic solver by a manufactured solution test and then verify the diffusion terms using a standard lock exchange test. An ideal river plume test case and a semi-closed estuary case of salinity intrusion are also applied to examine the performance of DGCEMS.

3.1 Baroclinic manufactured solution test

The baroclinic manufactured solution test was inspired by Kärnä et al. (2018), and their original z -coordinate formulation was converted into a σ -coordinate framework for validation. The computational domain is a rectangular box of length $L_x = 15$ km, width $L_y = 10$ km, and depth $D = 40$ m. At the initial time step, we define the velocity field and the tracer field as follows:

$$u = \frac{1}{2} \sin\left(\frac{2\pi x}{L_x}\right) \cos(3\sigma), \quad (28)$$

$$v = \frac{1}{3} \cos\left(\frac{\pi y}{L_y}\right) \sin\left(\frac{\sigma}{2}\right), \quad (29)$$

$$T = 15 + 15 \sin\left(\frac{\pi x}{L_x}\right) \sin\left(\frac{\pi y}{L_y}\right) \cos(\sigma), \quad (30)$$

where the above three expressions are also represented using the σ coordinate, ranging from -1 to 0 , in the vertical direction. In this case, the Coriolis parameter $f = 0.0001$, while the bottom friction term, the viscosity term, and the diffusion term are omitted. The linear state equation is applied with $\rho_0 = 1000 \text{ kg m}^{-3}$, $\alpha_T = 0.2 \text{ kg m}^{-3} \text{ }^\circ\text{C}^{-1}$, and $T_0 = 5.0 \text{ }^\circ\text{C}$. The value of β_s is not defined because the salinity in the whole field is a constant. We set 10 layers at vertical directions with horizontal mesh resolutions of 2500, 1250, 625, 312.5, and 156.25 m, where the grid shapes are isosceles right-angled triangles. In the computational domain, all the boundary edges are closed, and no external force needs to be applied, which causes the system to have a time-dependent problem. To achieve a steady-state solution, we derived the analytical functions of the advection term, the baroclinic term, and the Coriolis acceleration term and then added them to the source term to balance the initial velocity and tracer field. Details can be found in Appendix A.

For the five grid resolutions mentioned above, the model was run for 100 steps with time steps of 4.0, 2.0, 1.0, 0.5, and 0.25 s. The L_2 errors in the water elevation, horizontal velocity, vertical velocity, and temperature field across the entire computational domain were calculated, as shown in Fig. 2. The red lines shown in the figure represent the best-fit lines obtained using the least-squares method, with their slopes indicating the order of convergence. For varying grid resolutions, the water elevation, horizontal velocity, and temperature exhibit near-second-order convergence. However, the vertical velocity fails to achieve second-order convergence, which is attributed to the use of the vertically averaged continuity equation in the vertical velocity computation. The numerical solution of this equation inherently contains errors that accumulate throughout the calculation process, preventing the vertical velocity from reaching optimal second-order convergence. Overall, the results from this artificial analytical solution test are consistent with those of Kärnä et al. (2018), confirming that the discretisation of the advection

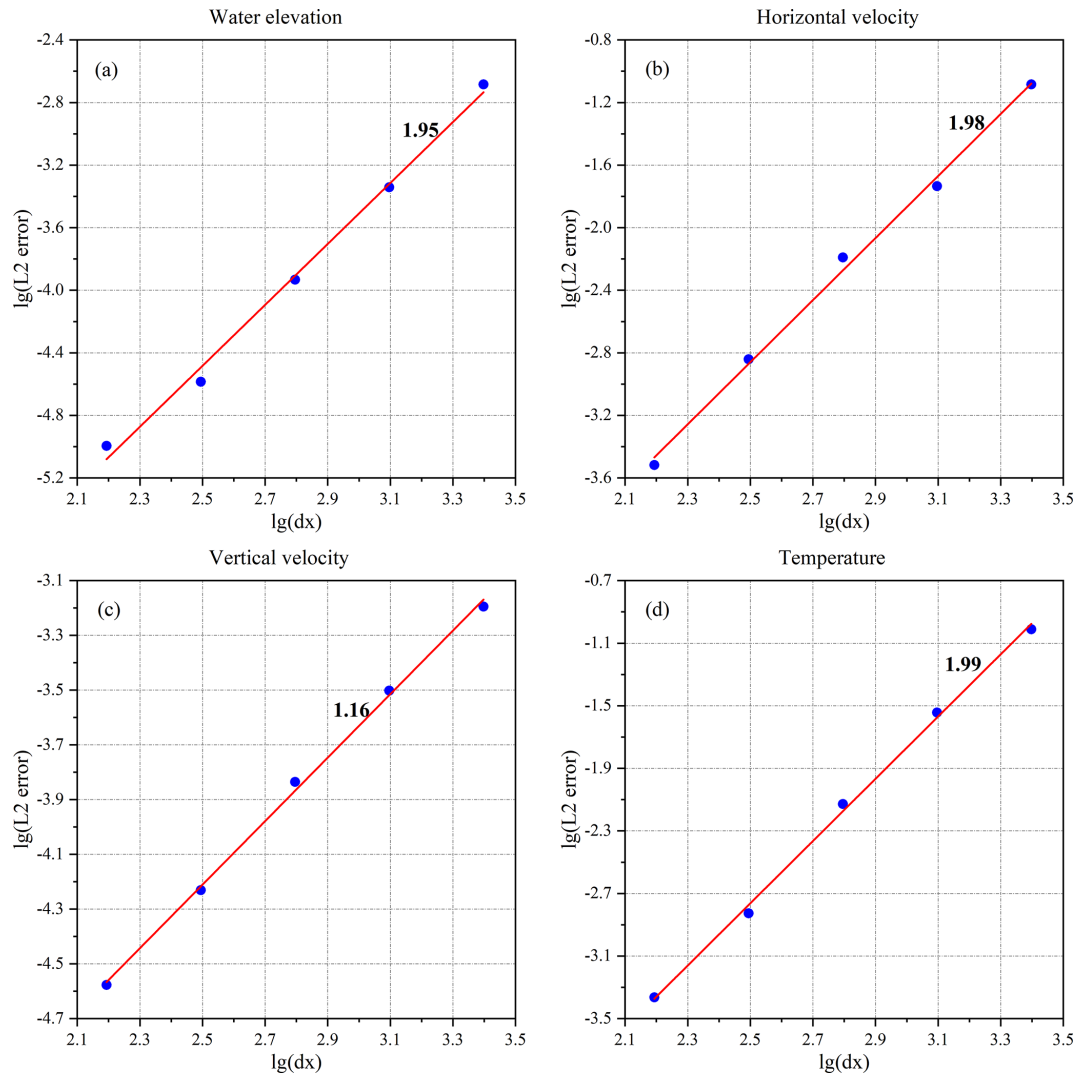


Figure 2. Convergence of the L_2 errors in the (a) surface elevation, (b) horizontal velocity, (c) vertical velocity, and (d) temperature field in the baroclinic manufactured solution test case.

term, baroclinic term, and Coriolis term in σ coordinates is accurate at both the coding and the algorithmic levels.

3.2 Lock exchange

In this subsection, we validate the baroclinic term and discuss the spurious mixing caused by horizontal diffusion with a standard lock exchange test (Ilıcak et al., 2012; Petersen et al., 2015; Kärrnä et al., 2018). The computational domain is a rectangular channel that is 64 km long, 1 km wide, and 20 m deep. Each element in the domain is $\Delta x = 500$ m of the triangle edge length and 1 m in the vertical direction (total of 20 layers). The initial salinity is set to a constant value of 35 psu in the whole domain, and the initial temperature is calculated by

$$T(x, y, \sigma) = 5^\circ\text{C} \text{ where } 0 \leq x < 32 \text{ km}, \quad (31)$$

$$T(x, y, \sigma) = 35^\circ\text{C} \text{ where } 32 \leq x \leq 64 \text{ km}. \quad (32)$$

We use Eq. (7) to update the density, where $\alpha_T = 0.2$, $\beta_S = 0$, $T_0 = 5^\circ\text{C}$, and $S_0 = 35$ psu. At the location of $x = 32$ km, the initial density difference is $\rho = 6.0 \text{ kg m}^{-3}$. In the model, only the convective term controls the tracer concentration, and the bottom friction is omitted. The vertical viscosity, K_v , is $10^{-4} \text{ m}^2 \text{ s}^{-1}$, and the horizontal viscosity, K_h , is varied with a value of 1, 10, and $200 \text{ m}^2 \text{ s}^{-1}$ so that the grid Reynolds numbers $Re = U_{\text{grid}} \Delta x / K_h = 250.0, 25.0$, and 1.25, where $U_{\text{grid}} = 0.5 \text{ m s}^{-1}$ represents the characteristic velocity scale.

Figure 3a shows the initial density field, and Fig. 3b–d are the density fields at the time of 17 h, where the Re values are 250.0, 25.0, and 1.25, respectively. It seems that as the Re decreases, the density front becomes smoother, indicating that

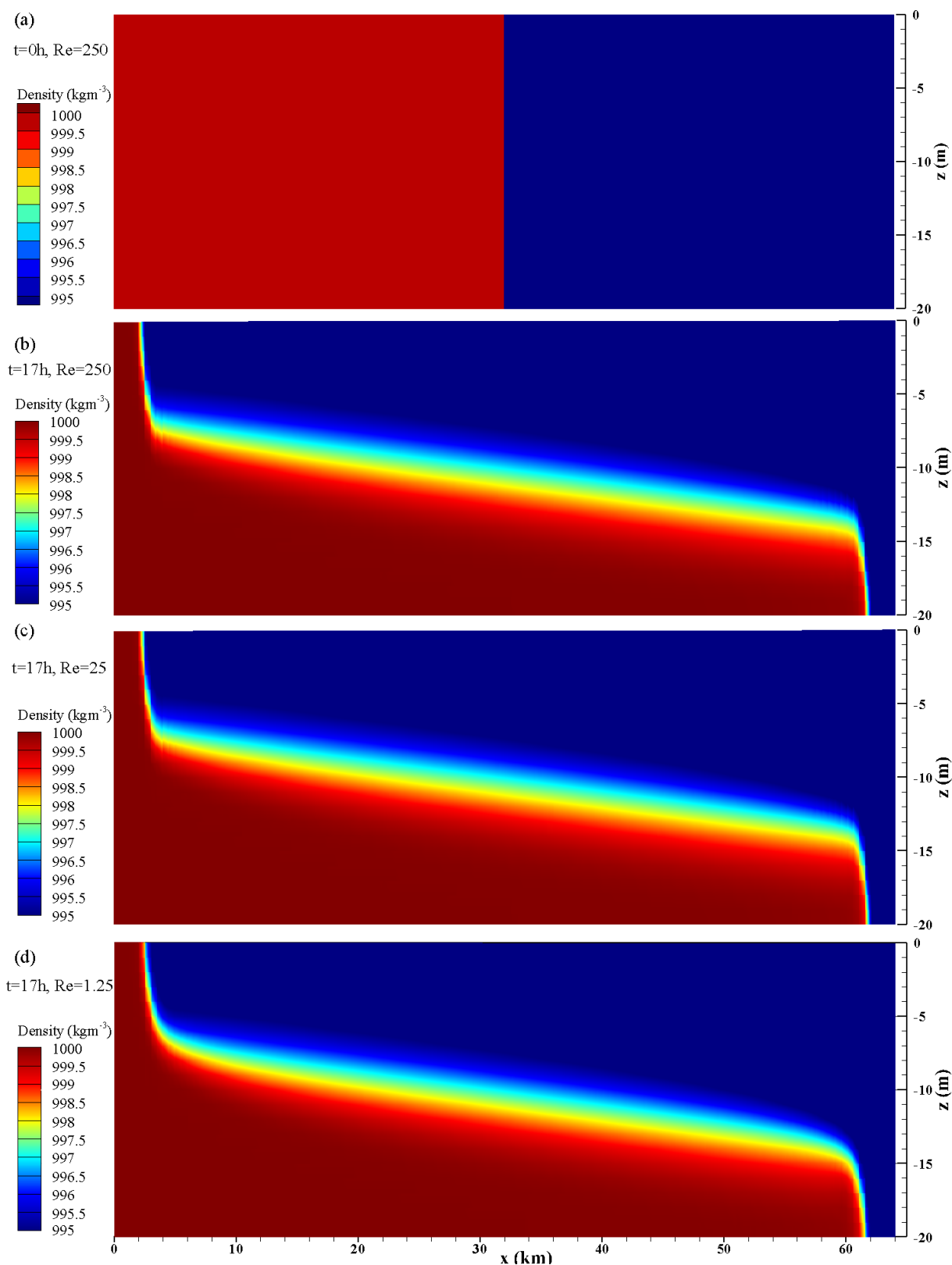


Figure 3. The density in the lock exchange test under three different grid Reynolds numbers.

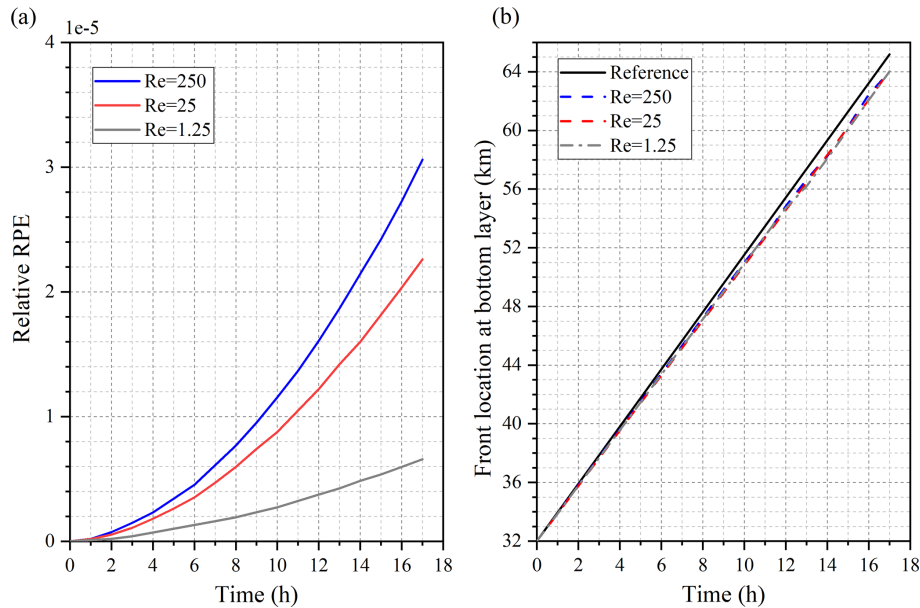


Figure 4. Simulation results of (a) relative RPE and (b) the density front location at the bottom layer of the lock exchange test with different Re .

the increase in background viscosity reduces overall mixing within the system. To quantify the role of spurious mixing, we also use a reference potential energy (RPE), which can be defined as

$$\text{RPE} = g \int \rho^*(z + D) dz, \quad (33)$$

where ρ^* is rearranged in descending order based on actual density values ρ from the bottom to the top. Using this computational approach, RPE represents the potential energy within the system that cannot be converted into kinetic energy. When spurious mixing is present, the value of RPE in the system increases. At any given time, t , the dimensionless relative reference potential energy ($\overline{\text{RPE}}(t)$) is defined as follows:

$$\overline{\text{RPE}}(t) = \frac{\text{RPE}(t) - \text{RPE}(0)}{\text{RPE}(0)}. \quad (34)$$

The calculation results of $\overline{\text{RPE}}(t)$ for the three Reynolds numbers are shown in Fig. 4a. At the 17th hour, the values of $\overline{\text{RPE}}(17)$ are 3.06, 2.26, and 0.66×10^{-5} , which agree with previous simulation results (Ilıcak et al., 2012; Kärnä et al., 2018). Additionally, the locations of the density fronts simulated under three different grid Reynolds number conditions were captured and compared with the theoretical maximum propagation distance, L , where $L = t/2\sqrt{gD\Delta\rho/\rho_0}$. The spurious mixing resulted in a loss of potential energy throughout the system, leading to discrepancies between the model simulations and the theoretical values (Fig. 4b). Overall, the error remains within 4.7 %, which is consistent with Kärnä et al. (2018).

3.3 An ideal river plume

We continue to experiment with an ideal river plume (Wu 2023) to analyse the model's performance. Saltwater can be replaced by various contaminants that affect water quality. Therefore, investigating surface plume dynamics not only validates the model's accuracy but also highlights its applicability to environmental modelling and pollutant transport simulations.

3.3.1 Settings

Figure 5 shows the topography of the estuary region, where there is a 7.5 km long, 3.0 km wide, and 15 m deep river that connects to the sea. The initial computational mesh domain is shown in Fig. 6. Freshwater runoff with a rate of $1500 \text{ m}^3 \text{ s}^{-1}$ and a salinity of 0 flows into a shelf with a bottom slope of 0.0007, where the salinity of the ocean water is 32 psu. We first use a grid resolution of 1500 m to partition the computational domain, which means that there are only two rows of grids along the width of the river channel. In this test, the Coriolis parameter is calculated by the latitude of 45° N , which is about 0.0001; the bottom friction coefficient is calculated by the MY-2.5 turbulent closure scheme with a bottom roughness parameter of 0.2 mm; and the eddy diffusivity coefficient is set to 0. Other external forces are not considered in this test. The open boundary is set to radiative according to Wu (2023) so that the boundary will have less influence on the currents in the computational domain. We also use Eq. (7) to update the density, where $\alpha_T = 0$, $\beta_S = 0.78$, $T_0 = 20^\circ \text{ C}$, and $S_0 = 32 \text{ psu}$.

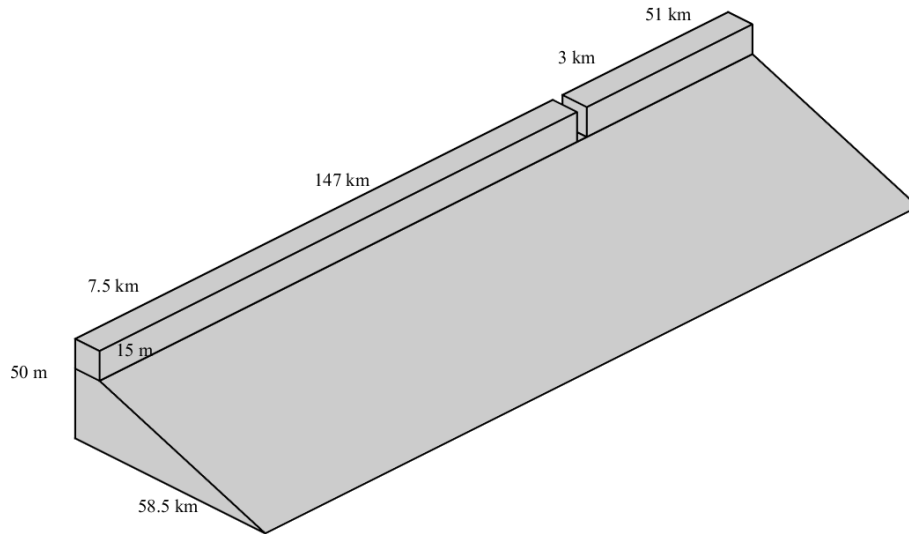


Figure 5. The topography of the ideal river plume experiment.

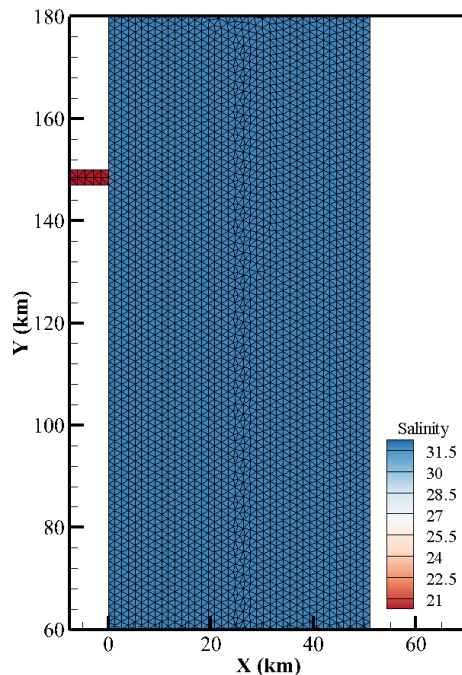


Figure 6. The computational mesh domain and the initial salinity field. The salinity is 0 psu at $x < 0$ km and 32 psu at $x \geq 0$ km.

3.3.2 Analysis of plume pattern

Due to the lack of analytical solutions in the test, we illustrate the accuracy of the simulation results from the perspective of grid convergence. For numerical methods, we prefer them to be less affected by grid resolution. Here, the mesh resolution is refined by a factor of 2 and 4 in both the x and the y directions, and the simulated results at 48 h are shown in Fig. 7. Considering the impact of differences in grid reso-

lution on images, the simulation results of the three types of grids were processed in the same way to minimise the impact of post-processing as much as possible. Based on the simulated plume pattern, as the computational grid resolution increases, the zigzag phenomenon at the plume front gradually diminishes, resulting in a smoother edge and a more pronounced bulge shape. Simultaneously, there are noticeable differences in the extent of freshwater transport along the coast; with higher grid resolution, the transport distance increases. This may be attributed to the finer grid's ability to more accurately capture buoyancy fronts, thereby enhancing the coastal extension of freshwater. Additionally, the finer resolution reduces numerical dissipation, contributing to the observed differences in transport distance.

3.3.3 Quantitative analysis

To further demonstrate the performance of the model, a series of quantitative methods are employed to analyse the numerical simulation results of freshwater content (Fc in short), plume characteristics in the near- and mid-field regions, and coastal plume transport in the far-field region under different computational grid resolutions.

Following the concept of the isohaline coordinate by MacCready et al. (2002), the calculation method of freshwater volume can be written as

$$V(S^*) = \int_{S < S^*} (S_0 - S)/S_0 dV, \quad (35)$$

where S^* represents each salinity level. Then, we have

$$Fc(S) = \frac{dV(S)}{dS}. \quad (36)$$

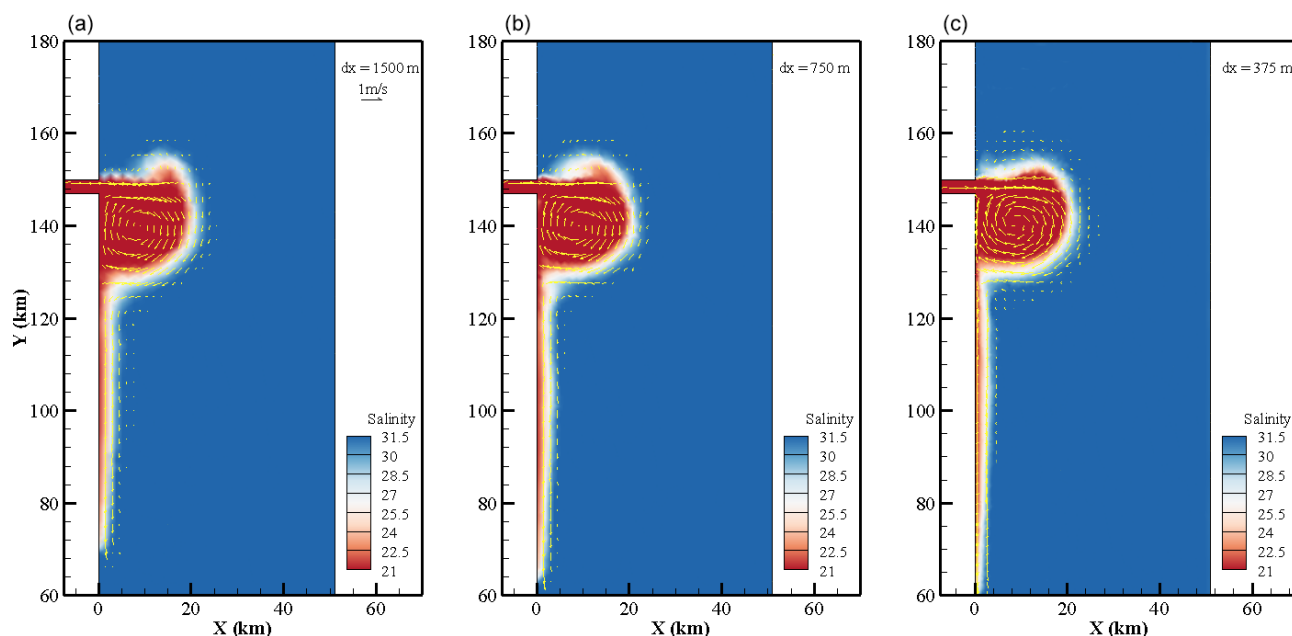


Figure 7. Simulated surface river plume and surface current velocity at 48 h. The grid resolution is refined from 1500 m (a) to 375 m (c).

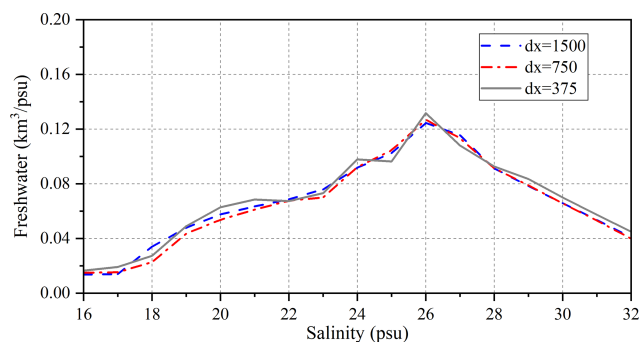


Figure 8. Freshwater content per salinity class at 48 h for results with different grid resolutions.

It can be seen in Fig. 8 that under the three grid resolutions, the freshwater content exhibits a similar trend: it increases initially and then decreases as salinity rises, with the peak freshwater content occurring at a salinity of 26 psu. This behaviour can be explained by the relationship between salinity and the distance from the river mouth. Lower salinity values indicate proximity to the river mouth, where the number of grids is relatively small, resulting in lower freshwater content in these regions. Conversely, higher salinity values correspond to regions farther from the river mouth, where more grids are present, but the freshwater content per cell is low, leading to lower overall freshwater content in high-salinity areas. These results are consistent with the research of Wu (2023).

Secondly, at $y = 140$ km, approximately at the location of the plume's rotational centre, a cross-sectional profile was

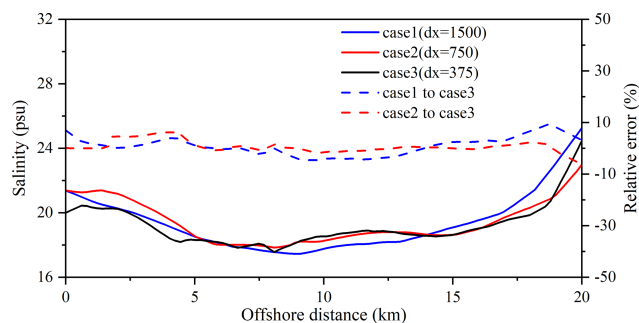


Figure 9. Surface salinity at the profile of $y = 140$ km under three grid resolutions. Dashed blue and red lines are the relative salinity errors for grid sizes of 1500 and 750 m, respectively, compared to the 375 m grid.

extracted to analyse the effect of grid resolution on the formation of the bulge in the near- and mid-field regions (see Fig. 9). When the grid resolution is 1500 m, the salinity at the centre of the bulge is relatively low, around 17.5 psu. As the grid resolution increases, the asymmetry of surface salinity from the centre of the bulge toward the edge of the inertial radius becomes more pronounced. Using the 375 m grid resolution as the reference, when the grid resolution is doubled or quadrupled, the salinity error relative to the reference case fluctuates within 10 %.

Finally, at $y = 100$ km, the relative freshwater transport carried by the coastal buoyancy current induced by the plume is calculated to assess the impact of grid resolution on the simulation results in the far-field region. The relative freshwater transport is determined by the ratio of the freshwa-

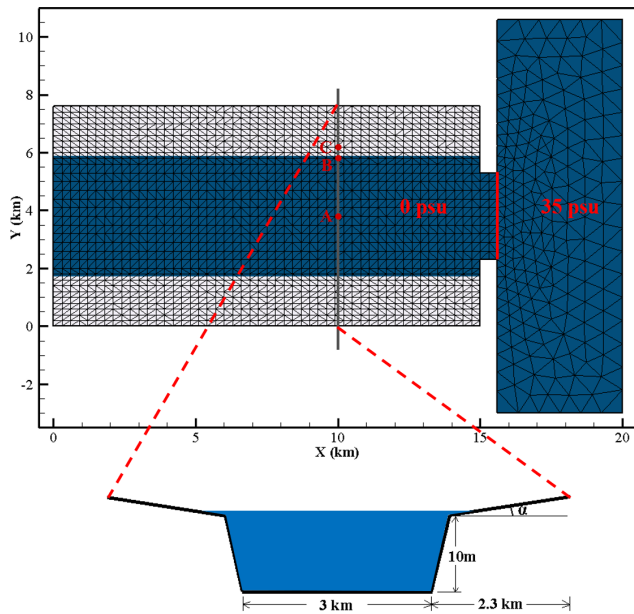


Figure 10. The calculation domain and its grid division of the semi-enclosed estuary. The minimum grid resolution is 200 m. The river channel cross-section is symmetric along the channel centreline and uniform along the channel direction. The initial water depth is 10.2 m and $\alpha = 0.0007$. Three characteristic points are marked: the channel centre (A), the WD boundary (B), and the initially dry location (C). The initial salinity is set to 0 on the left side of the solid red line and 35 psu on the right side.

ter content at this location to that at the river mouth. At $y = 100$ km, the simulated surface coastal current velocity is consistently 0.30 m s^{-1} across all grid resolutions, indicating minimal influence of grid resolution on current velocity. The percentage of freshwater transport at this location is 59.4 %, 57.5 %, and 60.6 % for grid resolutions of 1500, 750, and 375 m, respectively.

3.4 Saltwater enters an idealised semi-enclosed estuary

In addition to estuarine freshwater plumes, another phenomenon related to saltwater and freshwater interactions is saltwater intrusion into estuaries under tidal forcing. In this section, a semi-enclosed idealised river channel is configured, as shown in Fig. 10. This test case follows the setup described by Chen et al. (2022), with the computational mesh redefined accordingly. The channel is 15 km long, 3 km wide, and 10.2 m deep, with an initial salinity of 0. The channel features a riverbank slope of 0.033 and a floodplain slope of 0.0007. The right boundary is open to the sea, with a salinity of 35 psu, where a semi-diurnal tide with a 1 m amplitude is applied. The water temperature is uniformly set to 20°C . Given that existing 3D DG baroclinic models lack empirical validation for handling WD processes, this test examines the model's applicability and reliability in solving the salinity transport equation in the presence of WD

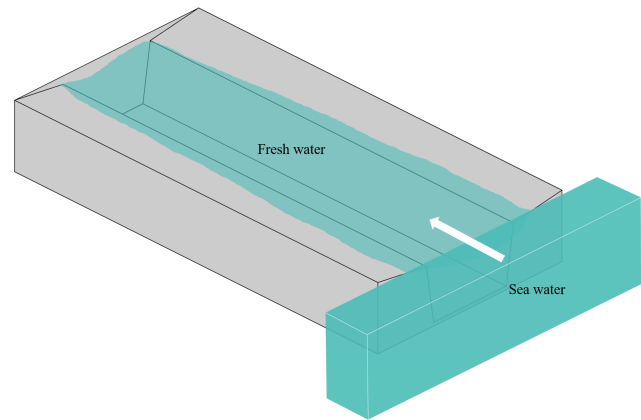


Figure 11. A 3D view of a flooding time.

boundaries. The computational domain features a grid resolution ranging from 200 to 800 m, with the vertical dimension divided into 10 layers. The model operates with a time step of 0.2 s. The linear state equation is applied again, with $\alpha_T = 0$, $\beta_S = 0.714$, $T_0 = 20^\circ\text{C}$, and $S_0 = 35$ psu, to update the density. The horizontal and vertical viscosities are set to $0.01 \text{ m}^2 \text{ s}^{-1}$, the bottom friction coefficient is set to 0.005, and the threshold water depth is set to 0.05 m. The limiters mentioned in Sect. 2.3 are all applied to maintain robustness.

During the flood tide, seawater from the open boundary flows into the semi-closed channel and overtops the floodplain (Fig. 11). Three representative points were selected to analyse temporal variations in tidal water elevation and velocities: site A, located at the centre of the channel; site B, near the WD interface on the top of the riverbank slope at the initial time; and site C, situated in the initially dry intertidal zone (Fig. 10). At sites A and B, the surface water elevations exhibit nearly identical variations over a lunar day. However, at site C, water levels show periodic fluctuations only during the 8 h corresponding to the high-water time, remaining 0 the rest of the time (Fig. 12a). For site A, the mean velocity in the along-channel direction exhibits well-defined periodic variation, while the mean velocity in the cross-channel direction remains close to 0, indicating that the velocity field within the channel is minimally affected by the WD boundary treatment. For site B, the mean velocity in the along-channel direction is generally similar to that of site A. However, during the two flood tide phases, the velocity is slightly reduced due to the influence of the wetting and drying boundary treatment. In contrast, the mean velocity in the cross-channel direction shows periodic variation consistent with the tidal cycle. At site C, velocity is only observed during high water levels, with both the magnitude and the direction of the velocity varying during the 8 h window of tidal inundation (Fig. 12b and c). Moreover, the salinity variations along the channel centreline are shown in Fig. 13. As seawater intrudes from the estuarine mouth, the salinity values within the channel progressively increase in the negative x

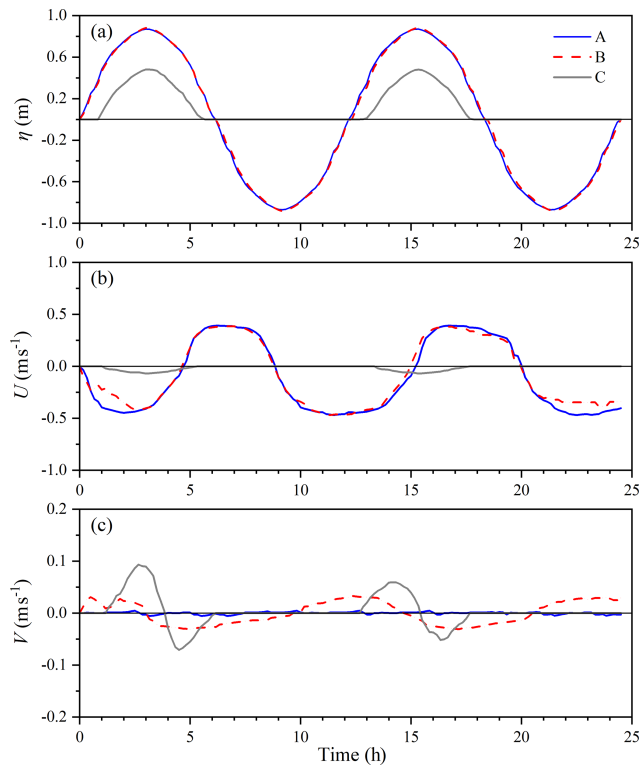


Figure 12. Time series of surface water elevation (a), depth-averaged velocity along the channel (b), and depth-averaged velocity across the channel (c) at points A, B, and C.

direction. Vertically, the salinity distribution exhibits a characteristic pattern of lower salinity near the surface and higher salinity near the bottom. Although this numerical experiment may not provide highly quantitative conclusions, the model's simulation results of the saltwater intrusion process involving WD interfaces are consistent with physical expectations.

4 Conclusions

This study presents the development of a novel 3D discontinuous Galerkin coastal and estuarine modelling system, DGCEMS. The model distinguishes itself from existing 3D DG-based ocean models by employing σ coordinates, a nonsplit-mode framework, and a quadrature-free nodal discontinuous Galerkin method. A semi-implicit Runge–Kutta scheme is applied in the model. The code is available at <https://doi.org/10.5281/zenodo.14803724> (Chen, 2025). Numerical tests show that the simulation results of the model have a second-order convergence for surface water elevation, horizontal velocity, and tracer fields, while the convergence of the vertical velocity field is approximately first order. The spurious mixing is also well controlled and comparable to the existing 3D DG coastal ocean model Thetis. The new model demonstrates low sensitivity to changes in grid resolution when simulating surface river plumes, consistently produc-

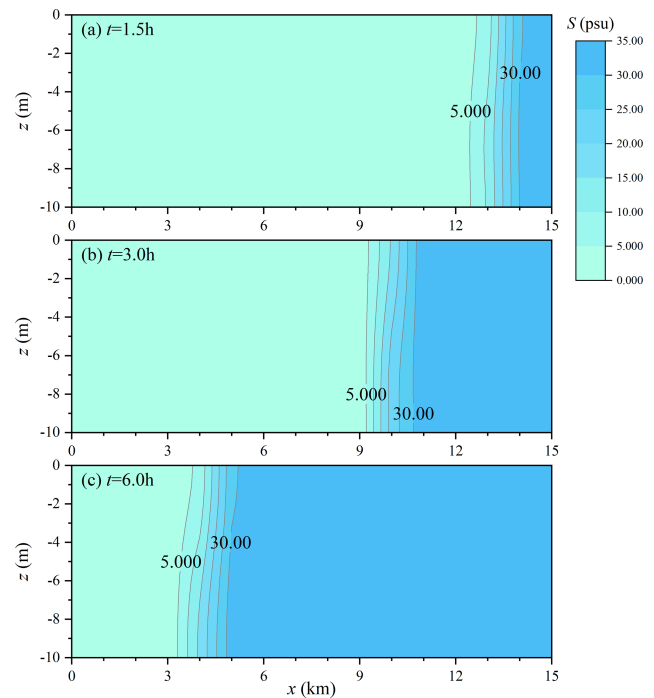


Figure 13. Salinity distribution along the channel centreline cross-section at 1.5, 3, and 6 h.

ing comparable results even with relatively coarse grids, and it has the capability to simulate salt–freshwater interactions in the presence of wetting and drying boundaries. In future work, the model will be further optimised in terms of its parallel computing strategy and overall solution framework to improve computational efficiency and scalability. Additional physical processes and modules, such as wave–current interactions, sediment transport, and biogeochemical dynamics, will also be incorporated to enhance the model's capability in simulating more complex and realistic estuarine and coastal systems.

Appendix A: Terms for the baroclinic manufactured solution test

Using Eqs. (28)–(30), the steady-state solution for other fields and terms are

$$\eta = 0, \quad (A1)$$

$$U = \frac{1}{6} \sin(3) \sin(2\pi x/L_x), \quad (A2)$$

$$V = \frac{-4}{3} \sin^2\left(\frac{1}{2}\right) \cos\left(\frac{\pi y}{L_y}\right), \quad (A3)$$

$$\omega = -\frac{D\pi}{3L_x L_y} \left(L_y \cos\left(\frac{2\pi x}{L_x}\right) (\sin(3) + \sin(3\sigma)) \right. \\ \left. + 2L_x \left(-\cos\left(\frac{1}{2}\right) + \cos\left(\frac{\sigma}{2}\right) \right) \sin\left(\frac{\pi y}{L_y}\right) \right), \quad (A4)$$

$$\frac{\partial DU}{\partial x} + \frac{\partial DV}{\partial y} = \frac{D\pi}{3L_x L_y} \left(L_y \sin(3) \cos\left(\frac{2\pi x}{L_x}\right) \right. \\ \left. + 4\sin^2\left(\frac{1}{2}\right) L_x \sin\left(\frac{\pi y}{L_y}\right) \right), \quad (A5)$$

$$\frac{\partial Duu}{\partial x} = \frac{D\pi}{2L_x} \cos^2(3\sigma) \sin\left(\frac{4\pi x}{L_x}\right), \quad (A6)$$

$$\frac{\partial Duu}{\partial y} = -\frac{D\pi}{6L_y} \cos(3\sigma) \sin\left(\frac{\sigma}{2}\right) \sin\left(\frac{2\pi x}{L_x}\right) \\ \times \sin\left(\frac{\pi y}{L_y}\right), \quad (A7)$$

$$\frac{\partial Duu}{\partial x} = \frac{D\pi}{3L_x} \cos(3\sigma) \sin\left(\frac{\sigma}{2}\right) \cos\left(\frac{2\pi x}{L_x}\right) \\ \times \cos\left(\frac{\pi y}{L_y}\right), \quad (A8)$$

$$\frac{\partial Dvv}{\partial y} = -\frac{D\pi}{9L_y} \sin^2\left(\frac{\sigma}{2}\right) \sin\left(\frac{2\pi y}{L_y}\right), \quad (A9)$$

$$\frac{\partial u\omega}{\partial \sigma} = \frac{D\pi \sin(3\sigma) \sin(2\pi x/L_x)}{2L_x L_y} \\ \times \left(L_y \cos\left(\frac{2\pi x}{L_x}\right) (\sin(3) + \sin(3\sigma)) \right. \\ \left. + 2L_x \left(-\cos\left(\frac{1}{2}\right) + \cos\left(\frac{\sigma}{2}\right) \right) \sin\left(\frac{\pi y}{L_y}\right) \right) \\ - \frac{D\pi \cos(3\sigma) \sin(2\pi x/L_x)}{6L_x L_y} \\ \times \left(3L_y \cos(3\sigma) \cos\left(\frac{2\pi x}{L_x}\right) \right. \\ \left. - L_x \sin\left(\frac{\sigma}{2}\right) \sin\left(\frac{\pi y}{L_y}\right) \right), \quad (A10)$$

$$\frac{\partial v\omega}{\partial \sigma} = -\frac{D\pi \cos(\sigma/2) \cos(\pi y/L_y)}{18L_x L_y} \\ \times \left(L_y \cos\left(\frac{2\pi x}{L_x}\right) (\sin(3) + \sin(3\sigma)) \right. \\ \left. + 2L_x \left(-\cos\left(\frac{1}{2}\right) + \cos\left(\frac{\sigma}{2}\right) \right) \sin\left(\frac{\pi y}{L_y}\right) \right) \\ - \frac{D\pi \cos(\pi y/L_y) \sin(\sigma/2)}{9L_x L_y} \\ \times \left(3L_y \cos(3\sigma) \cos\left(\frac{2\pi x}{L_x}\right) \right. \\ \left. - L_x \sin\left(\frac{\sigma}{2}\right) \sin\left(\frac{\pi y}{L_y}\right) \right), \quad (A11)$$

$$\frac{\partial DuT}{\partial x} = \frac{15D\pi \cos(3\sigma)}{4L_x} \left(4\cos\left(\frac{2\pi x}{L_x}\right) \right. \\ \left. + \cos(\sigma) \left(-\sin\left(\frac{\pi x}{L_x}\right) + 3\sin\left(\frac{3\pi x}{L_x}\right) \right) \right) \\ \times \sin\left(\frac{\pi y}{L_y}\right), \quad (A12)$$

$$\frac{\partial DvT}{\partial y} = \frac{5D\pi \sin(\sigma/2)}{L_y} \left(\cos(\sigma) \cos\left(\frac{2\pi y}{L_y}\right) \right. \\ \left. \times \sin\left(\frac{\pi x}{L_x}\right) - \sin\left(\frac{\pi y}{L_y}\right) \right), \quad (A13)$$

$$\frac{\partial \omega T}{\partial \sigma} = \frac{5D\pi \sin(\sigma) \sin(\pi x/L_x) \sin(\pi y/L_y)}{L_x L_y} \\ \times \left(L_y \cos\left(\frac{2\pi x}{L_x}\right) (\sin(3) + \sin(3\sigma)) \right. \\ \left. + 2L_x \left(-\cos\left(\frac{1}{2}\right) + \cos\left(\frac{\sigma}{2}\right) \right) \sin\left(\frac{\pi y}{L_y}\right) \right) \\ - \frac{5D\pi}{L_x L_y} \left(3L_y \cos(3\sigma) \cos\left(\frac{2\pi x}{L_x}\right) \right. \\ \left. - L_x \sin\left(\frac{\sigma}{2}\right) \sin\left(\frac{\pi y}{L_y}\right) \right) (1 + \cos(\sigma)) \\ \times \sin\left(\frac{\pi x}{L_x}\right) \sin\left(\frac{\pi y}{L_y}\right), \quad (A14)$$

$$S_{\text{baro},x} = -\frac{15\alpha_T D^2 g \pi}{L_x \rho_0} \cos\left(\frac{\pi x}{L_x}\right) \sin(\sigma) \sin\left(\frac{\pi y}{L_y}\right), \quad (A15)$$

$$S_{\text{baro},y} = -\frac{15\alpha_T D^2 g \pi}{L_y \rho_0} \cos\left(\frac{\pi y}{L_y}\right) \sin(\sigma) \sin\left(\frac{\pi x}{L_x}\right), \quad (A16)$$

$$S_{f,x} = -\frac{fD}{3} \cos\left(\frac{2\pi y}{L_y}\right) \sin\left(\frac{\sigma}{2}\right), \quad (A17)$$

$$S_{f,y} = \frac{fD}{2} \sin\left(\frac{2\pi x}{L_x}\right) \cos(3\sigma). \quad (A18)$$

The above terms are added to the right-hand side of Eq. (3) as source terms to balance the initial fields.

Code and data availability. The static version of the DGCEMS (v1.0.0) source code is available at <https://doi.org/10.5281/zenodo.14803724> (Chen, 2025). The code of DGCEMS is publicly available on GitHub via <https://github.com/ZerengChen/DGCEMS> (last access: 5 December 2024).

Data availability. No data sets were used in this article.

Author contributions. ZC: conceptualisation, methodology, software, validation, writing – original draft, visualisation. QZ: conceptualisation, writing – review and editing, supervision, project administration, funding acquisition. GR: methodology, software, supervision. YN: validation, formal analysis, visualisation.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. The authors gratefully acknowledge the financial support by the joint funds of the National Natural Science Foundation of China (grant no. U1906231) and the Chongqing Water Conservancy Science and Technology Project (grant no. CQSLK-2023004).

Financial support. This research has been supported by the joint funds of the National Natural Science Foundation of China (grant no. U1906231) and the Chongqing Water Conservancy Science and Technology Project (grant no. CQSLK-2023004).

Review statement. This paper was edited by Simone Marras and reviewed by two anonymous referees.

References

- Aizinger, V. and Dawson, C.: The local discontinuous Galerkin method for three-dimensional shallow water flow, *Comput. Method. Appl. M.*, 196, 734–746, <https://doi.org/10.1016/J.CMA.2006.04.010>, 2007.
- Bertsch, R., Glenis, V., and Kilsby, C.: Building level flood exposure analysis using a hydrodynamic model, *Environ. Modell. Softw.*, 156, 105490, <https://doi.org/10.1016/j.envsoft.2022.105490>, 2022.
- Blaise, S., Comblen, R., Legat, V., Remacle, J., Deleersnijder, E., and Lambrechts, J.: A discontinuous finite element baroclinic marine model on unstructured prismatic meshes: Part I: space discretization, *Ocean Dynam.*, 60, 1371–1393, <https://doi.org/10.1007/s10236-010-0358-3>, 2010.
- Burchard, H., Bolding, K., and Villarreal, M.: GOTM, a general ocean turbulence model: theory, implementation and test cases, Technical Report EUR 18745 EN, European Commission, Brussels, Belgium, 1999.
- Chen, C., Liu, H., and Beardsley, R.: An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: application to coastal ocean and estuaries, *J. Atmos. Oceanic Technol.*, 20, 159–186, [https://doi.org/10.1175/1520-0426\(2003\)020<0159:AUGFVT>2.0.CO;2](https://doi.org/10.1175/1520-0426(2003)020<0159:AUGFVT>2.0.CO;2), 2003.
- Chen, C., Qi, J., Liu, H., Beardsley, R., Lin, H., and Cowles, G.: A wet/dry point treatment method of FVCOM, part I: Stability experiments, *J. Mar. Sci. Eng.*, 10, <https://doi.org/10.3390/jmse10070896>, 2022.
- Chen, Z.: The discontinuous Galerkin coastal and estuarine modelling system (DGCEMS), Zenodo [code], <https://doi.org/10.5281/zenodo.14803724>, 2025.
- Chen, Z., Zhang, Q., Ran, G., and Nie, Y.: A wetting and drying approach for a mode-nonsplit discontinuous Galerkin hydrodynamic model with application to Laizhou Bay, *J. Mar. Sci. Eng.*, 12, 147, <https://doi.org/10.3390/jmse12010147>, 2024.
- Cheng, J., Liu, T., and Luo, H.: A hybrid reconstructed discontinuous Galerkin method for compressible flows on arbitrary grids, *Comput. Fluids*, 139, 68–79, <https://doi.org/10.1016/j.compfluid.2016.04.001>, 2016.
- Comblen, R., Blaise, S., Legat, V., Remacle, J., Deleersnijder, E., and Lambrechts, J.: A discontinuous finite element baroclinic marine model on unstructured prismatic meshes: Part II: implicit/explicit time discretization, *Ocean Dynam.*, 60, 1395–1414, <https://doi.org/10.1007/s10236-010-0357-4>, 2010.
- Conroy, C. and Kubatko, E.: *hp* discontinuous Galerkin methods for the vertical extent of the water column in coastal settings part I: Barotropic forcing, *J. Comput. Phys.*, 305, 1147–1171, <https://doi.org/10.1016/j.jcp.2015.10.038>, 2016.
- Dawson, C. and Aizinger, V.: A discontinuous Galerkin method for three-dimensional shallow water equations, *J. Sci. Comput.*, 22, 245–267, <https://doi.org/10.1007/s10915-004-4139-3>, 2005.
- Delandmeter, P.: Discontinuous Galerkin finite element modelling of geophysical and environmental flows, PhD thesis, Université Catholique de Louvain, 2017.
- Delandmeter, P., Lambrechts, J., Legat, V., Vallaey, V., Naithani, J., Thiery, W., Remacle, J.-F., and Deleersnijder, E.: A fully consistent and conservative vertically adaptive coordinate system for SLIM 3D v0.4 with an application to the thermocline oscillations of Lake Tanganyika, *Geosci. Model Dev.*, 11, 1161–1179, <https://doi.org/10.5194/gmd-11-1161-2018>, 2018.
- Epshteyn, Y. and Rivière, B.: Fully implicit discontinuous finite element methods for two-phase flow, *Appl. Numer. Math.*, 57, 383–401, <https://doi.org/10.1016/j.apnum.2006.04.004>, 2007.
- Faghih-Naini, S., Kuckuk, S., Aizinger, V., Zint, D., Grosso, R., and Köstler, H.: Quadrature-free discontinuous Galerkin method with code generation features for shallow water equations on automatically generated block-structured meshes, *Adv. Water Resour.*, 138, 103552, <https://doi.org/10.1016/j.advwatres.2020.103552>, 2020.

- Hesthaven, J. and Warburton, T.: Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Springer Science & Business Media, ISBN 978-0-387-72065-4, 2007.
- Ilicak, M., Adcroft, A., Griffies, S., and Hallberg, R.: Spurious dia-neutral mixing and the role of momentum closure, *Ocean Model.*, 45, 37–58, <https://doi.org/10.1016/j.ocemod.2011.10.003>, 2012.
- Jackett, D., McDougall, T., Feistel, R., Wright, D., and Griffies, S.: Algorithms for density, potential temperature, conservative temperature, and the freezing temperature of seawater, *J. Atmos. Oceanic Technol.*, 23, 1709–1728, <https://doi.org/10.1175/JTECH1946.1>, 2006.
- Kärnä, T., Kramer, S. C., Mitchell, L., Ham, D. A., Piggott, M. D., and Baptista, A. M.: Thetis coastal ocean model: discontinuous Galerkin discretization for the three-dimensional hydrostatic equations, *Geosci. Model Dev.*, 11, 4359–4382, <https://doi.org/10.5194/gmd-11-4359-2018>, 2018.
- Kowalik, Z. and Murty, T.: Numerical modeling of ocean dynamics, *Advanced Series on Ocean Engineering* (Vol. 5), World Scientific, <https://doi.org/10.1142/1970>, 1993.
- Le, H., Lambrechts, J., Ortleb, S., Gratiot, N., Deleersnijder, E., and Soares-Frazão, S.: An implicit wetting–drying algorithm for the discontinuous Galerkin method: Application to the Tonle Sap, Mekong River Basin, *Environ. Fluid Mech.*, 20, 923–951, <https://doi.org/10.1007/s10652-019-09732-7>, 2020.
- Lee, H.: Implicit discontinuous Galerkin scheme for shallow water equations, *J. Mech. Sci. Technol.*, 33, 3301–3310, <https://doi.org/10.1007/s12206-019-0625-2>, 2019.
- Lee, H.: Implicit discontinuous Galerkin scheme for discontinuous bathymetry in shallow water equations, *KSCE J. Civ. Eng.*, 24, 2694–2705, <https://doi.org/10.1007/s12205-020-2409-8>, 2020.
- Lesser, G., Roelvink, J., van Kester, J., and Stelling, G.: Development and validation of a three-dimensional morphological model, *Coast. Eng.*, 51, 883–915, <https://doi.org/10.1016/j.coastaleng.2004.07.014>, 2004.
- Li, L. and Zhang, Q.: A new vertex-based limiting approach for nodal discontinuous Galerkin methods on arbitrary unstructured meshes, *Comput. Fluids*, 159, 316–326, <https://doi.org/10.1016/j.compfluid.2017.10.016>, 2017.
- Li, W.: Quadrature-free forms of discontinuous Galerkin methods in solving compressible flows on triangular and tetrahedral grids, *Math. Comput. Simulat.*, <https://doi.org/10.1016/j.matcom.2023.11.030>, 2024.
- MacCready, P., Hetland, R., and Geyer, W.: Long-term isohaline salt balance in an estuary, *Cont. Shelf Res.*, 22, 1591–1601, [https://doi.org/10.1016/S0278-4343\(02\)00023-7](https://doi.org/10.1016/S0278-4343(02)00023-7), 2002.
- Medeiros, S. and Hagen, S.: Review of wetting and drying algorithms for numerical tidal flow models, *Int. J. Numer. Meth. Fl.*, 71, 473–487, <https://doi.org/10.1002/fld.3668>, 2013.
- Moulinec, C., Denis, C., Pham, C. T., Rougé, D., Hervouet, J., Razafindrakoto, E., Barber, R., Emerson, D., and Gu, X.: TELEMAC: An efficient hydrodynamics suite for massively parallel architectures, *Comput. Fluids*, 51, 30–34, <https://doi.org/10.1016/j.compfluid.2011.07.003>, 2011.
- Nair, R. D.: Quadrature-free implementation of a discontinuous Galerkin global shallow-water model via flux correction procedure, *Mon. Weather Rev.*, 143, 1335–1346, <https://doi.org/10.1175/MWR-D-14-00174.1>, 2015.
- Petersen, M., Jacobsen, D., Ringler, T., Hecht, M., and Maltrud, M.: Evaluation of the arbitrary Lagrangian–Eulerian vertical coordinate method in the MPAS-Ocean model, *Ocean Model.*, 86, 93–113, <https://doi.org/10.1016/j.ocemod.2014.12.004>, 2015.
- Ran, G. Q., Zhang, Q. H., and Chen, Z. R.: Development of a three-dimensional hydrodynamic model based on the discontinuous Galerkin method, *Water*, 15, <https://doi.org/10.3390/w15010156>, 2022.
- Sanz-Ramos, M., López-Gómez, D., Bladé, E., and Dehghan-Souraki, D.: A CUDA Fortran GPU-parallelised hydrodynamic tool for high-resolution and long-term eco-hydraulic modelling, *Environ. Model. Softw.*, 161, 105628, <https://doi.org/10.1016/j.envsoft.2023.105628>, 2023.
- Shahbazi, K.: An explicit expression for the penalty parameter of the interior penalty method, *J. Comput. Phys.*, 205, 401–407, <https://doi.org/10.1016/j.jcp.2004.11.017>, 2005.
- Shchepetkin, A. and McWilliams, J.: The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model, *Ocean Model.*, 9, 347–404, <https://doi.org/10.1016/j.ocemod.2004.08.002>, 2005.
- Smagorinsky, J.: General circulation experiments with the primitive equations: I. The basic experiment, *Mon. Weather Rev.*, 91, 99–164, [https://doi.org/10.1175/1520-0493\(1963\)091<0099:GCEWTP>2.3.CO;2](https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2), 1963.
- Toro, E.: Shock-capturing Methods for Free-surface Shallow Flows, Wiley-Blackwell, ISBN 978-0-471-98766-6, 2001.
- Vallaes, V.: Discontinuous Galerkin Finite Element Modelling of Estuarine and Plume Dynamics, UCL-Université Catholique de Louvain, <http://hdl.handle.net/2078.1/203013>, last access: 30 August 2018.
- Wu, H.: Evaluation on the tracer simulation under different advection schemes in the ocean model, *Anthropocene Coasts*, 6, 1, <https://doi.org/10.1007/s44218-023-00017-7>, 2023.
- Xing, Y. and Zhang, X.: Positivity-preserving well-balanced discontinuous Galerkin methods for the shallow water equations on unstructured triangular meshes, *J. Sci. Comput.*, 57, 19–41, <https://doi.org/10.1007/s10915-013-9695-y>, 2013.
- Zhang, Y., Ye, F., Stanev, E., and Grashorn, S.: Seamless cross-scale modeling with SCHISM, *Ocean Model.*, 102, 64–81, <https://doi.org/10.1016/j.ocemod.2016.05.002>, 2016.
- Zhou, W., Melack, J., and MacIntyre, S.: Hydrodynamic modeling of floodplains with inundated trees and floating plants, *Environ. Model. Softw.*, 179, 106117, <https://doi.org/10.1016/j.envsoft.2024.106117>, 2024.