

High-order discontinuous Galerkin schemes for the barotropic-baroclinic splitting in two-dimensional layered shallow water equations

Yao Gahounzo^{a,c,*}, Michal Kopera^{a,*}, Robert L., Higdon^b, Eric P. Chassignet^c, Alan Wallcraft^c, Alexandra Bozec^c

^a*Department of Mathematics, Boise State University, , Boise, 83716, ID, USA*

^b*Department of Mathematics, Oregon State University, , Corvallis, 97331-4605, OR, USA*

^c*Center for Ocean-Atmospheric Prediction Studies (COAPS), Florida State University, , Tallahassee, 32306-2741, FL, USA*

Abstract

The separation of fast (barotropic) and slow (baroclinic) motions into subsystems through barotropic-baroclinic splitting has been widely adopted in layered ocean circulation models. To date, the majority of models use finite difference or finite volume methods alongside this splitting technique. In this paper, we present an extension of the work in Higdon (2015) to two horizontal dimensions using an arbitrary high-order, nodal discontinuous Galerkin (DG) method for the resulting split subsystems to develop an ocean model. We carry out numerical tests to demonstrate the performance of the proposed schemes, and the numerical results of the double-gyre test are compared with those of the HYbrid Coordinate Ocean Model (HYCOM). The parallel performance shows that the new model has a larger per-degree-of-freedom computational cost compared to HYCOM, but achieves the same result in terms of resolved kinetic energy in an order of magnitude faster time, with fewer computational resources, and maintains good parallel efficiency even with very few grid elements per computational core.

Keywords:

Multilayer shallow water equations, DG method, Double-gyre circulation,

*Corresponding authors

Email addresses: yaogahounzo@u.boisestate.edu (Yao Gahounzo),
michalkopera@boisestate.edu (Michal Kopera)

1. Introduction

Multiple frameworks have been used to model large-scale ocean dynamics computationally. A relatively simple system consists of shallow water equations, which model a constant-density fluid for which the depth is much smaller than the horizontal length scales of the motions of interest. This scaling implies that the fluid is approximately hydrostatic. The shallow-water system captures some of the large-scale dynamics of the ocean, but a more complete picture requires an accounting for the effects of vertical variations of the density of the fluid.

This accounting requires the usage of a vertical coordinate within the fluid, and several different types of vertical coordinates are used in models of the general circulation of the ocean. These include level (z), terrain-fitted (σ), and isopycnic coordinates [1, 2]. The last of these is a density-related quantity that is nearly a material coordinate within the ocean’s interior; in this case, a vertical discretization divides the fluid into layers. The Hybrid Coordinate Ocean Model (HYCOM) [3, 4] uses a hybrid coordinate that employs all of these, with different coordinates in different regions.

In an isopycnic coordinate system, a vertical coordinate divides the fluid into layers that are physically distinct, and the fluid can then be approximated as a stack of layers, each having a constant density. This is the case that is addressed in the present paper; a future goal is to extend this work to a hybrid vertical coordinate. A stack of shallow-water models is also used in the multi-layer shallow-water equations (MLSWE) such as Audusse [5], Bouchut and Zeitlin [6], Izem and Seaid [7], Zhang et al. [8] among others.

1.1. Ocean time-scales

Ocean dynamics involves several space and time scales, from small-scale turbulence through internal and external gravity waves and slow-moving Rossby waves to the large-scale thermohaline circulation, which covers the global ocean and can take decades to complete. The external gravity waves are the fastest motions, with speeds up to two orders of magnitude higher than the speed of internal waves or slower motion waves [9]. Many ocean models use explicit time-stepping schemes due to their ease of implementation. However, these methods require relatively small time step sizes imposed

by the Courant-Friedrichs-Lewy (CFL) condition to ensure stable numerical simulations, resulting in computationally expensive simulations.

Given the vast separation of time scales between external and other waves, a popular approach has been to separate the layered system into two subsystems: fast (barotropic) and slow (baroclinic), known as barotropic-baroclinic splitting [10, 11, 12]. A two-dimensional subsystem resembling shallow-water equations models the barotropic motions of the entire ocean column, whereas the three-dimensional subsystem represents the baroclinic motions. Each of these subsystems can be solved with different time step sizes [13, 12, 14]. A large time step is used to advance the baroclinic subsystem, while a smaller time step is used to advance the barotropic subsystem, each satisfying the appropriate CFL conditions. In this paper, we follow the splitting approach presented in Higdon [11] and introduce the strong stability-preserving Runge-Kutta (SSPRK) methods [15] for the barotropic subsystem. The SSPRK methods were previously used in the splitting scheme by Lan et al. [16].

1.2. Spatial discretizations

Considering the various timescales of relevant ocean processes implies that ocean modeling is intrinsically multiscale, and accurately representing these physical processes presents computational challenges. The numerical methods used in ocean models should include desirable features such as low artificial dissipation, efficient resolution of localized flow features, and the capability to handle complex coastline geometries. High-order element-based discontinuous Galerkin (DG) methods promise to address those needs [17]. Similarly to the finite volume method, the domain is split into non-overlapping elements of (in principle) arbitrary shape and size, which allows the representation of complex geometries and control over the mesh resolution. DG uses a weak formulation of the governing equations and involves element-wise integration where the solution in each element is represented as an arbitrary-order polynomial, resulting in high-order spatial approximation and low dispersive errors [18, 19]. In contrast to finite and spectral elements, where neighboring elements are continuously coupled, the DG method allows for discontinuous solutions. Another benefit is its excellent parallel efficiency [20, 17, 19, 21].

Previous applications of DG methods in geophysical modeling include the shallow water flow models [22, 23, 24, 25, 26, 27]. The shallow water equations (SWE) DG models have also been used in the tsunami modeling (e.g., Blaise et al. [28], Bonev et al. [29], Arpaia et al. [30]) and storm surge

modeling (e.g., Dawson et al. [31], Beisiegel et al. [32]). Examples of the non-shallow water ocean models that use DG methods include the Imperial College Ocean Model (ICOM) [33], which solves the three-dimensional non-hydrostatic Boussinesq equations using an unstructured mesh and an implicit time-stepping scheme. The Second-generation Louvain-la-Neuve Iceocean Model (SLIM) model [34] uses DG to solve the hydrostatic Boussinesq equations with applications ranging from rivers to coastal oceans. Another example is Thetis [35], an unstructured grid model with second-order accuracy in space and time, developed to simulate the coastal ocean and river-estuary-plume systems.

Some distinctions between the preceding works and the present work include the following. The formulation given by Higdon [36, 11, 37] begins with an arbitrary vertical coordinate that includes level, isopycnic, terrain-fitted, and hybrid coordinates as special cases. While much of the development in those papers concerns the isopycnic case and an approximation with a stack of layers of constant density for simplicity and definiteness, the results for that case can be extended to the general case. Additionally, in Higdon’s work, the governing equations for mass density and momentum density are written in conservation form, with no assumptions about compressibility. The barotropic (i.e., fast) equations for both mass and momentum are obtained by a vertical summation of the layer equations, resulting in the barotropic mass and momentum equations being also in conservation form. For some previous splittings, such as those of Bleck and Smith [10], Higdon [9], and Kärnä et al. [35], the momentum equations are not in such a conservation form.

This paper presents a DG implementation of the multilayer shallow water equations based on Higdon [11] and extended to two horizontal dimensions. We implement the model in the Galerkin Numerical Modeling Environment (GNuME) framework. The work in Giraldo et al. [38] and the present work are connected via this framework. The GNuME framework uses an arbitrary polynomial basis function expansion and offers a choice of continuous Galerkin and discontinuous Galerkin methods [39]. The framework was previously used to construct the Non-Hydrostatic Unified Model of the Ocean [40], and we call the current MLSWE implementation h-NUMO to signify the hydrostatic aspect of the model. This study aims to evaluate the suitability of the DG-based h-NUMO model for general ocean circulation by comparing it with the HYCOM model using a well-established double-gyre test case. The remainder of the paper is organized as follows. In Section 2, we provide

a brief overview of the HYCOM model and introduce the governing equations and barotropic-baroclinic splitting in h-NUMO. Section 3 is devoted to the DG discretizations of the governing equations; the multistep time method for the barotropic-baroclinic equations is presented in Section 4. In Section 5, we provide numerical verification using benchmarking test cases to illustrate the capabilities of h-NUMO and compare with HYCOM; and lastly, the conclusion in Section 6.

2. Governing equations and barotropic-baroclinic splitting

2.1. Brief overview of HYCOM

HYCOM is a finite volume model developed on a C-grid with its dynamical core predominantly based on the MICOM model [3]. The model uses the variable splitting as outlined in Section 2.4 and split-explicit scheme to solve barotropic and baroclinic equations (see Bleck and Smith [10]). However, in HYCOM the splitting of the governing equations is different from the splitting used in the present paper, and the momentum equations are not in conservation form.

The model employs all three vertical coordinates (z , σ , and isopycnic), with different coordinates in different regions with the optimal distribution chosen at every time step. HYCOM, with its hybrid vertical coordinate generator, transitions smoothly between coordinate types via the continuity equation [41]. However, one can run HYCOM by using only one of these vertical coordinates. We did that in this paper by configuring HYCOM in purely isopycnic coordinates (stacked shallow water equations) for comparison with h-NUMO.

2.2. Layer equations

Assume that the fluid is in hydrostatic balance, and discretize the vertical dimension of the ocean into N_l layers of constant density ρ_k , where $k = 1, \dots, N_l$ is the layer index increasing downward (see Fig. 1). We consider equations for conservation of mass and momentum in layer k in the form

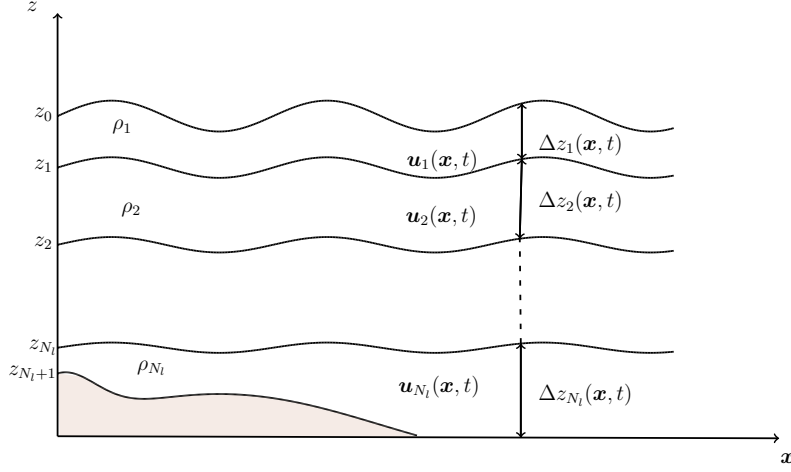


Figure 1: Illustration of the isopycnal layered shallow water system. The quantity $\mathbf{u}_k(\mathbf{x}, t)$ denotes the horizontal velocity, ρ_k is the density, and $\Delta z_k(\mathbf{x}, t)$ is the height of k -th layer and is related to the vertical pressure increment Δp_k over layer k

$$\frac{\partial \Delta p_k}{\partial t} + \nabla \cdot (\mathbf{u}_k \Delta p_k) = 0, \quad (1)$$

$$\begin{aligned} \frac{\partial \mathbf{u}_k \Delta p_k}{\partial t} + \mathbf{A}_k + f \mathbf{u}_k^\perp \Delta p_k = & -\nabla H_k + g(p_{k-1} \nabla z_{k-1} - p_k \nabla z_k + \Delta \boldsymbol{\tau}_k) \\ & + \nabla \cdot (A_H \Delta p_k \nabla \mathbf{u}_k). \end{aligned} \quad (2)$$

This system is an analog for the case of two horizontal dimensions, of the mass and momentum equations that were developed by Higdon [36] for the case of one horizontal dimension. In this system, Δp_k is the pressure at the bottom of layer k minus the pressure at the top of layer k , so Δp_k serves as a mass variable for layer k , $\mathbf{u}_k = (u_k, v_k)$ is the horizontal velocity in layer k , f is the Coriolis parameter defined in a beta-plane with $\mathbf{u}_k^\perp = (-v_k, u_k)^T$ and $\Delta \boldsymbol{\tau}_k = (\Delta \tau_k^x, \Delta \tau_k^y)$ is the shear stress which is the combination of wind stress and bottom stress (discussed later). The viscosity is denoted by A_H , where the subscript H indicates that this is the viscosity in the horizontal dimensions, $z_{k-1}(\mathbf{x}, t)$ and $z_k(\mathbf{x}, t)$ with $\mathbf{x} = (x, y)$ are the interface elevations of the top and bottom of layer k , respectively; we measure elevation with respect to the free surface at rest. The layer interface elevation $z_k(x, t)$ is

related to the layer pressure through $\Delta p_k(\mathbf{x}, t) = g\rho_k\Delta z_k$. Thus $z_k(\mathbf{x}, t)$ is computed as

$$\begin{aligned}\Delta z_k(\mathbf{x}, t) &= \Delta p_k(\mathbf{x}, t)/(g\rho_k) \\ z_k(\mathbf{x}, t) - z_{k+1}(\mathbf{x}, t) &= \Delta p_k(\mathbf{x}, t)/(g\rho_k) \\ z_k(\mathbf{x}, t) &= z_{k+1}(\mathbf{x}, t) + \Delta p_k(\mathbf{x}, t)/(g\rho_k).\end{aligned}$$

The bottom interface of the bottom layer $k = N_l$ is $z_{N_l+1} = z_b$, where $z_b(\mathbf{x})$ is the bottom topography. Then the remaining elevation $z_k(\mathbf{x}, t), k = N_l - 1, \dots, 1$ are computed from the bottom to the top.

We define $p_k(\mathbf{x}, t) = P(\mathbf{x}, z_k, t)$ and $p_{k-1}(\mathbf{x}, t) = P(\mathbf{x}, z_{k-1}, t)$ as the pressures at the bottom and top of the layer k . The term

$$H_k(\mathbf{x}, t) = g \int_{z_k}^{z_{k-1}} P(\mathbf{x}, z, t) dz \quad (3)$$

is the vertical integral of the horizontal pressure force, which we evaluate using the hydrostatic assumption as

$$H_k(\mathbf{x}, t) = g \int_{z_k}^{z_{k-1}} P dz = \int_{p_{k-1}}^{p_k} \alpha_k p dp = \frac{1}{2} \alpha_k (p_k^2(\mathbf{x}, t) - p_{k-1}^2(\mathbf{x}, t)), \quad (4)$$

where $\alpha_k = \frac{1}{\rho_k}$ is the specific volume. Finally, we denote the advection term as

$$\mathbf{A}_k = \frac{\partial}{\partial x} [u_k(\mathbf{u}_k \Delta p_k)] + \frac{\partial}{\partial y} [v_k(\mathbf{u}_k \Delta p_k)].$$

The governing equations (1) and (2) can be viewed as defining a stack of shallow-water models similar to Audusse [5]. The terms in the momentum equation (2) are standard, except for the expression $-\nabla H_k + g(p_{k-1} \nabla z_{k-1} - p_k \nabla z_k)$. The terms that involve ∇z_{k-1} and ∇z_k represent the lateral pressure forcing acting on layer k due to tilting interfaces at the top and bottom of that layer, and the term $-\nabla H_k$ represents the lateral pressure forcing within layer k . This representation of pressure forcing follows from a derivation of governing equations given by Higdon [36]. That derivation begins with the governing equations expressed in a general vertical coordinate s . In this system, the pressure forcing presents a particular problem, as the pressure gradient must be in a direction that is truly horizontal, which might not be in a direction of constant s . This situation causes difficulties that are described in Section 2.4 of that paper.

These difficulties are related to the task of expressing the momentum equations as point-wise partial differential equations in terms of s and the horizontal coordinates. The analysis in Higdon [36] side-steps these difficulties by proceeding directly to an integral weak form, which is all that is really needed for a DG method. Consider the fluid that resides in a given layer on a given horizontal grid cell, multiply by a smooth test function on that spatial region, and integrate by parts when appropriate. The result is the weak formulation described in Appendix A of the present paper. However, for the sake of clarity, the present Section gives the result of integrating only in the vertical direction, and the horizontal discretization is considered later. For the vertically-integrated momentum equation, the pressure forcing is $-g \int_{z_k}^{z_{k+1}} \nabla P(\mathbf{x}, z, t) dz$, and a calculation shows that this is equal to the pressure terms stated in Eq (2). For the case of the shallow water equations for a single-layer fluid of constant density, Higdon [42] shows that the above method for obtaining a weak integral form produces a formulation of pressure forcing that is automatically well-balanced.

2.3. Barotropic equations

We obtain the barotropic equations by vertical summation of the layer equations (1)-(2) and introducing barotropic variables representing the fast motion of the entire water column.

$$p_b = \sum_{k=1}^{N_l} \Delta p_k \quad (5)$$

is g times the mass per unit horizontal area for the entire water column, and it will serve as the barotropic mass variable, and

$$p_b \bar{\mathbf{u}} = \sum_{k=1}^{N_l} \mathbf{u}_k \Delta p_k \quad (6)$$

is the barotropic momentum, where $\bar{\mathbf{u}}$ is the mass-weighted vertical average of $\mathbf{u}_k$ over all layers. The barotropic equations are given by

$$\frac{\partial p_b}{\partial t} + \nabla \cdot (p_b \bar{\mathbf{u}}) = 0, \quad (7)$$

$$\frac{\partial p_b \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{A}} + f p_b \bar{\mathbf{u}}^\perp = -\nabla H + g (p_0 \nabla z_0 - p_{N_l} \nabla z_{N_l} + \Delta \tau) + \mathbf{L}_{AH}, \quad (8)$$

where $\mathbf{L}_{A_H}$ is the vertical summation of the viscosity term in the layer momentum equation (2), the barotropic advection term is

$$\bar{\mathbf{A}} = \frac{\partial}{\partial x} \begin{bmatrix} Q_u^x \\ Q_v^x \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} Q_u^y \\ Q_v^y \end{bmatrix},$$

with

$$Q_u^x(\mathbf{x}, t) = \sum_{k=1}^{N_l} u_k(u_k \Delta p_k), \quad Q_u^y(\mathbf{x}, t) = \sum_{k=1}^{N_l} v_k(u_k \Delta p_k), \quad (9)$$

$$Q_v^x(\mathbf{x}, t) = \sum_{k=1}^{N_l} u_k(v_k \Delta p_k), \quad Q_v^y(\mathbf{x}, t) = \sum_{k=1}^{N_l} v_k(v_k \Delta p_k), \quad (10)$$

and the total vertical integration of the horizontal pressure force gives

$$H(\mathbf{x}, t) = \sum_{k=1}^{N_l} H_k(\mathbf{x}, t). \quad (11)$$

2.4. Splitting of the prognostic variables

Following the splitting approach in Bleck and Smith (1990) also used in the HYCOM model, let p'_b denote the value of p_b when the fluid is in the total rest state, and let $\eta(\mathbf{x}, t)$ be the relative perturbation in $p_b(\mathbf{x}, t)$, compared to $p'_b(\mathbf{x})$. Then typically $|\eta| \ll 1$, and

$$p_b(\mathbf{x}, t) = (1 + \eta(\mathbf{x}, t)) p'_b(\mathbf{x}). \quad (12)$$

Introducing this to the barotropic mass equation (7) yields

$$\frac{\partial p'_b \eta}{\partial t} + \nabla \cdot (p_b \bar{\mathbf{u}}) = 0. \quad (13)$$

Layer velocity is then split into barotropic and baroclinic modes as

$$\mathbf{u}_k(\mathbf{x}, t) = \mathbf{u}'_k(\mathbf{x}, t) + \bar{\mathbf{u}}(\mathbf{x}, t), \quad (14)$$

where $\mathbf{u}'_k$ is the baroclinic velocity. The baroclinic velocity $\mathbf{u}'_k$ varies mainly on the slow time scales, and the fast signals in the system are confined mainly to the barotropic velocity $\bar{\mathbf{u}}$. The combination of Eq. (14) with (5) and (6)

shows that the baroclinic velocities have mass-weighted vertical average equal to zero

$$\sum_{k=1}^{N_l} u'_k \Delta p_k = 0, \quad \sum_{k=1}^{N_l} v'_k \Delta p_k = 0. \quad (15)$$

For a splitting of the pressure field, define a baroclinic mass variable $\Delta p'_k$ for layer k by

$$\Delta p_k(\mathbf{x}, t) = (1 + \eta(\mathbf{x}, t)) \Delta p'_k(\mathbf{x}, t). \quad (16)$$

This splitting is based on the idea that external motions cause all fluid layers to thicken or thin by approximately the same proportion [10, 11]. The factor $1 + \eta$ approximately captures the fast external signal, and the $\Delta p'_k$ varies mainly on the slow time scales.

Now let $p'_0(\mathbf{x}, t) = 0$, and define the baroclinic pressure at the bottom of layer k by

$$p'_k(\mathbf{x}, t) = \sum_{s=1}^k \Delta p'_s(\mathbf{x}, t). \quad (17)$$

for $1 \leq k \leq N_l$. It then follows that $\Delta p'_k(\mathbf{x}, t) = p'_k(\mathbf{x}, t) - p'_{k-1}(\mathbf{x}, t)$ and that the total pressure at the bottom of layer k is

$$p_k(\mathbf{x}, t) = p_0(\mathbf{x}, t) + \sum_{s=1}^k \Delta p_s(\mathbf{x}, t) = p_0(\mathbf{x}, t) + (1 + \eta(\mathbf{x}, t)) p'_k(\mathbf{x}, t), \quad (18)$$

where $p_0(\mathbf{x}, t)$ is the atmospheric pressure.

If the atmospheric pressure p_0 is constant, then p_0 can be deleted from formulas that involve the pressure forcing, in a sense that is described in the discussions after equation (22) in Higdon [36] and after equation (66) in Higdon [11]. This enables some simplifications for this special case, but we will not pursue this matter here.

The preceding splittings of the velocity and mass fields imply that the barotropic momentum fluxes in Eq.(9)-(10) can be written in the form

$$Q_u^x(\mathbf{x}, t) = \bar{u}(p_b \bar{u}) + p_b \left(\sum_{k=1}^{N_l} u'_k u'_k \frac{\Delta p'_k}{p'_b} \right), \quad Q_u^y(\mathbf{x}, t) = \bar{v}(p_b \bar{u}) + S p_b u, \quad (19)$$

$$Q_v^x(\mathbf{x}, t) = \bar{u}(p_b \bar{v}) + Sp_b u, Q_v^y(\mathbf{x}, t) = \bar{v}(p_b \bar{v}) + p_b \left(\sum_{k=1}^{N_l} v'_k v'_k \frac{\Delta p'_k}{p'_b} \right), \quad (20)$$

where $Sp_b u = p_b \left(\sum_{k=1}^{N_l} u'_k v'_k \frac{\Delta p'_k}{p'_b} \right)$. This follows from the property (15) of baroclinic velocities and the relation $\Delta p_k / p_b = \Delta p'_k / p'_b$, which follows from the properties (12) and (16) of the mass splitting. The vertical sums in the barotropic momentum fluxes (19)-(20) then vary primarily on the slow time scales. The governing equations for the present model consist of the layer equations (1)-(2) and the barotropic equations (13) and (8), combined with enforcement of consistency (discussed in Section 4.2) between those two subsystems.

2.5. Shear stress terms

To complete the discussion of Eq. 2 we define the shear stress term

$$\Delta \tau_k = \tau_{k-1}(\mathbf{x}, t) - \tau_k(\mathbf{x}, t), \quad (21)$$

where τ_{k-1} and τ_k are the shear stresses acting on the top and bottom interfaces of layer k , respectively. In this study, we neglect the friction between layers and only consider the wind stress acting on the top of the fluid and the bottom stress acting on the bottom of the fluid. Thus, the shear stress is given by

$$\tau_k = \tau_w + \tau_b \quad (22)$$

where $\tau_{w,b}$ are the wind and bottom stresses. We refer to Section 5 for details on how wind stress is prescribed for each test case.

Wind stress acts on the free surface of the fluid; however, we apply it as linear decay to zero over a predetermined vertical distance as in HYCOM [10], which prevents large forcing from being applied to arbitrarily thin layers. For a prescribed depth h_{stress} at which the wind stress decays to zero, the pressure at that depth is $p_{stress} = g\rho_1 h_{stress}$. The linear decay $\tau_{w,k}^*$ of the wind stress τ_w is

$$\tau_{w,k}^* = D_{w,k} \tau_w, \quad (23)$$

where

$$D_{w,k} = \frac{\min(p_k - p_0, p_{stress}) - \min(p_{k-1} - p_0, p_{stress})}{p_{stress}}.$$

Here p_{k-1} and p_k are pressure at the top and bottom of the k -th layer.

The bottom stress can be parameterized in linear or quadratic form. In this work, we use the linear bottom stress given by

$$\tau_b = c_d \mathbf{u}_{N_l}, \quad (24)$$

where $\mathbf{u}_{N_l}$ is the bottom velocity, and c_d (s^{-1}) is the friction coefficient. Like wind stress, the bottom stress is applied as linear decay to zero over a prescribed distance.

3. Discontinuous Galerkin method

We follow the DG method in [43] to discretize the split layered system (1), (2), (8) and (13), and here we outline the method for completeness. The element-based DG method in 2D decomposes the physical domain $\Omega \in \mathbb{R}^2$ into N_e non-overlapping elements Ω_e

$$\Omega = \bigcup_e \Omega_e, \quad (25)$$

where each element can be of arbitrary size. Implementing the DG method requires computing integrals over elements and their edges. In this work, we use quadrilateral elements, but in principle, other shapes are possible (e.g., triangles). Since the element may vary in size and shape, it would be very expensive to compute the integrals individually over each element in the mesh. A more effective approach is to use a change of variables to obtain an integral on a reference element [44, 45]. We introduce a two-dimensional reference element $I = [-1, 1]^2$ so that the coordinates $\mathbf{x} \in \Omega_e$ in the physical domain are mapped to coordinates $\boldsymbol{\xi}(\xi, \zeta) = \Theta(\mathbf{x})$ within the reference element using a bijective mapping $\Theta : \Omega_e \rightarrow I$.

For a solution variable q , let q_N be its approximation on the element Ω_e in the basis expansion ψ

$$q_N^{(e)}(\mathbf{x}, t) = \sum_{m=1}^{M_N} \psi_m(\boldsymbol{\xi}_m) q_m^{(e)}(t), \quad (26)$$

where the superscript $^{(e)}$ denotes the element-based entity, $\boldsymbol{\xi}_m = \Theta(\mathbf{x}_{ij})$, $q_N^{(e)} = q_N(\mathbf{x}_m, t)$ is the grid point values and $M_N = (N + 1)^2$ with N the polynomial order of one-dimensional Lagrange polynomial. The basis functions $\psi_m = l_i(\xi) \otimes l_j(\zeta)$ are a tensor product of one-dimensional Lagrange

polynomials l_i and l_j of order N associated with $M = N + 1$ Legendre-Gauss-Lobatto points $\boldsymbol{\xi}$. The local index m in the 2D is mapped from the 1D local indices (i, j) as $m = i + 1 + jM$ corresponding to a distinct nodal point in each element. To compute the integrals, we use the LGL nodal points as quadrature points. We refer to Giraldo [43] for more details on the basis functions and quadrature points.

In what follows, we define the quantities a_L and a_R as the values of a on either side of an element edge, where the labels L and R are chosen arbitrarily, and define the "average" value $\{\{a\}\}$ and "jumps" $[[a]]$ on an element edge as

$$\{\{a\}\} = (a_L + a_R)/2, \quad [[a]] = a_L \mathbf{n}_L + a_R \mathbf{n}_R, \quad [[\mathbf{a}]] = \mathbf{a}_L \cdot \mathbf{n}_L + \mathbf{a}_R \cdot \mathbf{n}_R,$$

where $\mathbf{n}_L$ and $\mathbf{n}_R$ are unit normal vectors pointing in directions L and R , respectively. The spatial discretization of DG methods is widely presented in the literature; however, for the spatial discretization of the governing equations presented in this work, see Appendix A.

3.1. Numerical flux terms

Since the DG method allows for discontinuities between element solutions, we need to calculate numerical fluxes between the element edges. Here, we outline the choices we have made for the numerical fluxes computation in different equations:

1. In solving the barotropic subsystem, we solve a Riemann problem for linearized barotropic equations. Riemann solvers [46] are a natural choice for computing the interface fluxes in finite volume and DG methods. Section 6 of Higdon [42], which concerns the shallow water equations, gives a detailed development and solution of the Riemann problem for this case, with the final conclusions stated in equations (47)-(48) of that paper. To derive the linear barotropic equations, we assumed that the velocity is small and neglected the Coriolis term. Denoting the wave speed by $c = \sqrt{gD}$, where D denotes the constant depth of the fluid at the rest state, the Riemann solution to the linearized equations is

$$(p'_b \eta)^\dagger = \{\{p'_b \eta\}\} + \frac{1}{2c} [[p'_b \bar{\mathbf{u}}]] \quad (27)$$

$$(p'_b \bar{\mathbf{u}})^\dagger = \{\{p'_b \bar{\mathbf{u}}\}\} + \frac{c}{2} [[p'_b \eta]]. \quad (28)$$

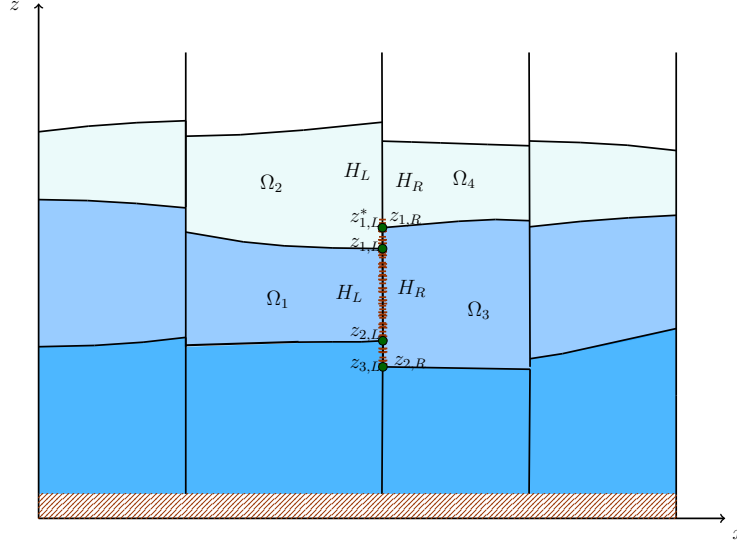


Figure 2: Multilayer system with spatial variation in pressure forcings on each cell edge. The green dots represent the points of intersections of the edge (red markers) between the left and right elements Ω_1 and Ω_3 .

These interpolated values of $p'_b \bar{\mathbf{u}}$ are used as values of mass fluxes $p_b \bar{\mathbf{u}}$ at cell edges in the weak form of the barotropic mass equation, and the interpolated values of $p'_b \eta$ are used to compute the vertically-integrated horizontal pressure forcing at cell edges in the weak forms of the barotropic momentum equations. Further details on these matters are given in Section 6.3 of Higdon [42]. The numerical fluxes in equations (27) and (28) are applied in the computation of the right-hand side $\mathbf{r}_{b,U}^{(e)}$ and $\mathbf{r}_{b,V}^{(e)}$ in the equations (A.16) and (A.16).

2. In the layered equation for the mass variable (A.1) and the momentum equations (A.2) and (A.3), we use centered fluxes. To compute the flux contribution of a variable q at a given edge, we take the average of its values from that edge's left-hand and right-hand sides. We also tried upwind fluxes, but the computational results were very similar to the centered flux results. We refer to Section 5.7 for more details on these comparisons.

3.2. Pressure forcing at cell edges

We compute the pressure within each grid element using Eq. (4) and determine the pressure at each cell edge following the idea developed in Higdon [11]. Here, we outline the main ideas of this method and refer to the earlier paper for details. Consider the baroclinic state of the fluid as highlighted in Fig. (2) where the interfaces of the layers are not necessarily continuous across cell edges due to the density difference between the layers. The total pressure at the left and right sides of an edge is given by

$$p_L(z) = (1 + \eta^\dagger) p'_L(z), \quad (29)$$

$$p_R(z) = (1 + \eta^\dagger) p'_R(z), \quad (30)$$

where $\eta^\dagger$ is from Eq. (27), p'_L and p'_R denote the left and right side values of p' . The pressure at a cell edge can be approximated as

$$P(z) = \frac{1}{2} (p_L(z) + p_R(z)), \quad (31)$$

and the contribution from the pressure along the cell edge in the weak forms (A.2) and (A.3) at the edge Γ_e can be represented as

$$H_k^\dagger = g \int_{z_{k,R}}^{z_{k-1,R}} P(z) dz = \frac{1}{2} g \int_{z_{k,R}}^{z_{k-1,R}} p_L(z) dz + \frac{1}{2} g \int_{z_{k,R}}^{z_{k-1,R}} p_R(z) dz. \quad (32)$$

For the configuration illustrated in Figure 2, consider the vertically integrated pressure force exerted on the fluid in region Ω_3 by the fluid that is immediately to the left of Ω_3 . The later expression at the dashed edge (red markers) is then

$$H_2^\dagger = \frac{1}{2} g \int_{z_{2,R}}^{z_{1,R}} p_L(z) dz + \frac{1}{2} g \int_{z_{2,R}}^{z_{1,R}} p_R(z) dz. \quad (33)$$

The second integral is straightforward since the interval of integration is the entire vertical extent of layer k . The first integral requires that the interval of integration be represented as a union of subintervals, and that leads to

$$\frac{1}{2} g \int_{z_{2,R}}^{z_{1,R}} p_L(z) dz = \frac{1}{2} g \int_{z_{3,L}}^{z_{2,L}} p_L(z) dz + \frac{1}{2} g \int_{z_{2,L}}^{z_{1,L}} p_L(z) dz + \frac{1}{2} g \int_{z_{1,L}}^{z_{1,L}^*} p_L(z) dz. \quad (34)$$

The integrand $p_L(z)$ uses left limits of p' , which is why we use the left-side values $(z_{k,L})$ of z_k instead of right-side values $(z_{k,R})$ in the integrals that appear on the right side of Eq. (34).

3.3. Viscosity terms

We rewrite the viscous term $\nabla \cdot (A_H \Delta p_k \nabla \mathbf{u}_k)$ in the baroclinic equations (2) by introducing split variables and assuming constant viscosity A_H

$$\nabla \cdot (A_H \Delta p_k \nabla \mathbf{u}_k) = A_H \nabla \cdot ((1 + \eta) \Delta p'_k \nabla (\mathbf{u}'_k + \bar{\mathbf{u}})). \quad (35)$$

Further assuming that $\eta(\mathbf{x}, t)$ is negligible for this term, we have $(1 + \eta) \Delta p'_k \approx \Delta p'_k$ and arrive at

$$\nabla \cdot (A_H \Delta p_k \nabla \mathbf{u}_k) = A_H \nabla \cdot (\Delta p'_k \mathbf{G}_{\mathbf{u},k} + \Delta p'_k \mathbf{G}_{\bar{\mathbf{u}},b}), \quad (36)$$

where $\mathbf{G}_{\mathbf{u},k} = \nabla \mathbf{u}'_k$ and $\mathbf{G}_{\bar{\mathbf{u}},b} = \nabla \bar{\mathbf{u}}$.

The barotropic viscosity term is the sum of the layer viscosity terms

$$\mathbf{L}_{A_H} = A_H \nabla \cdot \left(\sum_{k=1}^{N_l} \Delta p'_k \mathbf{G}_{\mathbf{u},k} + \mathbf{G}_{\bar{\mathbf{u}},b} \sum_{k=1}^{N_l} \Delta p'_k \right). \quad (37)$$

We implement the viscosity terms using the local discontinuous Galerkin method and centered numerical flux [36, 47, 48, 49, 43].

4. Time integration

4.1. Methods for the layer equations and barotropic equations

For the layer equations, (1)-(2), we use a two-level predictor-corrector time method that is described in detail in Higdon [11]. This method uses a relatively long time step that is appropriate for the slow motions in the system; denote this time step by $\Delta t = t_{n+1} - t_n$, where t_n and t_{n+1} are consecutive time levels.

For the vertically-integrated barotropic equations, we use a shorter time step $\Delta t_{btp} = \Delta t / N_{btp}$, where N_{btp} is a positive integer. In contrast to the two-level method used in Higdon [11] for the barotropic equations, here we use a strong-stability-preserving Runge-Kutta method [50], as it enables greater efficiency by allowing a longer time step. The particular method used here has order 3 and has 5 stages, and we denote the method by SSPRK35. It should be noted that in Ruuth [50], the method is denoted as SSPRK(5,3).

We anticipate that, in the near future, we will undertake a more systematic and thorough investigation of Runge-Kutta methods and related methods for usage in the present context.

4.2. Consistency between the layer equations and the barotropic equations

At the end of each of the long time steps, the vertical sums of the layer variables Δp_k and $\mathbf{u}_k \Delta p_k$ should equal the barotropic variables p_b and $p_b \bar{\mathbf{u}}$, respectively. However, this is not necessarily the case, as different numerical approaches are used to solve the two sets of equations. To correct such inconsistencies, we make adjustments to Δp_k and $\mathbf{u}_k \Delta p_k$ at each long time step. These adjustments are at the level of numerical truncation error, since in the analytical case the vertical sums of the layer variables are exactly equal to the corresponding barotropic variables.

For the mass equations, we achieve consistency with a flux-adjustment method that is described in detail by Higdon [11]. Here, the lateral mass fluxes in the layers, for a given long time interval $[t_n, t_{n+1}]$, are adjusted so that their vertical sums are equal to the time averages of the barotropic fluxes over all of the short substeps of $[t_n, t_{n+1}]$. It is noted in Section 4.2.3 of Higdon [11] that this adjustment provides a kind of time filtering for the layer mass equations. Also, the adjustment of lateral mass fluxes does not entail any transport of mass between different fluid layers, which is consistent with the fact that the governing equations (1)-(2) do not allow for any exchange of mass between layers.

For the momentum equations, consistency is obtained at each horizontal location, at each long time step, by adding a small depth-independent velocity to the horizontal velocity in each layer. This depth-independent velocity amounts to an adjustment to the model's representation of the barotropic velocity $\bar{\mathbf{u}}$. The exact values of $\bar{\mathbf{u}}$ vary on the fast time scale, which ultimately is the motivation for using a barotropic-baroclinic splitting. Before the adjustment of $\bar{\mathbf{u}}$, for a given time step and horizontal location, the barotropic velocity that is available is the value that can be obtained from the solution of the layer equations, which are computed with the long time step. A calculation shows that the adjustment of velocity has the effect of replacing the long-time-step value of $\bar{\mathbf{u}}$ with the value that is obtained stably with short barotropic substeps. In analogy with the mass equations, this adjustment provides a kind of time filtering for the momentum equations.

The time filtering in the layer mass and layer momentum equations enables a long time step to be used for these equations, even though the layer equations themselves admit motions on the fast time scales. The layer equations with this filtering can be regarded as “baroclinic” equations for modeling the slow motions in the system. In Section 2.4, we focus on variable

splitting, and in this Section, we complete barotropic-baroclinic splitting with time methods. We summarize the splitting model in the Algorithm 1.

Algorithm 1 Summary of the time splitting method

Require: Prognostic variables at time t_n

Prediction step ▷ for a long baroclinic time step

- 1: Solve the barotropic equations
- 2: **for** $m = 1, N_{btp}$ **do**
- 3: call the SSPRK35 method
- 4: **end for**
- 5: Solve the baroclinic equations
- 6: Enforce consistency between the two systems

Correction step ▷ for a long baroclinic time step

- 7: Solve the barotropic equations using the average between the solutions from the predicted step and from the time t_n
- 8: **for** $m = 1, N_{btp}$ **do**
- 9: call the SSPRK35 method
- 10: **end for**
- 11: Solve the baroclinic equations
- 12: Enforce consistency between the two systems

5. Results

To demonstrate the capabilities of the model, we first evaluated the mass conservation of the split system in h-NUMO by running a lake-at-rest test case. Then, we used a double-gyre test case [51] to compare h-NUMO and HYCOM across a range of resolutions, order of polynomials, and viscosity. In h-NUMO, we define the resolution as an average distance between nodal points. The grid consists of elements, each of which, in turn, has a grid of nodal points used to construct approximation polynomials. Since the degrees of freedom are located at nodal points, this was a natural choice for defining resolution.

In our simulations, we used the p4est library [52] for managing mesh and related data structures and partitioning the data between parallel processes. We generate structured grids using p4est and for unstructured grids in Section 5.5, we used Gmsh [53] software to create a primary grid for p4est.

The DG method has a Courant-Friedrichs-Lewy (CFL) number that decreases with the order of the polynomial approximation. The time steps used for the barotropic and baroclinic solvers are selected as follows. The barotropic time step Δt_{btp} is chosen according to the DG method time restriction [54] as follows

$$\Delta t \leq \frac{C \times \min\{\Delta x, \Delta y\}}{d(2N + 1) \times |\lambda_{max}|}, \quad (38)$$

where N is the polynomial order, d is the dimension space (here $d = 2$), C is defined in [50], $\lambda_{max} = \max(|\mathbf{u}_b| - \sqrt{gD}, |\mathbf{u}_b| + \sqrt{gD})$ is approximation of the maximum wave speed, and Δx and Δy are the element size along x and y directions respectively. The baroclinic time step is $\Delta t = N_{btp} \Delta t_{btp}$, and by following HYCOM we set $N_{btp} = 20$ in the double-gyre simulations.

5.1. Well-balanced test

In the standard formulation of SWE or MLSWE, the pressure term in the momentum equation involves the gradient of elevation of the bottom boundary of the fluid layer, which can act as a static forcing term that causes a stationary fluid to move. To avoid this issue Higdon [42] proves and verifies for SWE that representing the pressure term in momentum equations as an integral weak form (equations (A.2) and (A.3)) ensures the system is well-balanced. This proof applies to a single-layer hydrostatic fluid of constant density. For the multi-layer case, the same method of proof can be applied on any grid cell for which no layer interface intersects the bottom topography in the interior of that cell.

We verify the well-balanced property of the DG schemes for the multi-layer equations, where we evaluate the h-NUMO with the lake at rest test. We have initialized fluid with densities $\rho_k = 1027.01037 + 0.2110 \times (k - 1)$ kg/m³, $k = 1, \dots, N_l$ and the layer interface positions $z_k = -40/N_l$ m, where N_l is the number of layers. The horizontal extend of the domain was $(x, y) \in [0, 2000] \times [0, 2000]$ m² with wall boundary conditions on both ends and the bottom topography is given by

$$Z_b(x, y) = 3 \left(1 + \cos \left(\frac{\pi r}{250} \right) \right), \quad (39)$$

where $r = \sqrt{(x - 1000)^2 + (y - 1000)^2}$ as shown in Fig. 3.

For this test, we use a 4th order polynomial interpolation, resulting in a 50 m grid resolution, and the test ran until $t = 5$ days simulation time. The

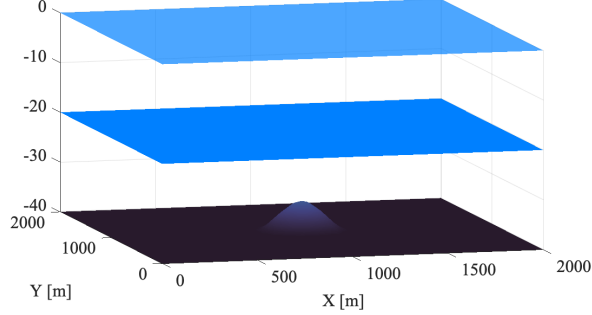


Figure 3: Initial conditions of lake-at-reset free surface and interface of the second layer with no-flat bottom topography.

barotropic time step $\Delta t_{btp} = 1.8 \text{ s}$ is chosen according to the DG method time restriction [55]. The baroclinic time step is $\Delta t = 100 \text{ s}$, resulting in 56 barotropic substeps per baroclinic time step.

In this test case, in the absence of external forcing (e.g., wind), the free surface should remain stationary, and water velocity should be zero throughout the simulation. Fig. 4 panel (a) illustrates the elevation of the free surface above the initial state at the cross-section ($y = 1000 \text{ m}$) at $t = 3$ hours and panel (b) shows the maximum deviation (L_∞ norm) of the free surface as a function of time until $t = 5$ days. This time corresponds to the barotropic wave traveling approximately 12000 times through the domain at the speed approximated by $c \approx \sqrt{gD}$ with $D = 40 \text{ m}$, which is sufficient to show the long-term behavior of the test case. We see that the L_∞ norm remains at 10^{-13} level throughout the simulation, which indicates that the spurious errors do not significantly grow with time. In panels (c) and (d), we present the cross-section of the water velocity at 3 hours and 5 days, respectively. The change in the velocity does not significantly grow with time. The increase in the number of layers from 2 to 20 layers has not deteriorated both the free-surface and velocity results, suggesting that the model does well in preserving the lake at rest conditions.

5.2. Perturbation of baroclinic wave propagation

To test whether h-NUMO captures the wave propagation speeds correctly, we add a small perturbation to the interface of the second layer of the well-balanced test case with a flat bottom topography. The initial interface position of the layers are $z_0 = 0 \text{ m}$, $z_1 = -20 + 0.5 \left(1 + \cos\left(\frac{\pi x}{250}\right)\right) \text{ m}$ and

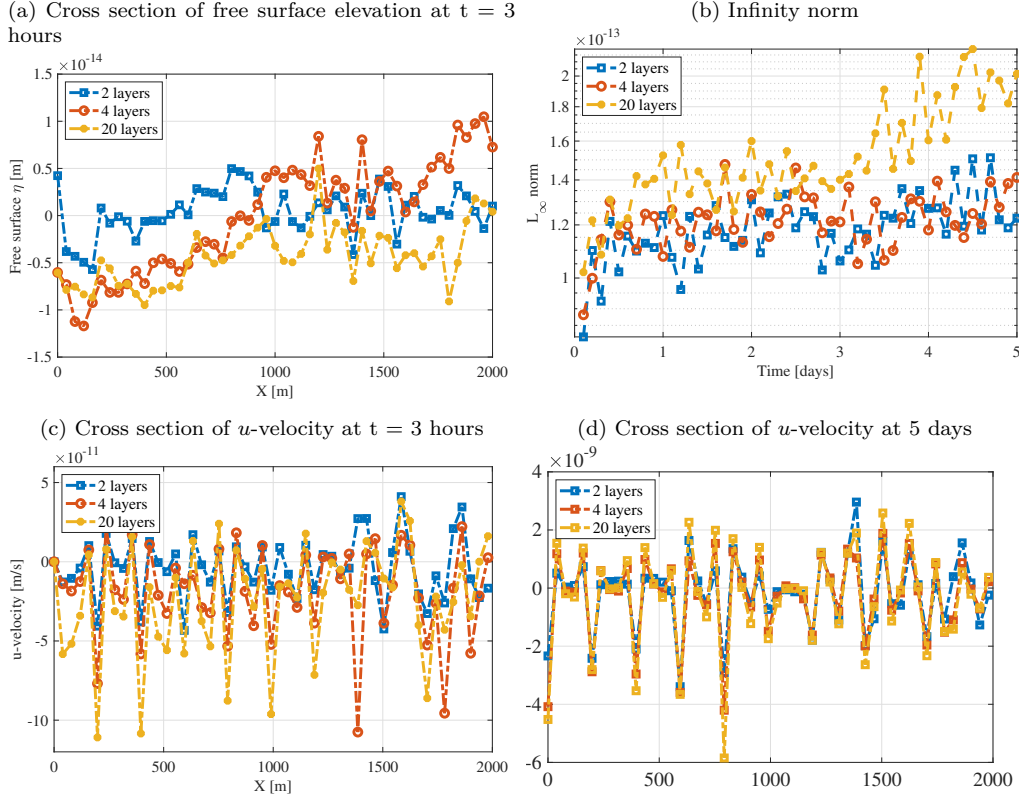


Figure 4: Free surface. Panel (a) shows a cross-section of the free surface at time $t = 3$ hours with 2, 4, and 20 layers, and panel (b) shows the infinity norm of the free surface solution over 5 days. Panel (c) and (d) show the cross-section of the u -velocity in the top layer at 3 hours and 5 days respectively.

$z_2 = -40$ m. The viscosity, wind stress, and bottom friction are neglected for this test. The test case set-up is shown in Fig. 5, with the initial condition (with exaggerated vertical scale) in panel (a) and the solution at $t = 3$ hours in panel (b).

We ran the simulation for different thicknesses of the top layer (while keeping the total water depth at $D = 40$ m) and measured the speed with which the baroclinic perturbation of the layer interface is moving. Fig. 6 shows the comparison of those measurements with the analytical prediction by Mandli [56]. The yellow dashed line represents the absolute error between the theoretical and measured wave speeds, which remains below 10^{-3} regardless of the chosen layer thickness. The theoretical wave speeds formulation

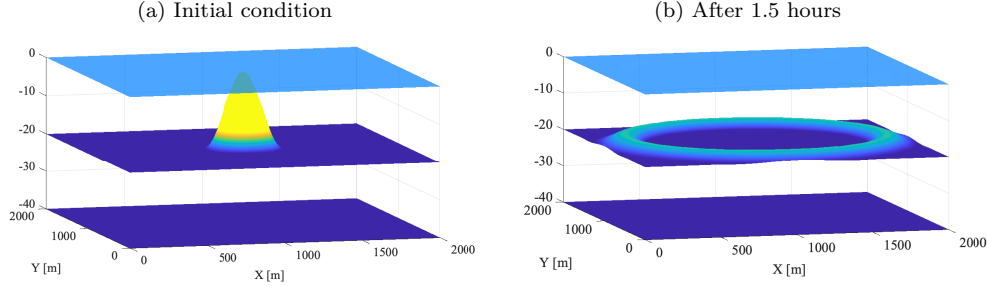


Figure 5: Free surface and interface of the second layer. Panel (a) shows the initial conditions with the perturbation in the second layer, and panel (b) shows solutions after 1.5 hours of simulation. The perturbation vertical scale is exaggerated in both panels.

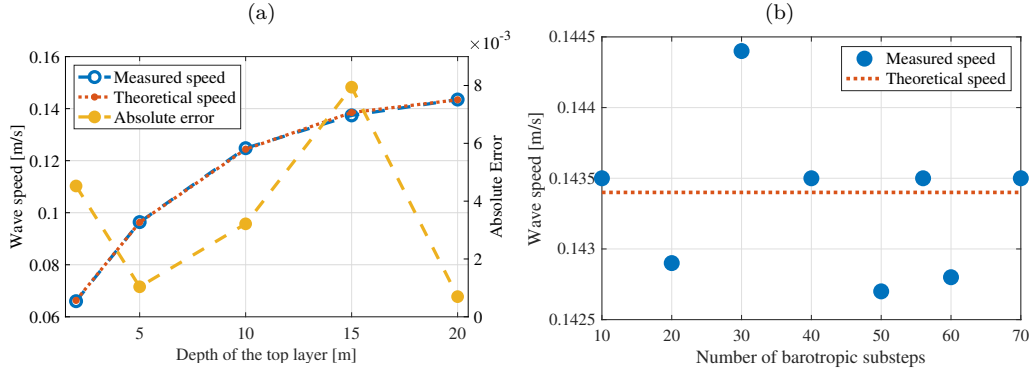


Figure 6: Wave speed as a function of depth of the first layer and absolute error between the theoretical and measured speed (a). Comparison of the wave speeds obtain using different numbers of barotropic substeps per baroclinic time step (b). The baroclinic time step is $\Delta t = N_{btp} \Delta t_{btp}$, where the barotropic time step is fixed at $\Delta t_{btp} = 1.8$ s.

in Mandli [56] is an approximation, and there is also measurement error, so having absolute errors close to 10^{-3} confirms that our scheme accurately captures baroclinic waves.

We also analyze how different numbers of barotropic substeps per baroclinic time step, N_{btp} , affect the wave speed. We consider $\Delta z_1 = \Delta z_2 = 20$ m, a fixed $\Delta t_{btp} = 1.8$ s and calculate $\Delta t = N_{btp} \Delta t_{btp}$. Fig. 6 (b) shows the measured wave speed for different N_{btp} , where the highest we can take to have a stable simulation for this problem is 70. We observe that the measured wave speeds oscillate around the theoretical speed ($c_{bcl} = 0.1434$ m/s). The results in panel (b) show that the wave speed does not depend on the number of

N_{btp} .

5.3. Double-gyre circulation

We consider an idealized double-gyre test [51] to validate h-NUMO’s ability to simulate the mesoscale and submesoscale processes and compare the results with HYCOM. The domain is a closed rectangular ocean basin with a flat bottom. The forcing is spatially varying wind stress with intense western boundary currents, which, together with Coriolis force, results in a counter-clockwise circulation in the northern part and a clockwise circulation in the southern part of the domain.

The horizontal extent is $L = 2000$ km in both the zonal and the meridional direction. The depth of the basin is $D = 10$ km consisting of two layers, with the upper and lower layers initially having $\Delta z_1 = 1.5$ km and $\Delta z_2 = 8.5$ km depths, respectively. The densities in the layers are $\rho_1 = 1027.01037$ kg/m³, $\rho_2 = 1027.22136$ kg/m³. The Coriolis force is prescribed using a beta-plane approximation centered at 45° N, with a parameter $f = f_0 + \beta(y - L/2)$, where $f_0 = 9.3 \times 10^{-4}$ s⁻¹ and $\beta = 2^{-11}$ m/s. We consider two different values of the horizontal viscosity $A_H = 50$ m²/s and $A_H = 500$ m²/s, the dimensionless bottom drag coefficient in the linear bottom stress (Section 2.5) is $c_d = 10^{-7}$ s⁻¹, and we assume no shear stress between layers. The system is forced by a purely zonal wind stress $\tau_w = (\tau_x, 0)$, where $\tau_x = -\tau_0 \cos(2\pi y/L)$, and $\tau_0 = 0.1$ N/m². We considered two different velocity boundary conditions: free-slip and no-slip. Each model year consists of 360 days, divided into 12 months, with 30 days per month.

We use HYCOM simulations with resolutions of 10 km and 20 km as references and design the h-NUMO runs to match those settings. We configured HYCOM in the same way as h-NUMO using purely isopycnic coordinates with the same number of layers, densities, wind stress, and linear bottom drag. For HYCOM, we define the resolution as the width of a finite volume grid cell, while in h-NUMO, we use the average distance between nodal points. An important aspect of the DG method is that the resolution can be controlled by both the element size and the interpolating polynomial order, so it is possible to have comparable resolutions using small elements with low polynomial order and large elements with high polynomial order. However, one has to keep in mind that the information carried by a degree of freedom in the nodal polynomial expansion used in the DG methods is not the same as in the case of the discretization used in HYCOM, as it represents not only a point value of the solution but is a weight used in the high-order

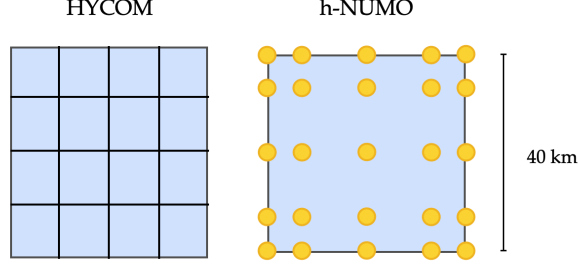


Figure 7: Schematic of the grid in HYCOM (left) and h-NUMO (right). In both cases, the gridded area is a square with a side of 40km. HYCOM covers it with a uniform grid of 4x4 volumes. h-NUMO uses a single element with a grid of 5x5 non-uniformly (4th order polynomial) distributed Legendre-Gauss-Lobatto points. For the sake of comparison, we consider both cases as having the same "resolution".

polynomial expansion, which contributes to the continuous approximation of the solution inside the entire element. Fig. 7 shows a schematic of both HYCOM and h-NUMO grids for 10 km resolution. We used 2nd, 4th, and 6th-order polynomials in this study.

5.3.1. Simulations with free-slip boundary condition

Fig. 8 shows the sea-surface height (SSH), which is the difference between the average depth of the fluid at rest and the total depth of the fluid at a given time, for 20 km and 10 km resolutions using free-slip boundary condition. For the 20 km resolution, h-NUMO was configured with 4-th polynomial order and time step $\Delta t = 600$ s for the baroclinic equations and $\Delta t_{btp} = 30$ s for the barotropic equations, while in HYCOM $\Delta t = 450$ s, $\Delta t_{btp} = 22.5$ s. The value of Δt_{btp} is limited by the CFL condition, which is different for both methods. For both codes, we kept the ratio of 20 barotropic substeps per one baroclinic time step. Due to the Coriolis effect and the wind stress, the flow develops two gyres with counterclockwise circulation in the northern half and clockwise in the southern half of the domain. As the simulation progresses, the flow becomes more complex and develops eddies by year 10 (not shown here) for viscosity $A_H = 50$ m²/s. Fig. 8 (a)-(b) shows that the eddies are more pronounced by year 20. For the higher viscosity $A_H = 500$ m²/s, some eddies start appearing in the northern boundary of the domain by year 20 (top panels (c)-(d)). Overall, the jet in the middle of the western boundary is similar in both models, and the general flow patterns are the

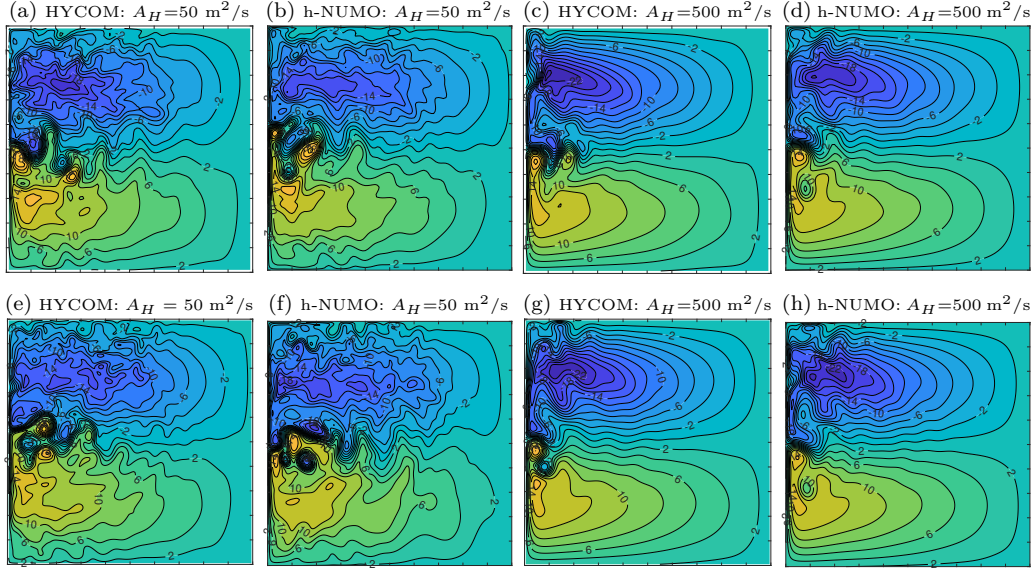


Figure 8: Snapshots of the sea-surface height at 20 years with h-NUMO and HYCOM using the free-slip boundary condition and viscosity $A_H = 50 \text{ m}^2/\text{s}$ and $A_H = 500 \text{ m}^2/\text{s}$. The results in top panels (a)-(d) are obtained with 20 km resolution, while the results in bottom panels (e)-(h) are obtained with 10 km resolution. The contour interval is 2 centimeters, with lower elevations in the northern region and higher elevations in the southern region.

same. The solutions from the two models are not point-wise identical due to the different numerical schemes used and the sensitivity of the non-linear equations to perturbations. The differences between snapshots are partially due to the phase shift in the oscillations of the currents.

To obtain the results in Fig. 8 (e)-(h) with 10 km resolution (where we increase the number of elements), we decreased the barotropic time step to $\Delta t_{btp} = 20 \text{ s}$ while keeping the 20 barotropic substeps to a baroclinic time step ratio in h-NUMO. In HYCOM, the baroclinic time step was $\Delta t = 225 \text{ s}$, and the barotropic time step was $\Delta t_{btp} = 11.25 \text{ s}$. Fig. 8 (e)-(f) shows that at a higher resolution, the flow is more complex, and the middle western boundary jet is stronger in h-NUMO compared to HYCOM. Similar to the 20 km resolution, the general circulation of the flow is similar in both codes, with more developed meanders and eddies for viscosity $A_H = 50 \text{ m}^2/\text{s}$. In both codes, Fig. 8 (c)-(d) and (g)-(h) show that the eddies are less noticeable for viscosity $A_H = 500 \text{ m}^2/\text{s}$ than the viscosity $A_H = 50 \text{ m}^2/\text{s}$, which makes

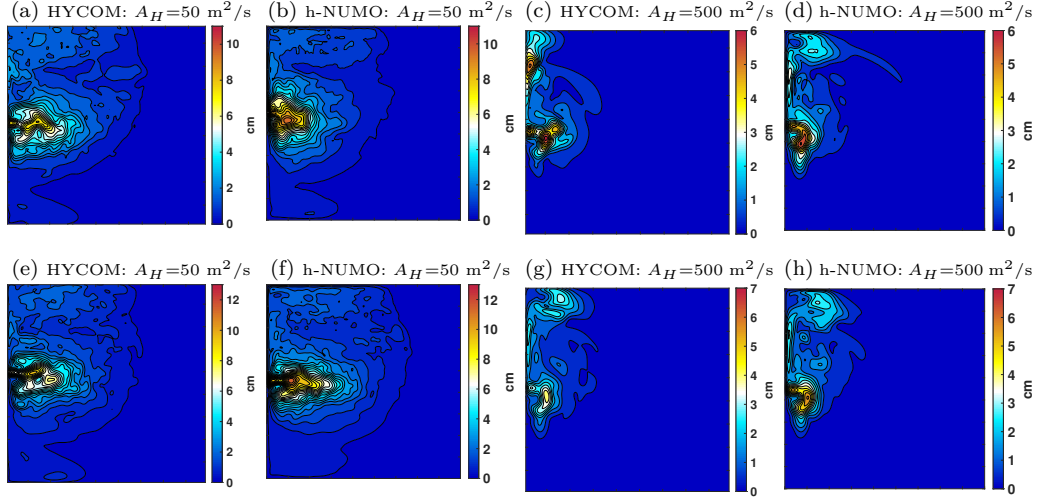


Figure 9: Comparison of sea surface height variability over the last five years (15-20) for the viscosity value $A_H = 50 \text{ m}^2/\text{s}$, $A_H = 500 \text{ m}^2/\text{s}$ and the free-slip boundary condition. The results in top panels (a)-(d) are obtained with 20 km resolution, while the results in bottom panels (e)-(h) are obtained with 10 km resolution. The contour line is after every 5 millimeters.

sense as we expect higher viscosity to diffuse more energy.

To analyze the redistribution of water in the ocean and the ocean dynamics across the resolved scales in both models, Fig. 9 shows a comparison of the sea surface height variability for the last 5 (15-20) years of the model runs at 20 km and 10 km resolutions. In panels (a)-(b) and (e)-(f) of Fig. 9, where the viscosity is $50 \text{ m}^2/\text{s}$, both codes have similar SSH variability patterns. However, the variability is higher in the middle of the western boundary in h-NUMO than in HYCOM. Similarly, at a higher viscosity $500 \text{ m}^2/\text{s}$ in panels (c)-(d) and (g)-(h), the SSH variability patterns remain the same but with slightly higher activity toward the west-northern boundary in HYCOM compared to h-NUMO. At the same, higher activity is observed in the middle-western boundary of h-NUMO. These results of SSH variability confirm that the flow is more developed in the middle-western boundary in h-NUMO than in HYCOM.

Fig. 10 shows plots of the total kinetic energy KE of the flow as a function of simulation time, where h-NUMO runs with 2nd, 4th, and 6th polynomial order are compared with HYCOM results. More precisely, in each graph, the quantity plotted is the pressure-weighted kinetic energy per unit horizontal

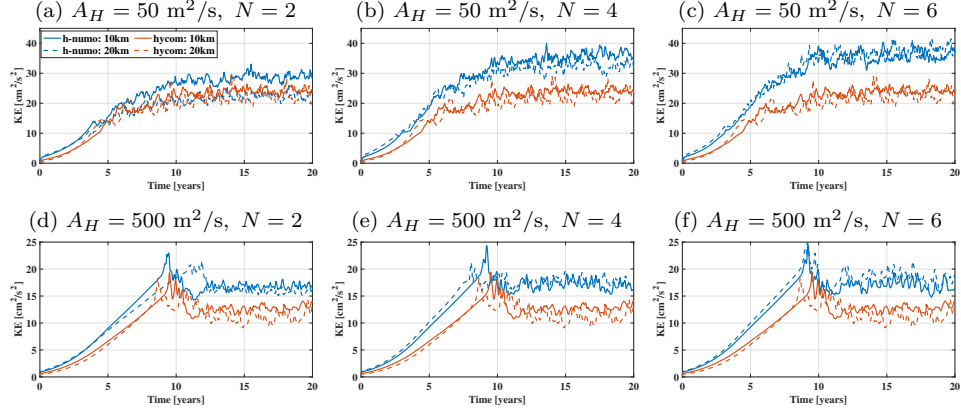


Figure 10: Total kinetic energy time history in the system obtained with h-NUMO and HYCOM models for 10 km and 20 km resolution, free-slip boundary condition, and 3 different polynomial order in h-NUMO. The energy in the top graphs is obtained with viscosity $A_H = 50 \text{ m}^2/\text{s}$, and the energy in the bottom graphs is obtained with viscosity $A_H = 500 \text{ m}^2/\text{s}$.

area (summed vertically over both layers), and it is computed as follows:

$$\text{KE} = \sum_{k=1}^{N_t} \frac{\int_{\Omega} \frac{1}{2} (u_k^2 + v_k^2) \Delta p_k d\Omega}{\int_{\Omega} \Delta p_k d\Omega}. \quad (40)$$

In h-NUMO, as the polynomial order increases, the number of elements is adjusted to obtain the same resolution. In Fig. 10 (a), both codes reach the same energy equilibrium for 20 km resolution, but h-NUMO retained higher energy for 10 km resolution. For polynomial order 4 and 6, Fig. 10 (b)-(c) show that h-NUMO has higher energy compared to HYCOM, and its energy increases as polynomial order increases. Fig. 10 (d)-(f) shows with viscosity $A_H = 500 \text{ m}^2/\text{s}$, the KE is noticeably higher with all the polynomial order and resolutions than HYCOM.

5.3.2. Simulations with no-slip boundary condition

We repeated the test in Section 5.3.1 with no-slip boundary conditions where all other aspects of the models remained the same, including the time steps. The time evolution of the sea-surface height at different times for 20 km and 10 km resolution are shown in Fig.11. The jet in the western boundary is less active in h-NUMO than in HYCOM with 20 km resolution. For viscosity $A_H = 50 \text{ m}^2/\text{s}$, we observe the development of eddies as in the case of the

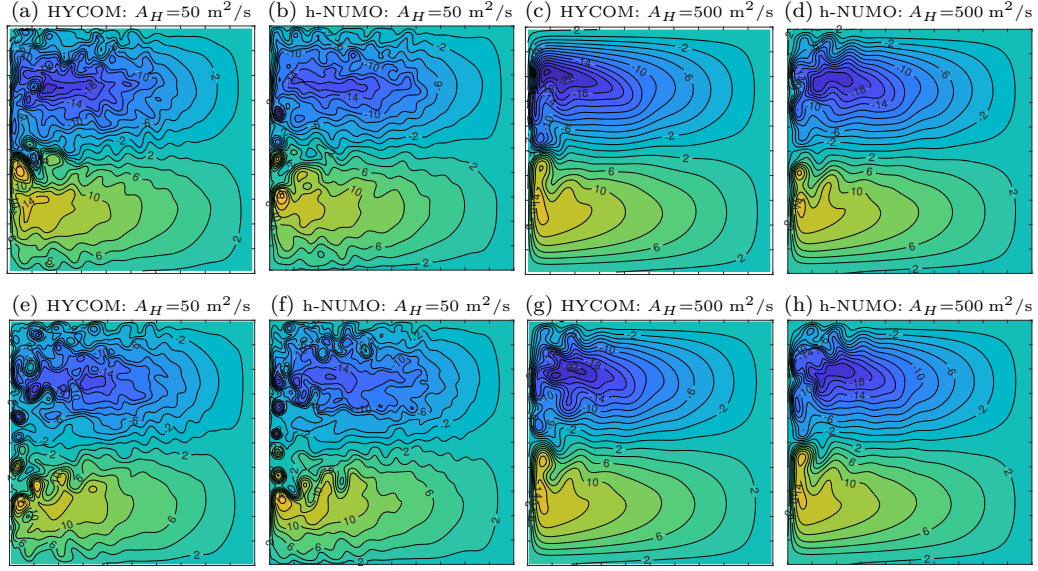


Figure 11: Snapshots of the sea-surface height at 20 years with h-NUMO and HYCOM using the no-slip boundary condition and viscosity $A_H = 50 \text{ m}^2/\text{s}$ and $A_H = 500 \text{ m}^2/\text{s}$. The results in top panels (a)-(d) are obtained with 20 km resolution, while the results in bottom panels (e)-(h) are obtained with 10 km resolution. The contour interval is 2 centimeters, with lower elevations in the northern region and higher elevations in the southern region.

free-slip condition, and for viscosity $A_H = 500 \text{ m}^2/\text{s}$, the western layer is thicker with less intense flows along the southern and northern boundary, especially for HYCOM, where at 20 years, we still do not see any eddies in the northern boundary. At 10 km resolution Fig.11 (e)-(f), the general flow circulation is similar in both h-NUMO and HYCOM with similar western boundary thickness.

Fig.12 shows a comparison of the sea surface height variability for the last 5 (15-20) years of the model run at resolutions 20 km and 10 km, with the no-slip boundary condition. At 20 km resolution, in panels (a)-(b) of Fig. 12, where the viscosity is $50 \text{ m}^2/\text{s}$, HYCOM presents a higher SSH variability in the middle of the western boundary than h-NUMO. However, the variability is higher along the western boundary in h-NUMO. At a higher viscosity $500 \text{ m}^2/\text{s}$ in panels (c)-(d), the SSH variability patterns remain similar but with slightly higher activity toward the west-northern boundary in h-NUMO compared to HYCOM. Fig. 12 (e)-(f) shows that at 10 km

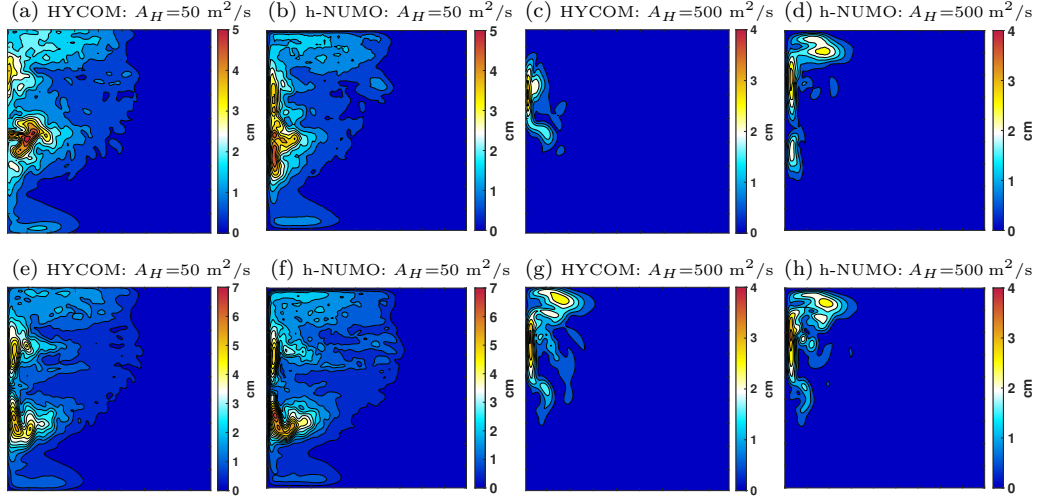


Figure 12: Comparison of sea surface height variability over the last five years (15-20) for the viscosity value $A_H = 50 \text{ m}^2/\text{s}$, $A_H = 500 \text{ m}^2/\text{s}$ and the no-slip boundary condition. The results in top panels (a)-(d) are obtained with 20 km resolution, while the results in bottom panels (e)-(h) are obtained with 10 km resolution. The contour line is after every 5 millimeters.

resolution and $A_H = 50 \text{ m}^2/\text{s}$, h-NUMO has a higher SSH variability. At $A_H = 500 \text{ m}^2/\text{s}$, both models' SSH variability is almost identical. From the results at $A_H = 500 \text{ m}^2/\text{s}$, we can conclude that both models are completely resolving all motions admitted by the viscosity.

Fig. 13 shows plots of the total kinetic energy in the system as a function of time. As compared to the free-slip boundary conditions in Fig. 10, the plots confirm that both models almost converge to the same kinetic energy for viscosity $A_H = 500 \text{ m}^2/\text{s}$. With this viscosity, there is no increase in the energy as the polynomial order increases in h-NUMO, indicating that all the necessary features have been resolved. Thus, with $A_H = 500 \text{ m}^2/\text{s}$, a 4th-order polynomial order is enough to resolve all scales of motion since we do not see a notable increase in the kinetic energy with the 6th-order polynomial. Similarly, for viscosity $A_H = 50 \text{ m}^2/\text{s}$, we observe a small difference in the kinetic energy in h-NUMO between the 4th-order and 6th-order polynomials results.

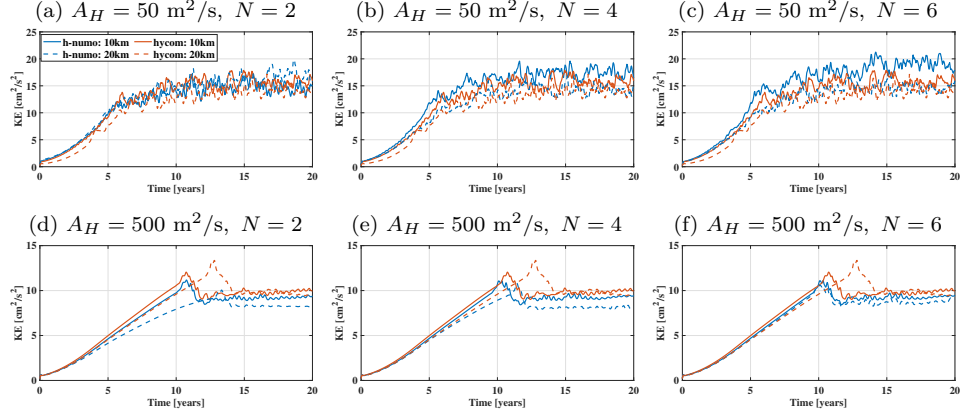


Figure 13: Total kinetic energy time history in the system obtained with h-NUMO and HYCOM models for 10 km and 20 km resolution, no-slip boundary condition, and 3 different polynomial order in h-NUMO. The energy in the top graphs is obtained with viscosity $A_H = 50 \text{ m}^2/\text{s}$, and the energy in the bottom graphs are obtained with viscosity $A_H = 500 \text{ m}^2/\text{s}$.

5.4. Eddy kinetic energy and time-averaged kinetic energy

Given the differences in quasi-steady total kinetic energy levels between different h-NUMO polynomial orders and HYCOM, in Fig. 14, we explore the relation between the mean kinetic energy (MKE) averaged over the years 15-20 and the resolution and the number of degrees of freedom (DOF). We run the simulations for h-NUMO with different polynomial order N . For $A_H = 50 \text{ m}^2/\text{s}$ and the free-slip boundary condition (panel a), MKE generally increases with the decrease of the average grid spacing Δx . We observe, however, that for the highest resolutions ($\Delta x = 10 \text{ km}$ to 5 km), the difference between the 4th and 6th-order polynomial MKE is very small, and the level of MKE is steady, suggesting that the simulation is fully resolved. The 2nd-order h-NUMO simulation and HYCOM result are approaching that level, but at 5 km , both are still underresolved. For $A_H = 500 \text{ m}^2/\text{s}$ (panel a, bottom set of lines), the values of MKE are almost constant for $\Delta x \leq 15 \text{ km}$ for all polynomial orders in h-NUMO, with HYCOM approaching that level at $\Delta x = 4 \text{ km}$. We plot the same result as a function of degrees of freedom in panel (c), as in h-NUMO we only use pressure and two momentum variables (but the DG grid introduces duplicate points along the element boundaries), while in HYCOM, we use a C-grid with pressure, two velocity variables and vorticity. This adjustment most significantly affects the $N = 2$ result, which

is shifted closer to the HYCOM reference in panel (c).

For the no-slip boundary condition (panel b,d), the 6th order h-NUMO result for the lower viscosity setting appears to be converged for $\Delta x \leq 15$ km, while all the other models reach the same value of MKE only for the highest resolution. Even though the simulations are stable, for $N = 2$ we observe oscillations at element boundaries due to not enough stabilization leading to grid imprinting for resolutions $\Delta x > 15$ km. This will trigger unphysical increased kinetic energy at cell interfaces, as observed for $\Delta = 12$ km. Thus, we remove the time-averaged kinetic energy for $N = 2$ and $\Delta x > 12$ km from the comparison. The results for higher viscosity setting (bottom set of lines) are fairly constant for the entire range of resolutions, with HYCOM showing a consistently higher level than all h-NUMO results. This is in opposition to the corresponding result in the free-slip simulation, where the HYCOM MKE was lower than h-NUMO. This discrepancy may be due to differences in the implementation of the no-slip condition in both models.

To understand the spatial distribution of energy, in Fig. 15 we plot the time-average (over the years 15-20) of eddy kinetic energy (EKE) for the three polynomial orders and HYCOM at 10 km resolution with the free-slip boundary condition. The EKE pattern is similar in all the simulations with significantly higher magnitudes for polynomial orders $N = 4$ and $N = 6$. Similarly, in Fig. 16 EKE for the no-slip boundary condition simulation has a similar pattern across all simulations, with h-NUMO having a higher magnitude than HYCOM. These results are consistent with the results in Fig. 14 and support the hypothesis that the DG method with high-order polynomials does not dissipate as much energy as lower-order DG and HYCOM.

5.5. Unstructured mesh

Unstructured grids have become an important feature of modern ocean models, as they allow for better alignment of computational resources with important regions in the modeled domain. Unstructured grids can more accurately capture complex geometrical features such as steep bathymetry and coastlines. The numerical methods in GNUME are capable of supporting arbitrary quadrilateral unstructured meshes [57] both as conforming and non-conforming, possibly dynamically adaptive grids [58].

To demonstrate h-NUMO's ability to use unstructured meshes, we have designed a grid using GMSH [53] software. The grid has quadrilateral elements of side length 83 km at the western boundary and 250 km at the eastern

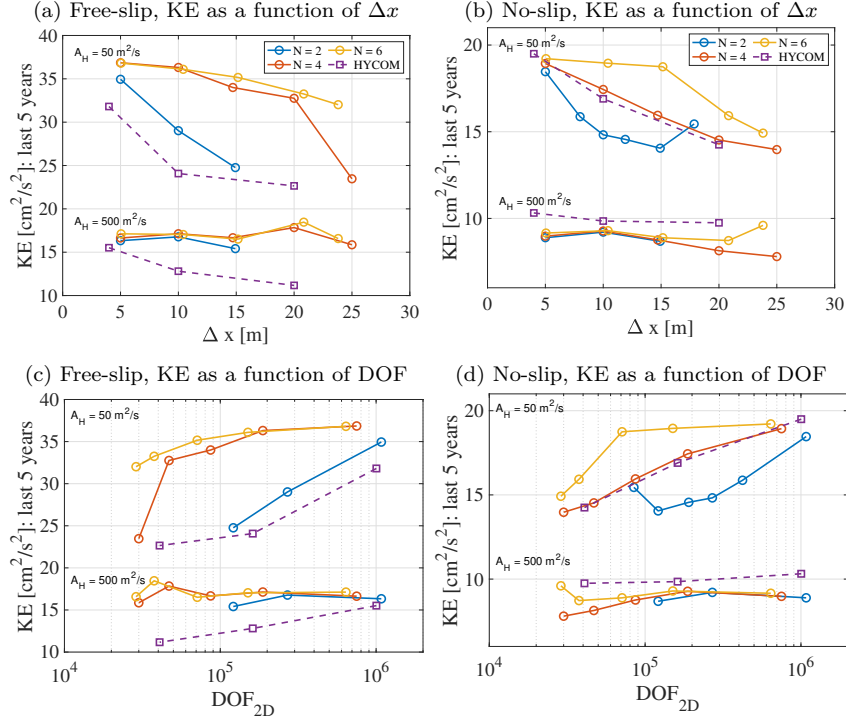


Figure 14: Time-averaged kinetic energy as a function of resolution (panels a,b) and degrees of freedom in 2D grid (DOF_{2D} , panel c,d) for free-slip (a,c) and no-slip (b,d) boundary conditions and two different settings of viscosity A_H . The DOF_{2D} are calculated for a single layer only, not accounting for the number of layers.

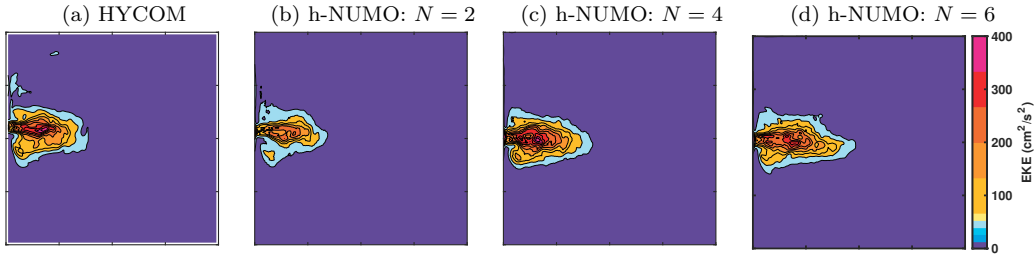


Figure 15: Comparison of h-NUMO and HYCOM top layer mean eddy kinetic energy per unit mass fields calculated over five years from the model simulations at 10 km grid resolution. The viscosity value $A_H = 50 \text{ m}^2/\text{s}$ and the free-slip boundary conditions are used. The contour interval is $40 \text{ cm}^2/\text{s}^2$.

boundary (Fig.17 (a)). We used 4th-order polynomials with h-NUMO for a

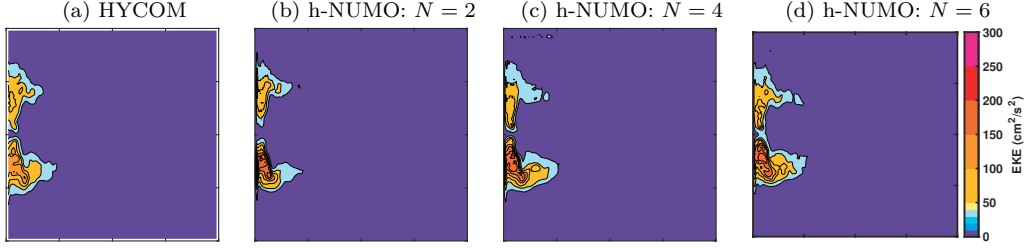


Figure 16: Comparison of h-NUMO and HYCOM top layer mean eddy kinetic energy per unit mass fields calculated over five years from the model simulations at 10 km grid resolution. The viscosity value $A_H = 50 \text{ m}^2/\text{s}$ and the no-slip boundary conditions are used. The contour interval is $30 \text{ cm}^2/\text{s}^2$.

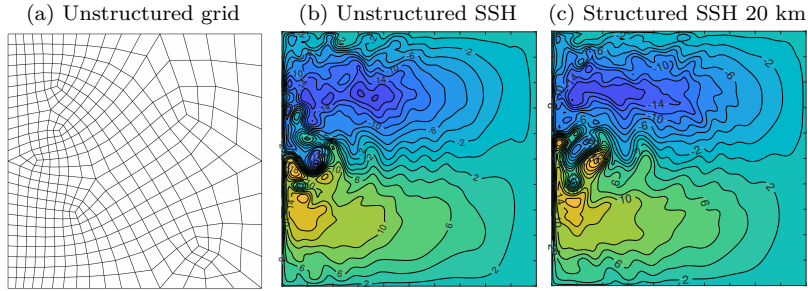


Figure 17: Unstructured grid (a) and the sea surface height contour plots for unstructured (b) and uniform 20 km (c) grids simulated at 20 years with viscosity $A_H = 50 \text{ m}^2/\text{s}$, 4th order polynomials with h-NUMO. The resolution of the unstructured grid varies from 20 km at the western boundary to 62.5 km at the eastern boundary.

resolution varying from 20 km to 60 km. We specified the resolutions based on the knowledge that there are more dynamic features close to the western boundary (see Fig. 8, 9) and used the 20km scale to match the previous results. This grid could be further optimized with specific information regarding the flow, i.e., the sea surface height in Fig. 9 or other measures like eddy kinetic energy, etc. The results presented below are meant to show the technical capability of h-NUMO and are not an exhaustive demonstration of the unstructured grid performance.

Fig. 17 panels (b) and (c) compare the instantaneous sea surface height at 20 years between the unstructured and the uniform 20 km grids. We note similar features between the two results, with the structured grid exhibiting more dynamic structures in the eastern part of the boundary. We attribute this discrepancy to significantly less resolution in this area.

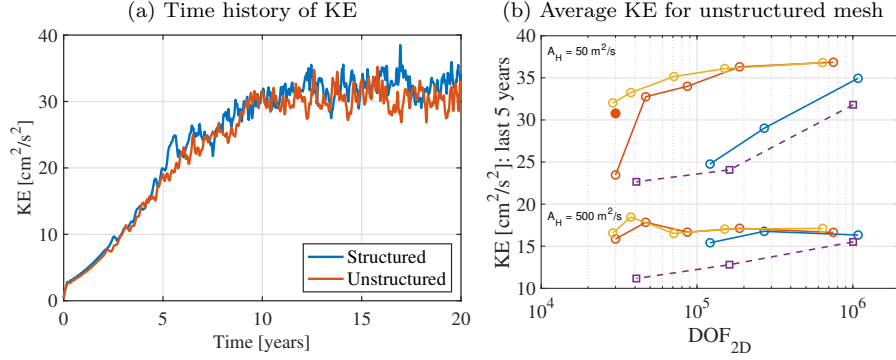


Figure 18: Time history of kinetic energy in h-NUMO with structured 20 km grid and an unstructured grid (a) and the comparison of time-averaged kinetic energy in unstructured mesh simulation (red filled circle in panel b) with convergence study performed on structured meshes. The unstructured simulation used 4th order polynomials and viscosity $A_H = 50 \text{ m}^2/\text{s}$.

Instantaneous results suggest smaller kinetic energy in the unstructured simulation. Fig. 18 (a) shows the energy comparison between the uniform mesh of 20 km resolution and the unstructured mesh results. The KE for the unstructured mesh ($N_e = 400$ elements) behaves very similarly to the 20 km uniform mesh ($N_e = 625$ elements), with only slightly lower quasi-steady state KE, which is consistent with the result in Fig. 17. The time-averaged kinetic energy over the last 5 years for the unstructured mesh simulation is $30.59 \text{ cm}^2/\text{s}^2$, which is less compared to the time-averaged energy of structure mesh ($32.75 \text{ cm}^2/\text{s}^2$). This is also visible in panel (b), where the averaged kinetic energy for the 20 km uniform simulation is slightly higher than the unstructured result, marked by a filled red circle. We compare the unstructured result in terms of the number of degrees of freedom, as the resolution is changing throughout the mesh. The unstructured mesh significantly improves the resolved KE over the same DOF simulation with the structured grid. The unstructured $N = 4$ result is close to the energy obtained with 6th-order polynomials with comparable uniform resolution. This result highlights significant potential for unstructured meshes combined with high-order DG methods in ocean modeling. We plan to explore unstructured meshes in future work, particularly their applications to coastline simulations.

	$\Delta t_{bcl}[s]$	$\Delta t_{btp}[s]$	N_e	2D grid points
HYCOM 4km	112.5	5.625		1 002 001
h-NUMO 4km $N = 2$	100	5	62 500	1 687 500
h-NUMO 4km $N = 4$	100	5	15 625	1 171 905
h-NUMO 4km $N = 6$	100	5	6889	1 012 683
h-NUMO 8km $N = 2$	200	10	15 625	421 875
h-NUMO 20km $N = 4$	600	30	625	46 875
h-NUMO 25km $N = 6$	400	25	169	24 843

Table 1: Baroclinic and barotropic time steps (Δt_{bcl} and Δt_{btp} , respectively), number of h-NUMO elements (N_e) and the number of grid points on a 2D mesh used in performance tests.

5.6. Parallel performance

We compared the performance of h-NUMO and HYCOM on a double-gyre test with 40 layers to mimic the computational effort required of a realistic model. The first two layers followed the specifications outlined in Section 5.3, and we set up the subsequent layers to have the same density as the second layer and a thickness of 0.001 m. In both models, the cost of each layer is the same regardless of thickness, so we are mimicking 40 non-vanishing layers. We used viscosity $A_H = 50 \text{ m}^2/\text{s}$. Each model used 20 barotropic sub-steps in the splitting scheme and an appropriate size of the barotropic time step, guaranteeing model stability. We have designed two experiments: I) the equal resolution experiment, where HYCOM and h-NUMO with different polynomial order settings all had comparable resolution of 4km, and II) the equal energy experiment, where h-NUMO resolutions for different polynomial orders were chosen to match the kinetic energy resolved by 4 km HYCOM simulation. To decide on the h-NUMO resolutions, we used Fig. 14 (a), and chose 8 km resolution for h-NUMO $N = 2$, 20 km resolution for $N = 4$ and 25 km for $N = 6$. Table 1 summarizes the time-step choices and the number of 2D grid points in each experiment. Note that we report the points in a 2D grid, which is the number of grid points used by the barotropic solver, and the baroclinic solver uses that many points in each layer. This number in the baroclinic solver must be multiplied by 40 to get the total number of DOFs.

For each experiment, we have conducted a strong-scaling test for up to 384 MPI ranks on Boise State University’s high-performance computing cluster Borah. Figure 19 shows the speed-up (panel a), parallel efficiency (panel

b), time-to-solution of one model-day simulation (panel c), and the CPU-hours used for a four model-day simulation (panel d) as a function of the number of MPI ranks used (nproc) for the equal resolution experiment. We used 16 cores as a reference for this study, as 4km problem is large and was taking a very long time on low processor counts. We see that h-NUMO simulations scale significantly better, reaching 80-90% parallel efficiency at the largest number of processors used. HYCOM's performance drops to about 40 % at 256 cores. Despite better parallel efficiency, h-NUMO is 8x to 17x slower than HYCOM for the same resolution, even at maximum processor counts. However, with parallel efficiency maintained at about 80% (see Fig. 20), h-NUMO will likely close this gap at higher processor counts. It is also worth noticing that the resolution definition we use (average distance between nodal points) results in a much higher number of grid points for h-NUMO simulations, a significant factor in the overall cost. Figure 19 (c) shows the performance of the model with polynomials of order 2, 4, and 6 at the same resolution (4 km), meaning we adjusted the number of elements corresponding to each polynomial to have approximately the same DoFs. We observed that high-order polynomials exhibit a slower time to solution compared to low-order ones. This is due to the $(2N + 1)^2$ quadrature points used in the integrations per element, resulting in 25, 81, and 169 quadrature points per element for polynomial orders 2, 4, and 6, respectively.

We have repeated the strong-scaling test for the equivalent energy experiment. Note that in the lower-resolution simulations, the parallel domain decomposition is limited by how many elements are in the grid. Table 2 summarizes the number of elements and 2D grid points per core at the highest processor count used. The higher-order h-NUMO simulations use only one ($N = 6$) or two ($N = 4$) elements and about 150 DOF per core (considering only 2D DOF, baroclinic solver uses 40 layers, so 40x that number of DOF for both h-NUMO and HYCOM) at the largest machine size. Figure 20 (b) shows that even in this extreme situation, the model maintains 75-80% efficiency (panel b). This implies that this is the lower bound of the parallel efficiency for h-NUMO in its current form, as the communication volume to computation ratio is highest for one element per core case.

Even for lower resolutions but with the same resolved kinetic energy level, h-NUMO scales better than HYCOM (Fig. 20 (a,b)), and the higher-order simulations ($N = 4, 6$) achieve 7x to 16x faster time-to-solution (panel c) and consistently lower CPU-hour cost (panel d) throughout the tested processor count range. Due to better parallel efficiency at higher processor counts, the

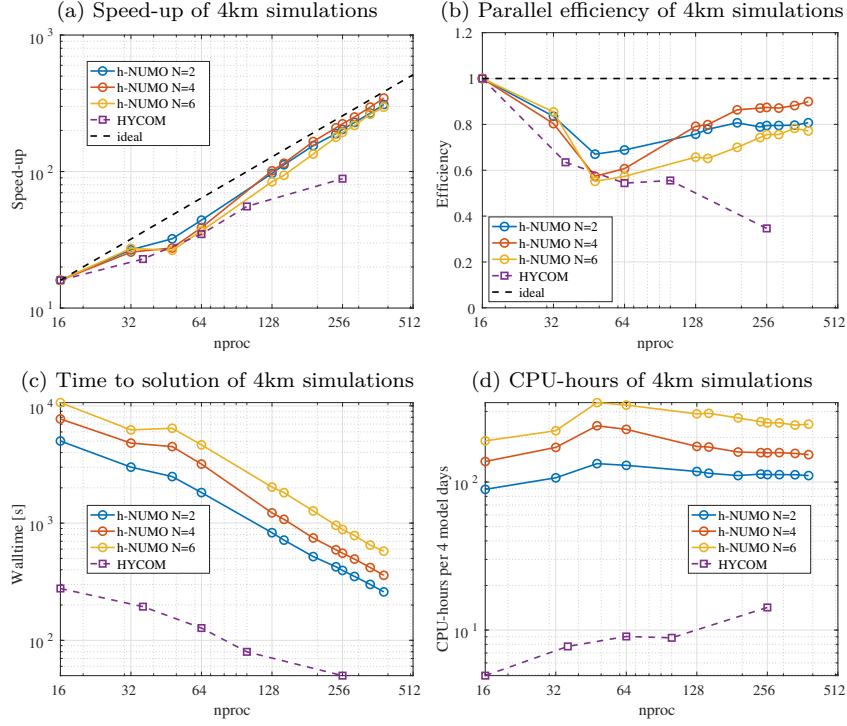


Figure 19: Strong-scaling experiment of the 4km resolution double-gyre test using HYCOM reference and different polynomial orders in h-NUMO. The results are compared in terms of (a) speed-up, (b) parallel efficiency, (c) time-to-solution for 1 model-day simulation, and (d) CPU-hours cost of 4 model-day simulations. For speed-up and performance plots, we used timing at 16 cores as reference.

	4 km resolution		energy-equivalent resolutions	
	elements per core	2D DOF per core	elements per core	2D DOF per core
HYCOM	-	3914	-	3914
h-NUMO $N = 2$	163	4401	41	1107
h-NUMO $N = 4$	41	3075	2	150
h-NUMO $N = 6$	19	2793	1	147

Table 2: Number of elements and 2D DOF per core at number of parallel processes used. All 4km simulations and the 8km h-NUMO used a maximum of 384 cores, while 20 km h-NUMO used a maximum of 313, 25 km h-NUMO used 169, and 4 km HYCOM used 265 cores.

computational cost gap between h-NUMO and HYCOM widens to about an order of magnitude. At 256 cores, $N = 2$ simulation achieves the same time-

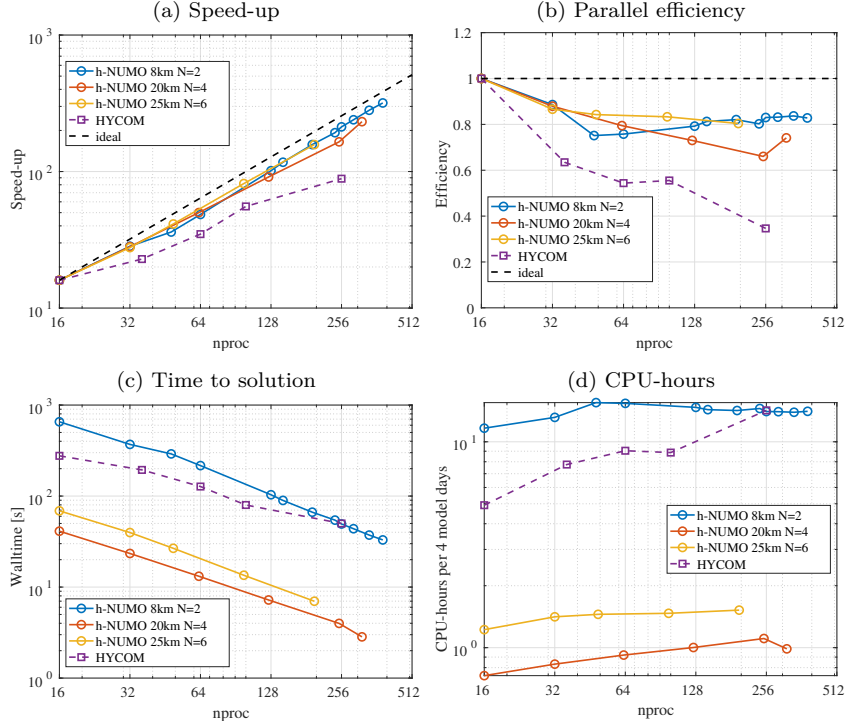


Figure 20: Strong-scaling experiment of the double-gyre test using HYCOM 4 km resolution reference and equal-energy simulations with different polynomial orders in h-NUMO (8 km with $N = 2$, 20 km with $N = 4$ and 25 km with $N = 6$). The results are compared in terms of (a) speed-up, (b) parallel efficiency, (c) time-to-solution for 1 model day simulation, and (d) CPU-hours cost of 4 model-day simulation. For speed-up and performance plots we used timing at 16 cores as reference.

to-solution and computational cost as HYCOM and, with more resources, beats the reference code.

This performance comparison features the h-NUMO with exact integration of the integrals in the weak form equations (A.2)-(A.4), using $(2N + 1)^2$ quadrature points per element. Our preliminary tests with inexact integration (using only $(N + 1)^2$ quadrature points per element) show that a serial performance improvement of up to 5x is possible if we use this approach in barotropic and baroclinic solvers. At the same time, we expect that this change will have little effect in a parallel setting, as by lowering the computational intensity due to the inexact integration, we will also reduce the communication volume. This will likely, in fact, further improve the parallel

efficiency, as we were able to scale other applications built with GNUME and inexact integration to 99% parallel efficiency at up to 3 million cores Müller et al. [21].

5.7. Comparison between centered and upwind fluxes

We used central flux in the numerical schemes in this work, which worked well for all our tests and the numerical results reported in this paper. However, we have also implemented an upwind flux in the numerical schemes for verification purposes. In the long run, we intend to switch to the upwind flux in the model, as it can become an issue in certain applications, particularly in the following case.

In a more realistic case, one might have vanishing layers (zero thickness); suppose that in element 1, the layer thickness is zero ($\Delta z_k = 0$) which implies $\Delta p_k(\mathbf{x}, t) = g\rho_k\Delta z_k = 0$, and in an adjacent element 2, the thickness is positive ($\Delta z_k > 0$) which implies $\Delta p_k(\mathbf{x}, t) = g\rho_k\Delta z_k > 0$. At the edge between those elements, the average thickness is positive. However, if the fluid velocity calculated at the element edge points out of element 1, and if the average of $\Delta p_k(\mathbf{x}, t)$ is used in the expression $\mathbf{u}_k\Delta p_k$, then positive mass is transported out of an element with zero mass, and the layer thickness goes negative. Therefore, the use of central flux would be problematic in that situation. However, in all the test cases used in this paper, the layer thickness never goes to zero.

It is challenging to find a test case with an analytic solution for multilayer shallow water equations to do a convergence study with the model using central flux. However, we have added the results with the upwind flux for comparison. Fig. 21 shows that the total kinetic energy obtained based on the central and upwind fluxes are similar for the three polynomials used, confirming the accuracy of the central flux results. Fig. 22 shows the sea-surface height (SSH). The solutions from the two numerical fluxes are not point-wise identical due to the different numerical schemes used and the sensitivity of the non-linear equations to perturbations.

6. Conclusion

The main focus of this work is to develop the multilayer shallow water ocean model, h-NUMO, in the GNUMe framework using a high-order DG method and extend the one-dimensional splitting scheme developed in [11] to two horizontal dimensions for more realistic applications. The results in

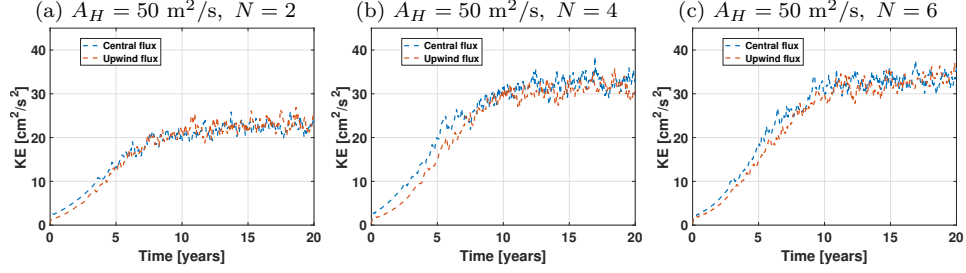


Figure 21: Total kinetic energy time history in the system obtained with h-NUMO using central and upwind fluxes and three different polynomial orders. We use 20 km resolution, free-slip boundary condition, and viscosity $A_H = 50 \text{ m}^2/\text{s}$.

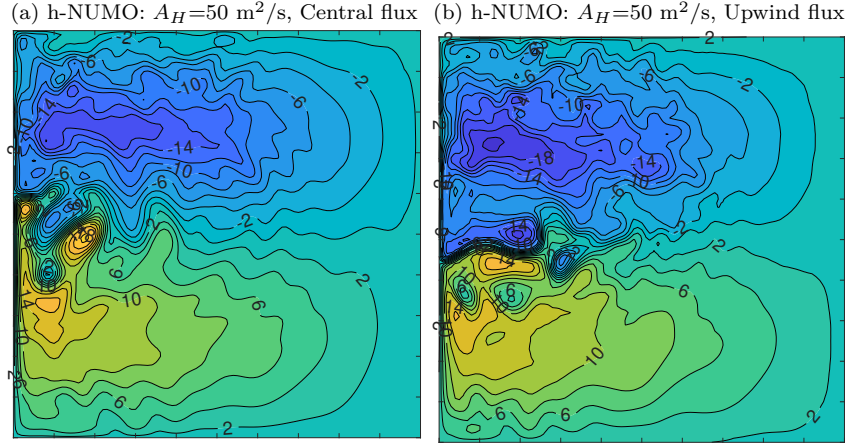


Figure 22: Snapshots of the sea-surface height at 20 years obtained with h-NUMO using central and upwind fluxes. The results are obtained with 20 km resolution, and viscosity $A_H = 50 \text{ m}^2/\text{s}$. The contour interval is 2 centimeters, with lower elevations in the northern region and higher elevations in the southern region.

Section 5 verify the accuracy of the numerical methods used in the h-NUMO model and validate it on a range of test cases, with an extensive numerical test on the double-gyre test. We demonstrate that the model preserves the constant free surface height for the lake-at-rest test for different numbers of layer and barotropic substeps N_{btp} . The well-balanced test indicates that the absolute errors do not significantly grow with time regardless of the number of layers, and the analysis of the velocity results reveals that there is no significant spurious numerical mixing; however, further verification is needed, such as analyzing the error for long-time simulation results. We

also verify the accuracy of the model by showing that it accurately predicts the wave propagation speeds in various settings. The comparisons with the HYCOM model show that the high-order DG method resolves more dynamic features using a similar number of degrees of freedom. The analysis of the mean kinetic energy reveals that all models converge to the same quasi-steady state, with high-order DG resolving all the features contributing to the flow energy with lower resolutions. When using the no-slip boundary condition, we observe that the energy produced by HYCOM using higher viscosity is higher than h-NUMO, which might be because we are not resolving the boundary layer at a lower resolution or the difference in the implementation of the no-slip boundary condition in both models. Overall, we conclude that h-NUMO reproduces ocean dynamics features and resolves the flow better as we increase the polynomial order.

From the preceding discussions, we observed that the DG method is well suited for resolving the ocean dynamical flow as one could achieve the same results in h-NUMO using a high-order polynomial with a much smaller number of elements observed with the unstructured mesh simulation. The Gauss-Lobatto-Legendre distribution of nodal points within elements provides optimal interpolation with Lagrangian polynomial, and the localized features can be more resolved. The high-order approximation of the solution within elements is very effective as the flow features were more resolved as we increased the polynomial order. The parallel scalability shows that the h-NUMO code scales well, and with one element per computer core for polynomial order $N = 6$, the code maintains 90% parallel efficiency. For the equal energy experiment, h-NUMO achieves an order of magnitude faster time-to-solution and cheaper CPU-hours cost than HYCOM, thus achieving a similarly resolved flow much faster and with significantly fewer resources.

Future work involves optimizing the code as we expect significant improvements due to improved communication algorithms, which will communicate the nodal point data instead of the quadrature point data, which is done now due to simpler implementation. We will conduct more in-depth parallel performance studies on larger problem sizes and compare them with HYCOM. There are still some important questions and tasks worthy of further study, including how the change in the number of substeps of the barotropic system affects the simulation results and the implementation of different numerical flux schemes such as Russanov flux. Some other issues to address are the vertical mixing between layers and wetting and drying schemes for negative layer depth.

Software and reproducibility statement

The source code for h-NUMO used to produce all the results in this work is available at <https://github.com/ygahounzo/h-NUMO>, and HYCOM source code is available at <https://github.com/HYCOM>.

Acknowledgments

The authors would like to thank Antone Chacartegui and James Stringham, master and undergraduate students at Boise State University, for their help in running some of the double-gyre simulations on the Boise State University HPC cluster. We are grateful to Prof. Frank X. Giraldo for his remarks on boundary conditions, Laplacian flux formulation, code implementation, and general feedback on the paper. We acknowledge support from the Office of Naval Research through Grant N00014-20-1-2038.

Appendix A. Spatial discretization

The weak form of the layer mass equation (1) is obtained over element Ω_e

$$\int_{\Omega_e} \psi_i \frac{\partial}{\partial t} (\Delta p_k) d\Omega_e + \int_{\Gamma_e} \psi_i \mathbf{n} \cdot (\mathbf{u}_k \Delta p_k)^\dagger d\Gamma_e - \int_{\Omega_e} (\mathbf{u}_k \Delta p_k) \cdot \nabla \psi_i d\Omega_e = 0, \quad (\text{A.1})$$

where Γ_e represents the boundary of the grid element Ω_e , $\mathbf{n} = (n_x, n_y)$ is an outward-facing normal vector to the element boundary, ψ_i is a test function, and the superscript † denotes numerical flux. We discuss the computation of the numerical flux terms in the subsection 3.1.

Taking $U_k(\mathbf{x}, t) = u_k \Delta p_k$ and $V_k(\mathbf{x}, t) = v_k \Delta p_k$, the weak form of the u -component on the layer momentum equation (2) without the viscosity term is

$$\begin{aligned} \int_{\Omega_e} \psi_i \left[\frac{\partial U_k}{\partial t} - f V_k \right] d\Omega_e = & - \int_{\Gamma_e} \psi_i \mathbf{n} \cdot (\mathbf{u}_k U_k)^\dagger d\Gamma_e + \int_{\Omega_e} \mathbf{u}_k U_k \cdot \nabla \psi_i d\Omega_e \\ & - \int_{\Gamma_e} \psi_i n_x H_k^\dagger d\Gamma_e + \int_{\Omega_e} H_k \frac{\partial \psi_i}{\partial x} d\Omega_e \\ & + g \int_{\Omega_e} \psi_i \left(p_{k-1} \frac{\partial z_{k-1}}{\partial x} - p_k \frac{\partial z_k}{\partial x} + \Delta \tau_k^x \right) d\Omega_e, \quad (\text{A.2}) \end{aligned}$$

where $H_k^\dagger$ is the value of H_k at the element edge. Similarly, we compute the v -component of the momentum is

$$\begin{aligned} \int_{\Omega_e} \psi_i \left[\frac{\partial V_k}{\partial t} + f U_k \right] d\Omega_e = & - \int_{\Gamma_e} \psi_i \mathbf{n} \cdot (\mathbf{u}_k V_k)^\dagger d\Gamma_e + \int_{\Omega_e} \mathbf{u}_k V_k \cdot \nabla \psi_i d\Omega_e \\ & - \int_{\Gamma_e} \psi_i n_y H_k^\dagger d\Gamma_e + \int_{\Omega_e} H_k \frac{\partial \psi_i}{\partial y} d\Omega_e \\ & + g \int_{\Omega_e} \psi_i \left(p_{k-1} \frac{\partial z_{k-1}}{\partial y} - p_k \frac{\partial z_k}{\partial y} + \Delta \tau_k^y \right) d\Omega_e. \quad (\text{A.3}) \end{aligned}$$

$\Delta \tau_k^x$ and $\Delta \tau_k^y$ are scalar in Eq. (A.2) and Eq. (A.3) because those equations are not in vector form.

We also obtain the weak form of the barotropic mass equation (13) as

$$\int_{\Omega_e} \psi_i \frac{\partial}{\partial t} (p'_b \eta) d\Omega_e + \int_{\Gamma_e} \psi_i \mathbf{n} \cdot (p_b \mathbf{u})^\dagger d\Gamma_e - \int_{\Omega_e} (p_b \mathbf{u}_b) \cdot \nabla \psi_i d\Omega_e = 0. \quad (\text{A.4})$$

The weak forms of the barotropic momentum equations (8) are derived similarly as in Eq. (A.2)-(A.3).

To obtain a DG discretization, within each element Ω_e , we expand the solution vector $\mathbf{q}(\mathbf{x}, t)$ inside the element Ω_e :

$$\mathbf{q}_k(\mathbf{x}, t) \approx \mathbf{q}_{k,N}^{(e)}(\mathbf{x}, t) = \sum_{j=1}^{M_N} \psi_j(\mathbf{x}) \mathbf{q}_{k,j}^{(e)}(t), \quad (\text{A.5})$$

where $\mathbf{q}_{k,j}^{(e)} = [\Delta p_{k,j}^e, U_{k,j}^e, V_{k,j}^e]^T$ is the expansion coefficient corresponding to node j with $\mathbf{q}_k(\mathbf{x}, t) = [\Delta p_k, U_k, V_k]^T$. We derive the DG discretization in the weak form of the layer mass equation (A.1) which can be easily adapted to the remaining equations. Introducing the expansion of Δp_k from Eq. (A.5) into (A.1), we obtain

$$\begin{aligned} \int_{\Omega_e} \psi_i \frac{d}{dt} \left(\sum_{j=1}^{M_N} \psi_j \Delta p_{k,j}^{(e)} \right) d\Omega_e = & - \int_{\Gamma_e} \mathbf{n} \cdot \psi_i (\mathbf{u}_k \Delta p_k)^{(e,\dagger)} d\Gamma_e \\ & - \int_{\Omega_e} \nabla \psi_i \cdot (\mathbf{u}_k \Delta p_k)^{(e)} d\Omega_e. \quad (\text{A.6}) \end{aligned}$$

Rearranging the summation, derivative, and integral gives

$$\sum_{j=1}^{M_N} \int_{\Omega_e} \psi_i \psi_j d\Omega_e \frac{d\Delta p_{k,j}^{(e)}}{dt} = - \int_{\Gamma_e} \mathbf{n} \cdot \psi_i (\mathbf{u}_k \Delta p_k)^{(e,\dagger)} d\Gamma_e - \int_{\Omega_e} \nabla \psi_i \cdot (\mathbf{u}_k \Delta p_k)^{(e)} d\Omega_e, \quad (\text{A.7})$$

which can be represented in matrix form as

$$\mathbf{M}^{(e)} \frac{d\Delta \mathbf{p}_k^{(e)}}{dt} = -\mathbf{f}_k^{(e)} + \mathbf{a}_k^{(e)}, \quad (\text{A.8})$$

where $\Delta \mathbf{p}_k^{(e)}$ is the solution vector of size M_N that contains $\Delta p_{k,j}^{(e)}$ within the element e , $\mathbf{M}^{(e)}$ is the mass matrix with its ij -th entry given by

$$M_{ij}^{(e)} = \int_{\Omega_e} \psi_i \psi_j d\Omega_e = \sum_{l=1}^{M_Q} w_l J^{(e)}(\boldsymbol{\xi}_l) \psi_i(\boldsymbol{\xi}_l) \psi_j(\boldsymbol{\xi}_l), \quad (\text{A.9})$$

where $i, j = 1, \dots, M$, $e = 1, \dots, N_e$, w_l and $J^{(e)}$ are the quadrature weights and determinant of the Jacobian evaluated at the quadrature point $\boldsymbol{\xi}_l$, and $M_Q = m_q^2$ is the number of quadrature points within an element, where $m_q = 2N + 1$. The only difference between M_N and M_Q is that M_N is used in the polynomial approximation or basis expansion of the prognostic variables, while M_Q is the number of quadrature points used for integration. The quantity $\mathbf{f}_k^{(e)}$ in Eq. (A.8) is a vector resulting from the integration of the first term on the right-hand side of Eq. (A.7), with its entries defined as

$$f_{k,j}^{(e)} = \int_{\Gamma_e} \mathbf{n} \cdot \psi_j (\mathbf{u}_k \Delta p_k)^{(e,\dagger)} d\Gamma_e = \sum_{l=1}^{m_Q} w_l^{(f)} J^{(e)}(\boldsymbol{\xi}_l) \psi_{jl} \mathbf{n} \cdot (\mathbf{u}_k(\boldsymbol{\xi}_l) \Delta p_k(\boldsymbol{\xi}_l))^{(e,\dagger)} \quad (\text{A.10})$$

where the superscript $^{(f)}$ denotes the edge lying on Γ_e . The term $\mathbf{a}_k^{(e)}$ in Eq. (A.8) is a vector resulting from the evaluation of the second integral in the right-hand side of Eq. (A.7), with its entries given by

$$a_{k,j}^{(e)} = \int_{\Omega_e} \nabla \psi_j \cdot (\mathbf{u}_k \Delta p_k)^{(e)} d\Omega_e = \sum_{l=1}^{M_Q} w_l J^{(e)}(\boldsymbol{\xi}_l) \nabla \psi_j(\boldsymbol{\xi}_l) \cdot (\mathbf{u}_k(\boldsymbol{\xi}_l) \Delta p_k(\boldsymbol{\xi}_l))^{(e)}. \quad (\text{A.11})$$

The DG discretization Eq. (A.8) of the mass equation can be rewritten in a simple form as

$$\frac{d\Delta\mathbf{p}_k^{(e)}}{dt} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_k^{(e)}, \quad (\text{A.12})$$

where $\mathbf{r}_k^{(e)}, k = 1, \dots, N_l$ is obtained from the combination of the terms in the right-hand side of the equation (A.8).

The DG discretization of the weak form momentum equations (A.2)-(A.3) are obtained in a similar way as in the case of mass equation. The final form of the DG discretization of the momentum equations are given by

$$\frac{d\mathbf{U}_k^{(e)}}{dt} - f\mathbf{V}_k^{(e)} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_{U,k}^{(e)}, \quad (\text{A.13})$$

$$\frac{d\mathbf{V}_k^{(e)}}{dt} + f\mathbf{U}_k^{(e)} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_{V,k}^{(e)}, \quad (\text{A.14})$$

where $\mathbf{r}_{U,k}^{(e)}, k = 1, \dots, N_l$ is a vector of size M_N obtained from the spatial discretization of the right-hand side of the equation (A.2) and $\mathbf{r}_{V,k}^{(e)}, k = 1, \dots, N_l$ is a vector of size M_N obtained from the spatial discretization of the right-hand side of v -momentum equation (A.3) within the element e . The quantity $\mathbf{U}_k^{(e)}$ and $\mathbf{V}_k^{(e)}$ are the solution vectors of size M_N within the element e .

The DG discretizations of the barotropic equations are obtained in a similar way as described for the baroclinic equations in the proceedings

$$\frac{d(\boldsymbol{\eta}\mathbf{p}'_b)^{(e)}}{dt} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_b^{(e)}, \quad (\text{A.15})$$

$$\frac{d\mathbf{U}_b^{(e)}}{dt} - f\mathbf{V}_b^{(e)} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_{b,U}^{(e)}, \quad (\text{A.16})$$

$$\frac{d\mathbf{V}_b^{(e)}}{dt} + f\mathbf{U}_b^{(e)} = (\mathbf{M}^{(e)})^{-1} \mathbf{r}_{b,V}^{(e)}, \quad (\text{A.17})$$

where $(\boldsymbol{\eta}\mathbf{p}'_b)^{(e)}, \mathbf{U}_b^{(e)}$ and $\mathbf{V}_b^{(e)}$ are the solution vectors within the element e for the barotropic mass, u -momentum and v -momentum respectively. The quantities $\mathbf{r}_b^{(e)}, \mathbf{r}_{b,U}^{(e)}$ and $\mathbf{r}_{b,V}^{(e)}$ are vectors obtained from the spatial discretization of the right-hand side of the equations (13) and (8). The equation (A.15) is the DG discretization of the equation (13) while the equations (A.16) and (A.17) are the component-wise discretizations of the vector form of the momentum equation in (8).

References

- [1] S. M. Griffies, C. Böning, F. O. Bryan, E. P. Chassignet, R. Gerdes, H. Hasumi, A. Hirst, A.-M. Treguier, D. Webb, Developments in ocean climate modelling, *Ocean Modelling* 2 (2000) 123–192.
- [2] E. P. Chassignet, H. E. Hurlburt, O. M. Smedstad, G. R. Halliwell, A. J. Wallcraft, E. J. Metzger, B. O. Blanton, C. Lozano, D. B. Rao, P. J. Hogan, et al., Generalized vertical coordinates for eddy-resolving global and coastal ocean forecasts, *Oceanography* 19 (2006) 118.
- [3] R. Bleck, An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates, *Ocean modelling* 4 (2002) 55–88.
- [4] E. P. Chassignet, L. T. Smith, G. R. Halliwell, R. Bleck, North Atlantic simulations with the Hybrid Coordinate Ocean Model (HYCOM): Impact of the Vertical Coordinate Choice, Reference Pressure, and Thermobaricity, *Journal of Physical Oceanography* 33 (2003) 2504–2526.
- [5] E. Audusse, A multilayer Saint-Venant model: derivation and numerical validation, *Discrete and Continuous Dynamical Systems-B* 5 (2005) 189–214.
- [6] F. Bouchut, V. Zeitlin, A robust well-balanced scheme for multi-layer shallow water equations, *Discrete and Continuous Dynamical Systems-Series B* 13 (2010) 739–758.
- [7] N. Izem, M. Seaid, A well-balanced Runge-Kutta discontinuous Galerkin method for multilayer shallow water equations with non-flat bottom topography, *Advances in Applied Mathematics and Mechanics* 14 (2022).
- [8] Z. Zhang, H. Tang, J. Duan, High-order accurate well-balanced energy stable finite difference schemes for multi-layer shallow water equations on fixed and adaptive moving meshes, *Journal of Computational Physics* 517 (2024) 113301.
- [9] R. L. Higdon, A two-level time-stepping method for layered ocean circulation models: further development and testing, *Journal of Computational Physics* 206 (2005) 463–504.

- [10] R. Bleck, L. T. Smith, A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean: 1. Model development and supporting experiments, *Journal of Geophysical Research: Oceans* 95 (1990) 3273–3285.
- [11] R. L. Higdon, Multiple time scales and pressure forcing in discontinuous Galerkin approximations to layered ocean models, *Journal of Computational Physics* 295 (2015) 230–260.
- [12] H. Berntsen, Z. Kowalik, S. Sælid, K. Sørli, Efficient numerical simulation of ocean hydrodynamics by a splitting procedure, *Modeling, Identification and Control* 2 (1981) 181–199.
- [13] R. L. Higdon, A two-level time-stepping method for layered ocean circulation models, *Journal of Computational Physics* 177 (2002) 59–94.
- [14] P. D. Killworth, D. J. Webb, D. Stainforth, S. M. Paterson, The development of a free-surface Bryan–Cox–Semtner ocean model, *Journal of Physical Oceanography* 21 (1991) 1333–1348.
- [15] R. J. Spiteri, S. J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, *SIAM Journal on Numerical Analysis* 40 (2002) 469–491.
- [16] R. Lan, L. Ju, Z. Wang, M. Gunzburger, P. Jones, High-order multirate explicit time-stepping schemes for the baroclinic-barotropic split dynamics in primitive equations, *Journal of Computational Physics* 457 (2022) 111050.
- [17] J. A. Escobar-Vargas, P. J. Diamessis, F. X. Giraldo, High-order discontinuous element-based schemes for the inviscid shallow water equations: Spectral multidomain penalty and discontinuous Galerkin methods, *Applied Mathematics and Computation* 218 (2012) 4825–4848.
- [18] M. Ainsworth, Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods, *Journal of Computational Physics* 198 (2004) 106–130.
- [19] R. Gandham, High performance high-order numerical methods: applications in ocean modeling, Ph.D. thesis, Rice University, 2015. doi:10.13140/RG.2.1.1856.6641.

- [20] P. Fischer, D. Gottlieb, On the optimal number of subdomains for hyperbolic problems on parallel computers, *The International Journal of Supercomputer Applications and High Performance Computing* 11 (1997) 65–76.
- [21] A. Müller, M. A. Kopera, S. Marras, L. C. Wilcox, T. Isaac, F. X. Giraldo, Strong scaling for numerical weather prediction at petascale with the atmospheric model NUMA, *The International Journal of High Performance Computing Applications* 33 (2019) 411–426.
- [22] F. Giraldo, J. Hesthaven, T. Warburton, Nodal High-Order Discontinuous Galerkin Methods for the Spherical Shallow Water Equations, *Journal of Computational Physics* 181 (2002) 499–525. URL: <https://www.sciencedirect.com/science/article/pii/S0021999102971391>. doi:<https://doi.org/10.1006/jcph.2002.7139>.
- [23] C. Dawson, V. Aizinger, A discontinuous Galerkin method for three-dimensional shallow water equations, *Journal of Scientific Computing* 22 (2005) 245–267.
- [24] V. Aizinger, C. Dawson, The local discontinuous Galerkin method for three-dimensional shallow water flow, *Computer Methods in Applied Mechanics and Engineering* 196 (2007) 734–746.
- [25] E. J. Kubatko, J. J. Westerink, C. Dawson, hp discontinuous Galerkin methods for advection dominated problems in shallow water flow, *Computer Methods in Applied Mechanics and Engineering* 196 (2006) 437–451.
- [26] D. Wirasaet, E. Kubatko, C. Michoski, S. Tanaka, J. Westerink, C. Dawson, Discontinuous Galerkin methods with nodal and hybrid modal/nodal triangular, quadrilateral, and polygonal elements for non-linear shallow water flow, *Computer Methods in Applied Mechanics and Engineering* 270 (2014) 113–149.
- [27] S. Faghih-Naini, S. Kuckuk, V. Aizinger, D. Zint, R. Grosso, H. Köstler, Quadrature-free discontinuous Galerkin method with code generation features for shallow water equations on automatically generated block-structured meshes, *Advances in Water Resources* 138 (2020) 103552.

- [28] S. Blaise, A. St-Cyr, D. Mavriplis, B. Lockwood, Discontinuous Galerkin unsteady discrete adjoint method for real-time efficient tsunami simulations, *Journal of Computational Physics* 232 (2013) 416–430.
- [29] B. Bonev, J. S. Hesthaven, F. X. Giraldo, M. A. Kopera, Discontinuous Galerkin scheme for the spherical shallow water equations with applications to tsunami modeling and prediction, *Journal of Computational Physics* 362 (2018) 425–448.
- [30] L. Arpaia, M. Ricchiuto, A. G. Filippini, R. Pedreros, An efficient covariant frame for the spherical shallow water equations: Well balanced dg approximation and application to tsunami and storm surge, *Ocean Modelling* 169 (2022) 101915.
- [31] C. Dawson, C. J. Trahan, E. J. Kubatko, J. J. Westerink, A parallel local timestepping Runge–Kutta discontinuous Galerkin method with applications to coastal ocean modeling, *Computer Methods in Applied Mechanics and Engineering* 259 (2013) 154–165.
- [32] N. Beisiegel, S. Vater, J. Behrens, F. Dias, An adaptive discontinuous Galerkin method for the simulation of hurricane storm surge, *Ocean Dynamics* 70 (2020) 641–666.
- [33] R. Ford, C. Pain, M. Piggott, A. Goddard, C. De Oliveira, A. Umpleby, A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part i: Model formulation, *Monthly Weather Review* 132 (2004) 2816–2831.
- [34] S. Blaise, R. Comblen, V. Legat, J.-F. Remacle, E. Deleersnijder, J. Lambrechts, A discontinuous finite element baroclinic marine model on unstructured prismatic meshes: Part I: space discretization, *Ocean Dynamics* 60 (2010) 1371–1393.
- [35] T. Kärnä, S. C. Kramer, L. Mitchell, D. A. Ham, M. D. Piggott, A. M. Baptista, Thetis coastal ocean model: discontinuous Galerkin discretization for the three-dimensional hydrostatic equations, *Geoscientific Model Development* 11 (2018) 4359–4382.
- [36] R. L. Higdon, Pressure forcing and dispersion analysis for discontinuous Galerkin approximations to oceanic fluid flows, *Journal of Computational Physics* 249 (2013) 36–66.

- [37] R. L. Higdon, Discontinuous Galerkin methods for multi-layer ocean modeling: Viscosity and thin layers, *Journal of Computational Physics* 401 (2020) 109018.
- [38] F. X. Giraldo, J. F. Kelly, E. M. Constantinescu, Implicit-Explicit Formulations of a Three-Dimensional Nonhydrostatic Unified Model of the Atmosphere (NUMA), *SIAM Journal on Scientific Computing* 35 (2013) B1162–B1194. URL: <https://doi.org/10.1137/120876034>. doi:10.1137/120876034. arXiv:<https://doi.org/10.1137/120876034>.
- [39] D. S. Abdi, F. X. Giraldo, Efficient construction of unified continuous and discontinuous Galerkin formulations for the 3d Euler equations, *Journal of Computational Physics* 320 (2016) 46–68.
- [40] M. A. Kopera, Y. Gahounzo, E. M. Enderlin, F. X. Giraldo, W. Maslowski, Non-hydrostatic unified model of the ocean with application to ice/ocean interaction modeling, *GEM-International Journal on Geomathematics* 14 (2023) 2.
- [41] E. P. Chassignet, H. E. Hurlburt, O. M. Smedstad, G. R. Halliwell, P. J. Hogan, A. J. Wallcraft, R. Bleck, Ocean prediction with the Hybrid Coordinate Ocean Model (HYCOM), *Ocean weather forecasting: an integrated view of oceanography* (2006) 413–426.
- [42] R. L. Higdon, An automatically well-balanced formulation of pressure forcing for discontinuous Galerkin methods for the shallow water equations, *Journal of Computational Physics* 458 (2022) 111102.
- [43] F. X. Giraldo, *An Introduction to Element-Based Galerkin Methods on Tensor-Product Bases: Analysis, Algorithms, and Applications*, volume 24, Springer Nature, Cham, Switzerland, 2020.
- [44] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland (1977).
- [45] B. Rivière, *Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation*, SIAM, 2008.
- [46] E. F. Toro, *The HLL and HLLC Riemann Solvers*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 315–344. URL: https://doi.org/10.1007/b79761_10. doi:10.1007/b79761_10.

- [47] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM journal on numerical analysis* 35 (1998) 2440–2463.
- [48] P. Castillo, B. Cockburn, I. Perugia, D. Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, *SIAM Journal on Numerical Analysis* 38 (2000) 1676–1706.
- [49] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations, *Journal of computational physics* 131 (1997) 267–279.
- [50] S. Ruuth, Global optimization of explicit strong-stability-preserving Runge-Kutta methods, *Mathematics of Computation* 75 (2006) 183–207.
- [51] R. Bleck, D. Boudra, Wind-driven spin-up in eddy-resolving ocean models formulated in isopycnic and isobaric coordinates, *Journal of Geophysical Research: Oceans* 91 (1986) 7611–7621.
- [52] C. Burstedde, L. C. Wilcox, O. Ghattas, **p4est**: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees, *SIAM Journal on Scientific Computing* 33 (2011) 1103–1133. doi:10.1137/100791634.
- [53] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities, *International Journal for Numerical Methods in Engineering* 79 (2009) 1309–1331.
- [54] E. G. Fernández, M. C. Díaz, M. Dumbser, T. M. De Luna, An arbitrary high order well-balanced ader-dg numerical scheme for the multilayer shallow-water model with variable density, *Journal of Scientific Computing* 90 (2022) 52.
- [55] F. X. Giraldo, J. S. Hesthaven, T. Warburton, Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations, *Journal of Computational Physics* 181 (2002) 499–525.
- [56] K. T. Mandli, Finite volume methods for the multilayer shallow water equations with applications to storm surges, University of Washington, 2011.

- [57] S. Marras, M. A. Kopera, F. X. Giraldo, Simulation of shallow-water jets with a unified element-based continuous/discontinuous Galerkin model with grid flexibility on the sphere, *Quarterly Journal of the Royal Meteorological Society* 141 (2015) 1727–1739.
- [58] M. A. Kopera, F. X. Giraldo, Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations, *Journal of Computational Physics* 275 (2014) 92–117.