Comparing non-hydrostatic extensions to a discontinuous finite element coastal ocean model

Wei Pan[a,*], Stephan C. Kramer[a], Tuomas Kärnä[b], Matthew D. Piggott[a]

[a]*Department of Earth Science and Engineering, Imperial College London, London, United Kingdom*
[b]*Finnish Meteorological Institute, Helsinki, Finland*

Abstract

The unstructured mesh, discontinuous Galerkin finite element discretisation based coastal ocean model, *Thetis*, has been extended to include non-hydrostatic (buoyancy-driven and free surface) dynamics. Two alternative approaches to achieve this are described in this work. The first (a 3D based algorithm) makes use of prismatic element based meshes and uses a split-step pressure projection method for baroclinic and barotropic modes, while the second (a 2D based algorithm) adopts a novel multi-layer approach to convert a 3D problem into a combination of multiple 2D computations with only 2D triangle meshes required. Model development is carried out at high-level with the Firedrake library, using code generation techniques to automatically produce low-level code for the discretised model equations in an efficient and rapid manner. Through comparisons against several barotropic/baroclinic test cases where non-hydrostatic effects are important, the implemented approaches are verified and validated, and the proposed algorithms compared. Depending on whether the problems are dominated by dispersive, baroclinic or barotropic features, recommendation are given over the use of full 3D or multi-layer 2D based approaches to achieve optimal computational accuracy and efficiency. It is demonstrated that while in general the 2D approach is well-suited for barotropic problems and dispersive free surface waves, the 3D approach is more advantageous for simulating baroclinic buoyancy-driven flows due in part to the high vertical resolution typically required to represent the active tracer fields.

Keywords: Discontinuous Galerkin, Finite element, Unstructured mesh, Baroclinic flow, Non-hydrostatic, Dispersion, Free surface

---

1. Introduction

Although considerable progress has been made in coastal ocean modelling, efficiently simulating free surface and buoyancy-driven flows across a range of scales is still a challenging topic. In terms of the complicated domains, bathymetry and the multi-scale dynamics present in coastal domains, unstructured meshes have certain advantages due to their flexibility over resolution and their ability to efficiently represent complex geometries (Fringer et al., 2006; Piggott et al., 2008b; Ringler et al., 2013; Kärnä et al., 2018).

Unstructured meshes are generally adopted in association with two types of (spatial) discretisation approaches: finite volume (FV) and finite element (FE) methods, which were first developed for problems dominated by hyperbolic and elliptic terms, respectively. Finite volume unstructured mesh based models (e.g. FVCOM, Chen et al., 2006;

*Corresponding author
Email address: w.pan16@imperial.ac.uk (Wei Pan)

SUNTANS, Fringer et al., 2006) and FE models (e.g. Fluidity, Piggott et al., 2008a; SELFE, Zhang and Baptista, 2008; SLIM, Kärnä et al., 2013) have a increasingly large community of users. Although they are indeed widely used, especially for coastal applications, it should also be noted that models based upon structured grids, e.g. GETM (Burchard and Bolding, 2002), ROMS (Shchepetkin and McWilliams, 2005), MITgcm (Adcroft et al., 2008), NEMO-OCE (Madec et al., 2015), are generally more mature and have wider user bases. It remains an active research area for unstructured mesh based models to optimise spatial discretisation methods in the context of computational efficiency, numerical stability and low levels of spurious numerical dissipation.

This article focuses on non-hydrostatic extensions to an unstructured mesh coastal ocean model, *Thetis* (Kärnä et al., 2018; Vouriot et al., 2018; Angeloudis et al., 2018; Kärnä, 2019). The initially developed three-dimensional (3D) and depth-integrated two-dimensional (2D) solvers are both hydrostatic, and are based upon discontinuous Galerkin finite element (DG-FE) discretisations. The 3D solver handles the hydrostatic form of the Navier-Stokes equations under the Boussinesq approximation for baroclinic applications, while the 2D one solves the non-linear shallow water equations. In general, the 3D solver on prismatic meshes is more complicated in terms of maintaining high numeral accuracy and stability. Specifically, a strong stability preserving (SSP) time integration scheme and a slope limiter (Kuzmin, 2010) are implemented, combined with a mode splitting method, e.g. Higdon and de Szoeke (1997), decomposing the full solve into barotropic (2D external) and baroclinic (3D internal) components. With an arbitrary Lagrangian-Eulerian (ALE) technique (Donea et al., 2004) the free surface position solved for in the 2D component is in accordance with the mesh movement in the 3D terrain-following σ-coordinate like system. As demonstrated by Kärnä et al. (2018) the formulation is fully conservative with both second-order spatial and temporal accuracy.

The initial development of *Thetis* utilised the hydrostatic assumption as is common in many other coastal ocean models. This assumption can allow for significant computational cost savings, but is only valid in situations where the horizontal dimensions of the considered dynamics are large compared to the vertical, and that vertical acceleration effects can be ignored. In the case of pronounced vertical flows and high-frequency internal/external waves, the inclusion of non-hydrostatic effects is very important for accurate representation of the physics, e.g. wave shoaling over a submerged bar (Beji and Battjes, 1994), submarine landslide generated tsunamis (Enet and Grilli, 2007), dispersive internal solitary waves (Horn et al., 2001), exchange flow with strong stratification (Härtel et al., 2000).

To study such oceanographic flows where non-hydrostatic effects matter, the full incompressible 3D Navier-Stokes equations should be solved, without invoking the hydrostatic approximation. Therefore, here we extend the hydrostatic kernel of *Thetis* so that non-hydrostatic effects are accounted for in both the 3D and 2D solvers. A resulting consideration in this work is the resulting computationally expensive Poisson equation for the non-hydrostatic component of pressure. Apart from exploiting the flexibility that results from using an unstructured mesh and refining the local resolution to improve computational efficiency, total pressure is decomposed into hydrostatic and non-hydrostatic components, and solved with a fractional step method (Chorin, 1968). This pressure-split technique, reducing the computational workload, is sometimes referred to as 'quasi-hydrostatic' (Fringer et al., 2006), and is an approach that is often used to extend a 3D hydrostatic ocean model to its corresponding non-hydrostatic version, e.g. SUNTANS (Fringer et al., 2006), ROMS (Kanarska et al., 2007) and FVCOM-NH (Lai et al., 2010). As with most of these, here a pressure projection approach is employed, where the non-hydrostatic pressure q is initially neglected in the momentum equation solve, but is then determined from the continuity equation and the intermediate velocity field updated to satisfy the divergence-free constraint.

In non-hydrostatic *Thetis* (hereafter *Thetis-NH*), here two approaches for the inclusion of non-hydrostatic effects are implemented and compared. We refer to these as 3D and 2D based algorithms which rely on 3D prismatic and 2D triangular meshes, respectively. The 3D prismatic mesh is obtained via the vertical extrusion of the 2D triangles which discretise the domain in the horizontal. The 3D algorithm is the direct extension of the original 3D hydrostatic solver (Kärnä et al., 2018) based on the above mentioned methods, while the 2D algorithm is more of a departure from the existing approach and takes advantage of a depth-integrated 2D framework to handle the vertical multi-layer system. In practice the 2D algorithm converts the underlying 3D problem into a series of multiple 2D calculations, i.e. it solves the 2D layer-integrated equations with extra terms describing inter-layer fluxes. With this approach (Stelling and Zijlema, 2003) the velocities are layer-averaged and are located at the centre of layers, while the non-hydrostatic pressure is defined at the layer interfaces, allowing for the pressure to be precisely assigned to zero at the free-surface boundary. As shown in our previous work (Pan et al., 2019), this approach with an edge-based compact difference scheme (Stelling and Zijlema, 2003) means that only a small number of vertical layers is required in order to accurately simulate dispersive free surface processes. As will be shown in the current work, however, for mass transfer and exchange flows the 2D algorithm does not demonstrate the same advantages, since high vertical resolution is generally necessary in order to represent the velocity and density fields for the problems considered, reducing the computational efficiency significantly. In contrast, the 3D algorithm, adopting the linear discontinuous Galerkin discretisation ($\mathrm{P}_1^{\mathrm{DG}}$) in both the horizontal and vertical, is less computationally efficient for barotropic free surface flows while it can ensure strict tracer conservation properly for baroclinic flows, and is also more efficient than the 2D algorithm for baroclinic problems.

In the work presented here, the layer-based equations making up the 2D based algorithm are re-formulated with viscosity and baroclinic terms included (which were not considered in (Pan et al., 2019)), giving a more general formulation which is fully consistent with the incompressible Navier-Stokes equations in σ-coordinates. It is demonstrated how the 2D algorithm can be applied in a fixed σ-coordinate transformed system with additional interface terms introduced to account for the moving reference frame (e.g. vertical mesh velocity, horizontal interface gradient). The 2D algorithm can thus, to a certain degree, be considered similar to the use of a piecewise-constant (i.e. $\mathrm{P}_0^{\mathrm{DG}}$) discretisation of the layer-averaged variables in the vertical direction.

In terms of the 3D algorithm, quite a different solution procedure is adopted. The mode splitting method that is used in the existing hydrostatic *Thetis* implementation is replaced here by a two-stage SSP Runge-Kutta scheme (Gottlieb and Shu, 1998) for the whole solution, dropping the additional forcing term that is needed in the former mode splitting approach to couple the 2D external and 3D internal dynamics. This means that the horizontal velocity field is not split into depth-averaged and deviation components, and the requirement for compatibility between the time integration approaches used for the two modes can be relaxed, with the new second-order explicit method to be used for each term allowing for the good maintenance of consistency and flexibility in time stepping. In addition, compared to the hydrostatic algorithm, the vertical velocity is not computed from the continuity equation, but from a newly introduced vertical momentum equation which includes the non-hydrostatic correction.

Discontinuous Galerkin finite element based discretisations are certainly utilised in coastal ocean models. However, most prior work has been conducted in a hydrostatic context (Blaise et al., 2010; Kärnä et al., 2013; Kärnä et al., 2018; Bonev et al., 2018). In the non-hydrostatic case finite difference (FD) or finite volume (FV) based methods are mostly used in, e.g. UNTRIM (Casulli and Zanolli, 2002), ROMS (Shchepetkin and McWilliams, 2005), SUNTANS (Fringer et al., 2006), FVCOM-NH (Lai et al., 2010), on structured or unstructured meshes. In this paper a novel unstructured mesh, DG-FE based non-hydrostatic kernels are developed within the code base of the hydrostatic *Thetis* coastal ocean model. Two algorithms – a full 3D and a 2D multi-layer (quasi-3D) – are introduced and compared. The 3D algorithm demonstrates accuracy and efficiency in the simulation of baroclinic stratified flows with strict tracer conservation. The 2D algorithm in contrast has strengths in the accurate simulation of barotropic free surface processes with relatively low vertical resolution requirements. This work thus discusses their respective applicability and limitations for barotropic/baroclinic flows. According to whether the flow is dominated by nearshore free surface or internal baroclinic processes recommendations can therefore be made over the most appropriate approach to maximise accuracy and efficiency. Moreover, this work provides an approach that can be utilised to extend other depth-integrated shallow-water flow solvers to include non-hydrostatic effects.

As with the wider Thetis project, our non-hydrostatic solver development is carried out using the automated finite element-based PDE solver platform Firedrake (Rathgeber et al., 2016). As opposed to conventional models written by hand in a static programming language, e.g. C or Fortran, *Thetis-NH* is based upon a code generation framework that produces C code automatically from a high-level description of the Finite Element, weak formulation of PDEs written using domain-specific language UFL (unified form language) (Logg et al., 2012). Without sacrificing computational efficiency, this model development approach is extremely flexible and thus very appealing for benchmarking various numerical discretisations. A number of attractive properties are offered including readability, extensibility and maintainability, as well as enhanced code efficiency and parallel scalability. In a comparison of hydrostatic *Thetis* with SLIM3D (Kärnä et al., 2013; Vallaeys et al., 2018) for a baroclinic eddy test case, it was shown (Kärnä et al., 2018) that *Thetis* runs faster than the SLIM3D model in serial, and manages to maintain this speed up in a strong scaling analysis with similar parallel efficiencies as SLIM3D.

The remainder of this paper is organised as follows. The governing equations are presented in Section 2. Section 3 describes the time integration schemes and the solution strategies for non-hydrostatic pressure. The DG based finite element discretisations and their implementation via Firedrake are described in Section 4. Several numerical test cases are presented in Section 5, followed by final conclusion in Section 6.

2. Governing equations

2.1. Domain and generic equations

The computational domain, denoted by Ω, is assumed to lie within a Cartesian coordinate system (x, y, z), with the z direction aligned with gravity and oriented upwards. The domain (Fig. 1) is delimited vertically by the sea floor $z = -d(x, y, t)$ (where we acknowledge the potential for this to vary with time in order to account for underwater landslides in this work (cf. Smith et al. (2016)), and geomorphology in related work (Clare et al., 2019)), and the free-surface elevation $z = \eta(x, y, t)$, with t indicating time. The time-dependent total water depth is thus denoted by $H(x, y, t) = \eta(x, y, t) + d(x, y, t)$.

In this work, two algorithms for the computation of non-hydrostatic dynamics are presented, which we term 3D and 2D based algorithms. In both cases the governing equations are the 3D incompressible Navier-Stokes momentum and continuity equations:

$$\frac{\partial \hat{\boldsymbol{u}}}{\partial t} + \hat{\boldsymbol{u}} \cdot \nabla \hat{\boldsymbol{u}} - \nu \nabla^2 \hat{\boldsymbol{u}} = -\frac{1}{\rho} \nabla p + \boldsymbol{g} + \boldsymbol{S}, \tag{1}$$

$$\nabla \cdot \hat{\boldsymbol{u}} = 0, \tag{2}$$

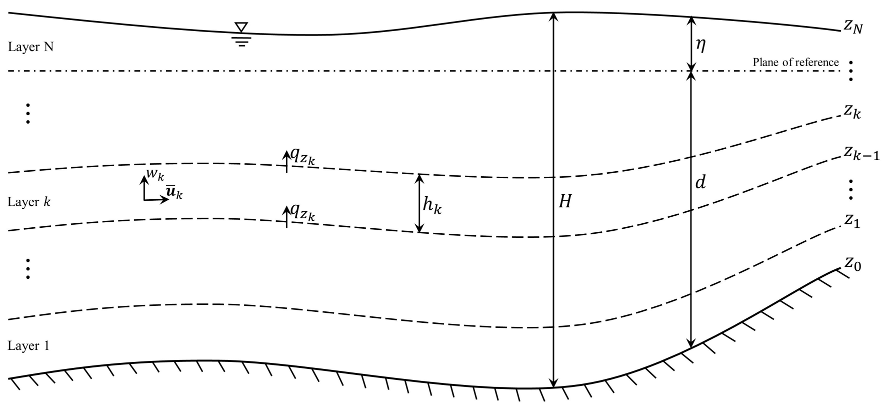

Figure 1: Computational domain bounded by the bottom boundary and the free surface with vertical layers and variables associated with the 2D algorithm.

where $\hat{\boldsymbol{u}} = (u, v, w)^T$ is the 3D velocity vector, ρ is the fluid density, p is the pressure, ν is the kinematic viscosity, $\boldsymbol{g}$ is the gravitational acceleration represented by $[0, 0, -g]^T$, and $\boldsymbol{S}$ presents any source terms.

2.2. The 3D based algorithm

In the 3D based algorithm, the discretisation occurs over prismatic 3D elements which are constructed by vertically extruding a horizontal 2D unstructured mesh. In this work this extrusion takes the form of a σ-coordinate-like mesh conforming to both the free surface and the sea floor made up of N layers everywhere, as shown in Fig. 1. This assumption could easily be generalised to z coordinate or hybrid meshes in the vertical. As part of this formulation a generic Arbitrary Lagrangian-Eulerian (ALE) approach (Donea et al., 2004) is utilised to handle the deformation of the grid. This deformation can be due to both perturbations of the free surface and/or a movable sea floor. A linear distribution of these boundary deformations is assumed over the water column in this work (again this could easily be generalised), and a mesh velocity w_m is introduced to account for the corresponding mesh movement.

Based on the properties of a prismatic mesh, it is beneficial to decouple the horizontal component of velocity $\boldsymbol{u}$ and the vertical velocity w, resulting in the governing equations used for the 3D algorithm written in the form

$$\nabla_h \cdot \boldsymbol{u} + \frac{\partial w}{\partial z} = 0, \tag{3}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_h \cdot (\boldsymbol{u}\boldsymbol{u}) + \frac{\partial (w - w_m)\boldsymbol{u}}{\partial z} = -\frac{1}{\rho_0}\nabla_h p - f\mathbf{e}_z \times \boldsymbol{u} + \nabla_h \cdot (\nu_h \nabla_h \boldsymbol{u}) + \frac{\partial}{\partial z}\left(\nu_v \frac{\partial \boldsymbol{u}}{\partial z}\right), \tag{4}$$

$$\frac{\partial w}{\partial t} + \nabla_h \cdot (\boldsymbol{u}w) + \frac{\partial (w - w_m)w}{\partial z} = -\frac{1}{\rho_0}\frac{\partial p}{\partial z} - \frac{\rho}{\rho_0}g + \nabla_h \cdot (\nu_h \nabla_h w) + \frac{\partial}{\partial z}\left(\nu_v \frac{\partial w}{\partial z}\right), \tag{5}$$

where $\boldsymbol{u}(x, y, z, t) = [u, v]^T$ represents the horizontal velocity vector, ∇_h is the horizontal gradient operator, ρ_0 is the constant mean density from the Boussinesq approximation, f is the Coriolis parameter, $\boldsymbol{e}_z$ is the vertical unit vector, ν_h and ν_v are the horizontal and vertical eddy viscosities, respectively.

Recalling the potential for the bottom of the domain as well as the free surface to move in this work, the mesh velocity w_m can be computed via (Kärnä et al., 2018)

$$w_m = \frac{\partial \eta}{\partial t}\frac{z + d}{\eta + d} + \frac{\partial d}{\partial t}\frac{z - \eta}{\eta + d}. \tag{6}$$

Impermeability boundary conditions at these boundaries are given by

$$w|_{z=\eta} = \frac{\partial \eta}{\partial t} + \boldsymbol{u}|_{z=\eta} \cdot \nabla_h \eta, \tag{7}$$

$$w|_{z=-d} = -\frac{\partial d}{\partial t} - \boldsymbol{u}|_{z=-d} \cdot \nabla_h d. \tag{8}$$

Using (7)–(8), and integrating (3) over depth, we obtain

$$\frac{\partial H}{\partial t} + \nabla_h \cdot (H\bar{\boldsymbol{u}}) = 0, \quad \text{with} \quad \bar{\boldsymbol{u}} = \frac{1}{H}\int_{-d}^{\eta} \boldsymbol{u}\, dz. \tag{9}$$

Note that throughout the paper the over-bar notation means depth average.

The total pressure is decomposed into a hydrostatic component p_h and a non-hydrostatic component q (Marshall et al., 1997; Casulli and Stelling, 1998), i.e. $p = p_h + q$, with the hydrostatic pressure defined as follows

$$\frac{\partial p_h}{\partial z} = -g\rho \qquad \text{with} \qquad \rho = \rho_0 + \rho'(T,S), \tag{10}$$

where the density deviation $\rho'(T,S)$ associated with temperature T and salinity S is computed based on an appropriate equation of state.

Integrating (10) from the free surface η down to a vertical location z, and adding the non-hydrostatic pressure q, leads to the relation

$$p = p_0 + g\rho_0(\eta - z) + g\int_{z}^{\eta} \rho'\, dz' + q, \tag{11}$$

where p_0 denotes the atmospheric pressure, which is neglected in this work.

Defining the baroclinic pressure head as

$$r = \frac{1}{\rho_0}\int_{z}^{\eta} \rho'\, dz', \tag{12}$$

the pressure gradient term in the momentum equations (4), (5) can be written as

$$\frac{1}{\rho_0}\nabla_h p = g\nabla_h(\eta + r) + \frac{1}{\rho_0}\nabla_h q, \tag{13}$$

$$\frac{1}{\rho_0}\frac{\partial p}{\partial z} = -\frac{\rho}{\rho_0}g + \frac{1}{\rho_0}\frac{\partial q}{\partial z}. \tag{14}$$

The full momentum equations then take the form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_h \cdot (\boldsymbol{u}\boldsymbol{u}) + \frac{\partial (w\boldsymbol{u})}{\partial z} + f\mathbf{e}_z \times \boldsymbol{u} + g\nabla_h(\eta + r) + \frac{1}{\rho_0}\nabla_h q = \nabla_h \cdot (\nu_h \nabla_h \boldsymbol{u}) + \frac{\partial}{\partial z}\left(\nu_v \frac{\partial \boldsymbol{u}}{\partial z}\right), \tag{15}$$

$$\frac{\partial w}{\partial t} + \nabla_h \cdot (\boldsymbol{u}w) + \frac{\partial (ww)}{\partial z} + \frac{1}{\rho_0}\frac{\partial q}{\partial z} = \nabla_h \cdot (\nu_h \nabla_h w) + \frac{\partial}{\partial z}\left(\nu_v \frac{\partial w}{\partial z}\right), \tag{16}$$

where for the sake of brevity, the vertical mesh velocity terms have been omitted although they are included in the model.

To simulate temperature and salinity, a generic advection-diffusion tracer equation is adopted:

$$\frac{\partial c}{\partial t} + \nabla_h \cdot (\boldsymbol{u}c) + \frac{\partial (wc)}{\partial z} = \nabla_h \cdot (\kappa_h \nabla_h c) + \frac{\partial}{\partial z}\left(\kappa_v \frac{\partial c}{\partial z}\right), \tag{17}$$

where the 3D tracers $c(x, y, z, t)$ can be temperature T or salinity S, and κ_h, κ_v represent the horizontal and vertical diffusivities, respectively. In this study for simplicity a linear equation of state is assumed of the form

$$\rho(T, S) = \rho_0 - \alpha_T (T - T_0) + \beta_S (S - S_0), \tag{18}$$

where α_T, β_S are the thermal expansion and saline contraction coefficients, and T_0, S_0 denote reference temperature and salinity values.

2.3. 2D based algorithm

In the 2D based kernel of the model, a multi-layer approach based upon the approach of Stelling and Zijlema (2003) is developed targeted in particular at the efficient simulation of nearshore dispersive free surface waves (Pan et al., 2019). The approach solves for layer-averaged momentum and inter-layer fluxes over the layers of the system. In this work the approach presented in (Pan et al., 2019) is extended to incorporate viscosity and baroclinic effects. Specifically, a new general formulation is given which is fully consistent with the incompressible Navier-Stokes equations in σ-coordinates, and layer-averaged continuity equations used for a Poisson equation are newly constructed using a combination of layer-averaged vertical velocities.

The underlying equations are integrated over each layer to obtain expressions for the average velocities located at the centre of each layer. The layer-averaged equations have a similar form to the classical depth-integrated equations, but with extra terms associated with the transfer of momentum and mass flux between adjacent layers. Via an edge-based compact difference scheme (a type of Keller-box scheme) (Stelling and Zijlema, 2003), a 3D problem can be converted into a combination of multiple 2D computations, and with only a relatively small number of vertical layers, the 2D based model has been demonstrated to efficiently predict nearshore dispersive wave processes (Pan et al., 2019).

Integrating equations (3), (15)–(17) over the kth (where $1 \le k \le N$) layer, decomposing the total pressure into hydrostatic and non-hydrostatic parts, applying Leibniz's rule, dropping the residual diffusion-like terms from the layer integration of advection (Bai and Cheung, 2016), and terms involving Coriolis, bed friction, wind stress, etc., we obtain layer-integrated 2D equations in the conservative form

$$\frac{\partial h_k}{\partial t} + \nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k) + M|_{z_{k-1}} - M|_{z_k} = 0, \tag{19}$$

$$\frac{\partial h_k \bar{\boldsymbol{u}}_k}{\partial t} + \nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k \bar{\boldsymbol{u}}_k) + g h_k \nabla_h (\eta + r_k) + \frac{1}{\rho_0} Q_k = \nabla_h \cdot (\nu_h \nabla_h (h_k \bar{\boldsymbol{u}}_k)) + F^{\boldsymbol{u}}|_{z_{k-1}} - F^{\boldsymbol{u}}|_{z_k}, \tag{20}$$

$$\frac{\partial h_k \bar{w}_k}{\partial t} + \nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k \bar{w}_k) + \frac{1}{\rho_0}(q_{z_k} - q_{z_{k-1}}) = \nabla_h \cdot (\nu_h \nabla_h (h_k \bar{w}_k)) + F^{w}|_{z_{k-1}} - F^{w}|_{z_k}, \tag{21}$$

$$\frac{\partial h_k \bar{c}_k}{\partial t} + \nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k \bar{c}_k) = \nabla_h \cdot (\kappa_h \nabla_h (h_k \bar{c}_k)) + F^{c}|_{z_{k-1}} - F^{c}|_{z_k}, \tag{22}$$

with

$$\begin{aligned}
\bar{f}_k &= \frac{1}{h_k} \int_{z_{k-1}}^{z_k} f \, dz, \quad \forall f \in \{\boldsymbol{u}, w, c\}, \\
M &= \frac{\partial z}{\partial t} + \boldsymbol{u} \cdot \nabla_h z - w, \\
Q_k &= \int_{z_{k-1}}^{z_k} \nabla_h q \, dz = \nabla_h \int_{z_{k-1}}^{z_k} q \, dz + q_{z_{k-1}} \nabla_h z_{k-1} - q_{z_k} \nabla_h z_k, \\
F^f &= -M f + \left[\nu_h \left(2 \nabla_h f \cdot \nabla_h z + f \nabla_h^2 z \right) - \nu_v \frac{\partial f}{\partial z} \right], \qquad \forall f \in \{\boldsymbol{u}, w, c\},
\end{aligned}$$

where z_k is the vertical coordinate of the layer interfaces from the bottom $z_0 = -d$ to the free surface $z_N = \eta$, and $h_k = z_k - z_{k-1}$ the kth layer thickness; $\bar{\boldsymbol{u}}_k$, $\bar{w}_k$, and $\bar{c}_k$ are the layer-averaged velocities and tracer; Q_k represents the layer-averaged non-hydrostatic pressure gradient terms, and q_{z_k} is the non-hydrostatic pressure located at the interfaces rather than layer centres. This allows for $q_{z_N} = 0$ to be specified exactly at the free surface and thus reducing the number of vertical layers required to accurately simulate certain problems (see Fig. 1).

The term M represents the mass flux across a layer interface, which is zero for both the bottom and free surface in accordance with the impermeability boundary conditions. The interface velocity term $\frac{\partial z}{\partial t}$ is actually identical to the mesh velocity (6) in the 3D algorithm. The F^f terms at the layer interfaces are naturally generated collecting boundary terms from the layer integration. The first term in F^f (i.e. $M\boldsymbol{u}$, Mw or Mc) denotes convective fluxes between layers and the terms in the square brackets indicate shear stresses (diffusive fluxes) on the interfaces. At the bottom interface, $k = 0$, the latter is replaced by the bottom friction, and at the free surface, $k = N$, by zero as no wind stress is applied in any of the test cases considered. It should be noted that for the one layer case ($N = 1$) equations (19) – (20) revert to the depth-integrated system with the free surface equation (9).

For the time-integration in the next section we derive the equivalent layer-*averaged* equations:

$$\frac{\partial \bar{\boldsymbol{u}}_k}{\partial t} + \bar{\boldsymbol{u}}_k \cdot \nabla_h \bar{\boldsymbol{u}}_k + g\nabla_h(\eta + r_k) + \frac{1}{\rho_0 h_k} Q_k = \nabla_h \cdot (\nu_h \nabla_h \bar{\boldsymbol{u}}_k) + S_k^{\boldsymbol{u}}, \tag{23}$$

$$\frac{\partial \bar{w}_k}{\partial t} + \bar{\boldsymbol{u}}_k \cdot \nabla_h \bar{w}_k + \frac{1}{\rho_0} \frac{q_{z_k} - q_{z_{k-1}}}{h_k} = \nabla_h \cdot (\nu_h \nabla_h \bar{w}_k) + S_k^w \tag{24}$$

$$\frac{\partial \bar{c}_k}{\partial t} + \bar{\boldsymbol{u}}_k \cdot \nabla_h \bar{c}_k = \nabla_h \cdot (\kappa_h \nabla_h \bar{c}_k) + S_k^c \tag{25}$$

$$S_k^f = \frac{1}{h_k}\Big(F^f|_{z_{k-1}} - F^f|_{z_k} + \big(M_{z_{k-1}} - M_{z_k}\big)\bar{f}_k + 2\nu_h \nabla_h \bar{f} \cdot \nabla_h h_k + \bar{f}\nabla_h^2 h_k \Big). \tag{26}$$

For the vertical spatial discretisation, the degrees of freedom are defined to be the layer-averaged horizontal velocities $\bar{\boldsymbol{u}}_k$, and tracer values $\bar{c}_k$ for each layer. The vertical velocities w_{z_k}, however, are stored at the interface beween the layers. Horizontal velocities and tracers at the interface, and layer-averaged vertical velocities are then approximated as:

$$\boldsymbol{u}_{z_k} = \frac{h_{k+1}\bar{\boldsymbol{u}}_k + h_k \bar{\boldsymbol{u}}_{k+1}}{h_k + h_{k+1}}, \quad c_{z_k} = \frac{h_{k+1}\bar{c}_k + h_k \bar{c}_{k+1}}{h_k + h_{k+1}}, \quad \bar{w}_k = \frac{w_{z_{k-1}} + w_{z_k}}{2}, \tag{27}$$

To enable time stepping of (23), and to facilitate the construction of a Poisson equation for the non-hydrostatic pressure, we also need the inverse of the mapping between interface and layer-averaged vertical velocities. Given layer-averaged vertical velocities $\bar{w}_k$, and the vertical velocity at the bottom boundary $w_{z_0} = \frac{\partial z_0}{\partial t} + \boldsymbol{u}_{z_0} \cdot \nabla_h z_0$ derived from the bottom boundary condition (8), we can compute the vertical velocities at all layer interfaces above by

$$\begin{aligned}
w_{z_1} &= 2\bar{w}_1 - w_{z_0}, \\
w_{z_2} &= 2\bar{w}_2 - w_{z_1} = 2\bar{w}_2 - 2\bar{w}_1 + w_{z_0}, \\
&\dots \\
w_{z_k} &= 2\bar{w}_k - 2\bar{w}_{k-1} + 2\bar{w}_{k-2} - \dots (-1)^k w_{z_0}.
\end{aligned} \tag{28}$$

For use in the Poisson equation, we expand (19) to

$$\nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k) + w_{z_k} - w_{z_{k-1}} = \boldsymbol{u}_{z_k} \cdot \nabla_h z_k - \boldsymbol{u}_{z_{k-1}} \cdot \nabla_h z_{k-1}. \tag{29}$$

and substitute (28), to obtain

$$\nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k) - 2\bar{w}_k + 4\sum_{n=0}^{k-1} (-1)^n \bar{w}_{k-n} = \boldsymbol{u}_{z_k} \cdot \nabla_h z_k - \boldsymbol{u}_{z_{k-1}} \cdot \nabla_h z_{k-1} - 2(-1)^k w_{z_0}. \tag{30}$$

Finally, summing (19) over all layers yields an equation for the free surface elevation η

$$\frac{\partial \eta}{\partial t} + \sum_{k=1}^{N} \nabla_h \cdot (h_k \bar{\boldsymbol{u}}_k) = 0, \tag{31}$$

which can be seen as a discretised form of (9). In terms of the depth-averaged velocity $\bar{\boldsymbol{u}} = \sum_{k=1}^{N} (h_k \bar{\boldsymbol{u}}_k)/H$, the depth-integrated horizontal momentum equation in the conservative form is introduced:

$$\frac{\partial H\bar{\boldsymbol{u}}}{\partial t} + \nabla_h \cdot (H\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}) + gH\nabla_h(\eta + \bar{r}) + \frac{1}{\rho_0}\sum_{k=1}^{N} Q_k = \nabla_h \cdot (H\nu_h \nabla_h \bar{\boldsymbol{u}}). \tag{32}$$

3. Time integration and pressure solution

In this section, time integration schemes are presented for use with the non-hydrostatic modelling approaches. Given the different solution strategies for pressure, the specific algorithms are described separately in the following sub-sections.

3.1. Solution procedure for the 3D algorithm

In the 3D version of our non-hydrostatic model, a two-stage ALE time integration scheme is adopted. The whole solution is advanced in time using a second-order strong stability preserving (SSP) Runge-Kutta scheme, SSPRK(2,2) (Gottlieb and Shu, 1998). It should be noted that the time integration in this section, even excluding the non-hydrostatic terms, differs from that discussed in Kärnä et al. (2018). For simplicity we have omitted the 2D, 3D split, and the associated 2D-3D coupling terms, with both the 2D and 3D dynamics treated explicitly. This incurs a timestep restriction typically associated with the barotropic mode. For the test cases presented in this paper, and comparisons with the 2D layered approach, which focus on spatial accuracy, this limitation is not significant. For larger scale applications the time-integration method will be extended to improve stability with large time steps.

For convenience the momentum and tracer equations (15) – (17) are rewritten as

$$\frac{\partial \boldsymbol{u}}{\partial t} + g\nabla_h \eta + \frac{1}{\rho_0}\nabla_h q = \mathcal{A}_e(\boldsymbol{u}, w, r) + \mathcal{A}_i(\boldsymbol{u}), \tag{33}$$

$$\frac{\partial w}{\partial t} + \frac{1}{\rho_0}\frac{\partial q}{\partial z} = \mathcal{A}_e(w, \boldsymbol{u}) + \mathcal{A}_i(w), \tag{34}$$

$$\frac{\partial c}{\partial t} = \mathcal{A}_e(c, \boldsymbol{u}, w) + \mathcal{A}_i(c), \tag{35}$$

where the subscripts e and i denote the terms are treated explicitly and implicitly, respectively; r is the baroclinic head (12). $\mathcal{A}_i(\boldsymbol{u})$, $\mathcal{A}_i(w)$, and $\mathcal{A}_i(c)$ contain all the implicit terms. Following the hydrostatic approach of *Thetis*, vertical viscosity, vertical diffusion and bottom friction terms can be treated implicitly, although in the test cases considered later in this work they are treated explicitly in order to make a fair comparison against the current implementation of the 2D based algorithm. The remaining terms are treated explicitly.

For a generic problem

$$\frac{\partial f}{\partial t} = \mathcal{A}(f), \tag{36}$$

the two-stage SSP Runge-Kutta scheme reads

$$\frac{f^{(1)} - f^n}{\Delta t} = \mathcal{A}(f^n), \tag{37}$$

$$\frac{f^{(2)} - f^{(1)}}{\Delta t} = \mathcal{A}(f^{(1)}), \tag{38}$$

$$f^{n+1} = \frac{1}{2} f^n + \frac{1}{2} f^{(2)}. \tag{39}$$

When applied to the momentum and tracer equations (33) – (35), these stages are with ALE updates, i.e. the computational mesh is updated from domain Ω^n to $\Omega^{(1)}$ and then Ω^{n+1}.

At the first stage, a split-step pressure projection method is used to evaluate the intermediate level (denoted by $*$) of the first estimate $\boldsymbol{u}^{(1)}$, $w^{(1)}$ and $c^{(1)}$:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\Delta t} = -g\nabla_h \eta^n + \mathcal{A}_e(\boldsymbol{u}^n, w^n, r^n), \qquad \frac{\boldsymbol{u}^{(1)} - \boldsymbol{u}^*}{\Delta t} = -\frac{1}{\rho_0}\nabla_h q^{(1)}, \tag{40}$$

$$\frac{w^* - w^n}{\Delta t} = \mathcal{A}_e(w^n, \boldsymbol{u}^n), \qquad \frac{w^{(1)} - w^*}{\Delta t} = -\frac{1}{\rho_0}\frac{\partial q^{(1)}}{\partial z}, \tag{41}$$

$$\frac{c^{(1)} - c^n}{\Delta t} = \mathcal{A}_e(c^n, \boldsymbol{u}^{(1)}, w^{(1)} - w_m^{(1)}), \tag{42}$$

where for brevity the implicit terms $\mathcal{A}_i$ have been omitted although they are included in the model; the free surface elevation $\eta^{(1)}$ is obtained by explicitly solving (9) with the depth-averaged velocity $\bar{\boldsymbol{u}}^{(1)}$ specified, which is then used to update the mesh geometry and compute the mesh velocity $w_m^{(1)}$ based on (6). The tracer equation is advanced on the new mesh geometry, and then the baroclinic head $r^{(1)}$ is computed.

At the second stage, the velocity and tracer fields are again updated to a second intermediate level using the same pressure projection, after which a final value of fields at the $n+1$ time level is obtained:

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^{(1)}}{\Delta t} = -g\nabla_h \eta^{(1)} + \mathcal{A}_e(\boldsymbol{u}^{(1)}, w^{(1)} - w_m^{(1)}, r^{(1)}), \qquad \frac{\boldsymbol{u}^{(2)} - \boldsymbol{u}^*}{\Delta t} = -\frac{1}{\rho_0}\nabla_h q^{(2)}, \tag{43}$$

$$\frac{w^* - w^{(1)}}{\Delta t} = \mathcal{A}_e(w^{(1)}, \boldsymbol{u}^{(1)}, w_m^{(1)}), \qquad \frac{w^{(2)} - w^*}{\Delta t} = -\frac{1}{\rho_0}\frac{\partial q^{(2)}}{\partial z}, \tag{44}$$

$$\boldsymbol{u}^{n+1} = \frac{1}{2}\boldsymbol{u}^n + \frac{1}{2}\boldsymbol{u}^{(2)}, \qquad w^{n+1} = \frac{1}{2}w^n + \frac{1}{2}w^{(2)}, \tag{45}$$

$$\frac{c^{(2)} - c^{(1)}}{\Delta t} = \mathcal{A}_e(c^{(1)}, \boldsymbol{u}^{n+1}, w^{n+1} - w_m^{n+1}), \tag{46}$$

$$c^{n+1} = \frac{1}{2}c^n + \frac{1}{2}c^{(2)}, \tag{47}$$

where the vertical velocity in (43) – (44) is adjusted by the mesh velocity $w_m^{(1)}$ due to the ALE geometry update. The free surface equation (9) is solved again after getting the final velocity solution in (45), leading to an updated elevation field from $\eta^{(1)}$ to η^{n+1} and consequently mesh geometry to Ω^{n+1}. It should be noted that at the second stage the tracer equation is advanced using the final velocities $\boldsymbol{u}^{n+1}$ and w^{n+1}, rather than $\boldsymbol{u}^{(2)}$ and $w^{(2)}$.

At each stage, given the intermediate and non-divergence-free velocities $\boldsymbol{u}^*$ and w^*, the non-hydrostatic pressure $q^{(k)}$ ($k = 1,2$ denotes the kth stage in the Runge–Kutta scheme) is used to enforce the divergence-free condition, i.e. we have a pressure update of the form

$$\boldsymbol{u}^{(k)} = \boldsymbol{u}^* - \frac{\Delta t}{\rho_0}\nabla_h q^{(k)}, \tag{48}$$

$$w^{(k)} = w^* - \frac{\Delta t}{\rho_0}\frac{\partial q^{(k)}}{\partial z}. \tag{49}$$

Substituting (48) and (49) into the continuity equation (3), yields the Poisson equation

$$\nabla^2 q^{(k)} = \frac{\rho_0}{\Delta t}\left(\nabla_h \cdot \boldsymbol{u}^* + \frac{\partial w^*}{\partial z}\right). \tag{50}$$

With appropriate boundary conditions, this equation for the non-hydrostatic pressure $q^{(k)}$ is solved. On closed boundaries, the zero Neumann and impermeability conditions are specified, while in terms of open boundaries, e.g. at inflows and the free surface, a Dirichlet condition $q = 0$ is enforced. After $q^{(k)}$ is obtained, the velocities are corrected and the surface elevation is updated.

3.2. Solution procedure for the 2D algorithm

The pressure projection method (Chorin, 1968) used in the proposed 2D algorithm is a specific type of split-step method, in which a Poisson equation is configured by first solving the layer-averaged momentum equations without the non-hydrostatic terms, and then substituting the corrected velocities dependent on q_{z_k} into the layer-based continuity equations.

In terms of the temporal discretisation, the θ-scheme is adopted in the 2D algorithm for the free surface and the pressure gradient terms, and an explicit scheme is used for advection and the remaining terms. If $\theta = 0.5$, the θ-scheme becomes a Crank-Nicolson scheme, while $\theta = 0$ corresponds to an explicit Euler scheme. For the test cases presented in this paper, $\theta = 0.5$ is applied.

In the first solution step, known as the hydrostatic step, the temporal discretisation of the depth-integrated equations (9) and the non-conservative form of (32) yield the following

$$\frac{\eta^* - \eta^n}{\Delta t} + \theta\nabla_h \cdot (H^n\bar{\boldsymbol{u}}^*) + (1-\theta)\nabla_h \cdot (H^n\bar{\boldsymbol{u}}^n) = 0, \tag{51}$$

$$\frac{\bar{\boldsymbol{u}}^* - \bar{\boldsymbol{u}}^n}{\Delta t} + \bar{\boldsymbol{u}}^n \cdot \nabla_h\bar{\boldsymbol{u}}^n + g\Big(\theta\nabla_h\eta^* + (1-\theta)\nabla_h\eta^n\Big) = F^n_{\bar{\boldsymbol{u}}}, \tag{52}$$

where $F^n_{\bar{\boldsymbol{u}}}$ denotes terms that are treated explicitly, including viscosity, baroclinicity and source terms. For $\theta \geq 0.5$ the implicitness in the divergence and pressure gradient terms ensures stability independent of the external wave celerity Courant number.

Here we define the coefficients $\alpha_k = h_k/H$ to represent the ratio of layer thickness to the total water depth. The velocity depth-averaged over the entire water column is then

$$\bar{\boldsymbol{u}} = \sum_{k=1}^{N}\alpha_k\bar{\boldsymbol{u}}_k. \tag{53}$$

The coupled equations (51) – (52) are first solved for the two unknowns η^* and $\bar{\boldsymbol{u}}^*$. Based on (23), the layer-averaged velocities $\bar{\boldsymbol{u}}_k$ are then obtained using this η^*:

$$\frac{\bar{\boldsymbol{u}}^*_k - \bar{\boldsymbol{u}}^n_k}{\Delta t} + \bar{\boldsymbol{u}}^n_k \cdot \nabla_h\bar{\boldsymbol{u}}^n_k + g\nabla_h\eta^* = F^n_{\bar{\boldsymbol{u}}_k}, \qquad \forall k \in [1, N-1]. \tag{54}$$

Note for the surface layer ($k = N$), we do not need to solve for the velocity $\bar{\boldsymbol{u}}_N$, as this can be recovered directly from (53), i.e. $\bar{\boldsymbol{u}}_N^* = \bar{\boldsymbol{u}}^* - \sum_{k=1}^{N-1} \alpha_k \bar{\boldsymbol{u}}_k^*$.

Similarly, the vertical velocities $\bar{w}_k$ are obtained:

$$\frac{\bar{w}_k^* - \bar{w}_k^n}{\Delta t} + \bar{\boldsymbol{u}}_k^* \cdot \nabla_h \bar{w}_k^n = F_{\bar{w}_k}^n, \qquad \forall k \in [1, N]. \tag{55}$$

In the second solution step, i.e. the non-hydrostatic step, the velocities from the first step are corrected using the non-hydrostatic pressure:

$$\bar{\boldsymbol{u}}_k^{n+1} = \bar{\boldsymbol{u}}_k^* - \Delta t Q_k^{n+1}, \qquad \forall k \in [1, N], \tag{56}$$

$$\bar{w}_k^{n+1} = \bar{w}_k^* - \Delta t \frac{q_{z_k}^{n+1} - q_{z_{k-1}}^{n+1}}{h_k}, \qquad \forall k \in [1, N], \tag{57}$$

with

$$Q_k^{n+1} = \frac{1}{2} \nabla_h \left(q_{z_{k-1}}^{n+1} + q_{z_k}^{n+1} \right) - \frac{q_{z_k}^{n+1} - q_{z_{k-1}}^{n+1}}{h_k} \nabla_h \frac{z_{k-1} + z_k}{2}. \tag{58}$$

The corrected velocities (56) – (57) are substituted into the constructed layer-averaged continuity equations (30), yielding a Poisson equation for the non-hydrostatic pressure $\boldsymbol{q} = [q_{z_0}, q_{z_1}, \ldots, q_{z_{N-1}}]^T$ (with $q_{z_N} = 0$ at the free surface) in the form of

$$\boldsymbol{A} \nabla_h^2 \boldsymbol{q} + \boldsymbol{B} \nabla_h \boldsymbol{q} + \boldsymbol{C} \boldsymbol{q} + \boldsymbol{D} = \boldsymbol{0}, \tag{59}$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are $N \times N$ block matrices, and $\boldsymbol{D}$ is a $N \times 1$ block matrix, in all of which the submatrices have the same shape which is dependent on the resolution of the 2D horizontal mesh. Note that the entries in the matrices $\boldsymbol{A}$–$\boldsymbol{D}$ are associated with known terms, including $\alpha_1, \ldots, \alpha_N$, the total water depth H^*, the intermediate and previous time step level velocities, etc.

Finally, the layer-averaged velocities are updated using the obtained non-hydrostatic pressure, and are then used to advance the tracer and update the surface elevation by explicitly solving (25) and (31), respectively.

4. Finite Element Discretisation

4.1. Function spaces

The proposed full 3D and multi-layer 2D non-hydrostatic algorithms adopt different types of mesh, where the prismatic elements used in the 3D algorithm are obtained from the extrusion of horizontal triangular elements which are used in the 2D algorithm. The common horizontal domain Γ is partitioned into a finite set of triangular sub-domains $\mathit{\Gamma}_e$, upon which a piecewise polynomial function space $\mathbb{W}$ is defined based on Lagrangian basis functions $\phi_i : E \subset \mathbb{R}^2 \to \mathbb{R}$ which are polynomials of a specified order when restricted to an element (triangle) $E \in \mathit{\Gamma}_e$. In this paper we restrict ourselves to linear polynomials and no continuity is assumed between adjacent triangles, i.e. we consider the discontinuous Galerkin discretisation using $\mathbb{W} = \mathrm{P}_1^{\mathrm{DG}}(\mathit{\Gamma}_e)$.

Similarly, for the 3D domain Ω, the finite set of prisms is defined as $\mathit{\Omega}_e$ where each prism $E \in \mathit{\Omega}_e$ is the Cartesian product $E = T \times V$ of a triangle $T \in \mathit{\Gamma}_e$, and interval $V \subset \mathbb{R}$ which is a sub-interval of the vertical water column. The function space defined on $\mathit{\Omega}_e$ is the tensor product of the horizontal and vertical function space, e.g. $\mathbb{V} = \mathrm{P}_1^{\mathrm{DG}}(\mathit{\Gamma}_e) \otimes \mathrm{P}_1^{\mathrm{DG}}(V_e)$, where V_e is a sub-division of the water column into intervals $V \in V_e$. For simplicity of notation we have assumed a constant vertical water column with the same sub-division throughout the domain, but this is readily generalized for variable bathymetry and layer depths.

In addition to the scalar function spaces $\mathbb{W} = \mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$ and $\mathbb{V} = \mathrm{P}_1^{\mathrm{DG}}(\Gamma_e) \otimes \mathrm{P}_1^{\mathrm{DG}}(V_e)$, we define the vector function spaces

$$\mathbb{W} \times \mathbb{W} = \left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\right]^2 \text{ and } \mathbb{V} \times \mathbb{V} = \left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e) \otimes \mathrm{P}_1^{\mathrm{DG}}(V_e)\right]^2$$

for horizontal velocity fields in the 2D and 3D domains respectively.

In terms of the depth-integrated equations, a velocity–elevation element pair ($\mathrm{P}_1^{\mathrm{DG}}$–$\mathrm{P}_1^{\mathrm{DG}}$) is used to discretise this system. In the 2D algorithm which only involves 2D variables, more flexibility over function spaces is currently available, e.g. higher order continuous/discontinuous Galerkin finite element spaces (Pan et al., 2019). It should also be noted that a quadratic continuous Galerkin discretisation, P_2, is used as the vertical function space for the 3D non-hydrostatic pressure. Although it is possible to adopt $\mathrm{P}_1^{\mathrm{DG}}$ as well, our preliminary experiments indicated that this choice appears to deteriorate the accuracy in solving the non-hydrostatic pressure, and thus does not resolve the the frequency dispersion as well. Formal assessment in terms of discretisations of the non-hydrostatic pressure will be investigated in future work.

To make the derivation of the discretisation concise, we use the following shorthand notation for the spatial integrals that appear in the weak forms:

$$\langle \bullet \rangle_\Gamma = \sum_{E \in \Gamma_e} \int_E \bullet \, d\boldsymbol{x}_h, \qquad \langle\langle \bullet \rangle\rangle_{\mathcal{I}_\Gamma} = \sum_{I \in \mathcal{I}_\Gamma} \int_I \bullet \, dS,$$
$$\langle \bullet \rangle_\Omega = \sum_{E \in \Omega_e} \int_E \bullet \, d\boldsymbol{x}, \qquad \langle\langle \bullet \rangle\rangle_{\mathcal{I}_\Omega} = \sum_{I \in \mathcal{I}_\Omega} \int_I \bullet \, dA,$$

where $\langle \bullet \rangle$ and $\langle\langle \bullet \rangle\rangle$ denote the integral over elements and their interfaces, respectively; $\mathcal{I}_\Gamma$ and $\mathcal{I}_\Omega$ are the set of element interfaces (edges) in Γ_e and Ω_e.

In addition, we use the notation $\{\bullet\} = \frac{\bullet^- + \bullet^+}{2}$ and $[[\bullet]] = \frac{\bullet^- - \bullet^+}{2}$ for mean and jump operators, respectively, which are defined for general scalar a and vector $\boldsymbol{u}$ fields as:

$$\{a\} = \frac{1}{2}(a^- + a^+), \qquad \{\boldsymbol{u}\} = \frac{1}{2}(\boldsymbol{u}^- + \boldsymbol{u}^+),$$
$$[[a\boldsymbol{n}]] = a^-\boldsymbol{n}^- + a^+\boldsymbol{n}^+, \qquad [[\boldsymbol{u} \cdot \boldsymbol{n}]] = \boldsymbol{u}^- \cdot \boldsymbol{n}^- + \boldsymbol{u}^+ \cdot \boldsymbol{n}^+, \qquad [[\boldsymbol{u}\boldsymbol{n}]] = \boldsymbol{u}^-\boldsymbol{n}^- + \boldsymbol{u}^+\boldsymbol{n}^+,$$

where the variables on the negative and positive side of the interface are denoted by the superscripts $-$ and $+$, respectively, and $\boldsymbol{n}$ represents the outward unit normal vector at the interface.

4.2. Weak formulation of the 3D algorithm

4.2.1. Horizontal momentum equation

Multiplying the underlying equation (15) by a test function $\boldsymbol{\psi_u} \in \mathbb{V} \times \mathbb{V}$, and integrating over the 3D time-dependent domain Ω, the following weak formulation can be

3D non-hydrostatic algorithm

Field		Function space
Free surface elevation	η	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
Depth-averaged velocity	$\bar{\boldsymbol{u}}$	$\left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\right]^2$
Horizontal velocity	$\boldsymbol{u}$	$\left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_1^{\mathrm{DG}}(V_e)\right]^2$
Vertical velocity	w	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_1^{\mathrm{DG}}(V_e)$
Tracer (temperature & salinity)	c	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_1^{\mathrm{DG}}(V_e)$
3D Water density	ρ'	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_1^{\mathrm{DG}}(V_e)$
3D Baroclinic head	r	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_2(V_e)$
Baroclinic pressure gradient	$\boldsymbol{p}$	$\left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_1^{\mathrm{DG}}(V_e)\right]^2$
3D non-hydrostatic pressure	q	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\otimes\mathrm{P}_2(V_e)$

2D non-hydrostatic algorithm

Field		Function space
Free surface elevation	η	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
Depth-averaged velocity	$\bar{\boldsymbol{u}}$	$\left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\right]^2$
Layer-averaged velocity	$\bar{\boldsymbol{u}}_k$	$\left[\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)\right]^2$
Layer-averaged vertical velocity	$\bar{w}_k$	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
Layer-averaged tracer	$\bar{c}_k$	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
2D Water density	ρ_k'	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
2D Baroclinic head	r_k	$\mathrm{P}_1^{\mathrm{DG}}(\Gamma_e)$
2D non-hydrostatic pressure	q_{z_k}	$\mathrm{P}_2(\Gamma_e)$

Table 1: Summary of the function spaces used in the 3D and 2D non-hydrostatic algorithms. The superscript DG denotes a discontinuous function space.

obtained:

$$
\begin{aligned}
&\Big\langle \frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{\psi}_{\boldsymbol{u}} \Big\rangle_{\Omega} + \underbrace{\Big\langle\Big\langle (\boldsymbol{u}^{\mathrm{up}}\{\boldsymbol{u}\}) : [[\boldsymbol{\psi}_{\boldsymbol{u}} \boldsymbol{n}_h]] \Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle (\boldsymbol{u}\boldsymbol{u}) : \nabla_h \boldsymbol{\psi}_{\boldsymbol{u}} \Big\rangle_{\Omega}}_{\text{horizontal advection } \nabla_h\cdot(\boldsymbol{u}\boldsymbol{u})} \\
&\quad + \underbrace{\Big\langle\Big\langle \boldsymbol{u}^{\mathrm{up}}\{w - w_m\} \cdot [[\boldsymbol{\psi}_{\boldsymbol{u}} n_z]] \Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle \boldsymbol{u}\,(w - w_m) \cdot \frac{\partial \boldsymbol{\psi}_{\boldsymbol{u}}}{\partial z} \Big\rangle_{\Omega}}_{\text{vertical advection } \frac{\partial (w-w_m)\boldsymbol{u}}{\partial z}} + \underbrace{\Big\langle f \mathbf{e}_z \times \boldsymbol{u} \cdot \boldsymbol{\psi}_{\boldsymbol{u}} \Big\rangle_{\Omega}}_{\text{Coriolis } f\mathbf{e}_z\times\boldsymbol{u}} \\
&\quad + \underbrace{\Big\langle\Big\langle g\eta [[\boldsymbol{\psi} \cdot \boldsymbol{n}_h]] \Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle g\eta \nabla_h \cdot \boldsymbol{\psi}_{\boldsymbol{u}} \Big\rangle_{\Omega}}_{\text{3D elevation gradient } g\nabla_h\eta} + \underbrace{\Big\langle g\nabla_h r \cdot \boldsymbol{\psi}_{\boldsymbol{u}} \Big\rangle_{\Omega} + \Big\langle\Big\langle \gamma [[\boldsymbol{u}]] \cdot [[\boldsymbol{\psi}_{\boldsymbol{u}}]] \Big\rangle\Big\rangle_{\mathcal{I}_\Omega}}_{\text{baroclinic pressure gradient } g\nabla_h r} = \mathrm{Vis}_h^{\boldsymbol{u}} + \mathrm{Vis}_v^{\boldsymbol{u}},
\end{aligned}
\tag{60}
$$

with

$$
\begin{aligned}
\mathrm{Vis}_h^{\boldsymbol{u}} = &-\Big\langle \nu_h (\nabla_h \boldsymbol{u}) : (\nabla_h \boldsymbol{\psi}_{\boldsymbol{u}}) \Big\rangle_{\Omega} + \Big\langle\Big\langle \{\nu_h \nabla_h \boldsymbol{u}\} : [[\boldsymbol{\psi}_{\boldsymbol{u}} \boldsymbol{n}_h]] \Big\rangle\Big\rangle_{\mathcal{I}_\Omega} \\
&+ \Big\langle\Big\langle [[\boldsymbol{u}\boldsymbol{n}_h]] : \{\nu_h \nabla_h \boldsymbol{\psi}_{\boldsymbol{u}}\} \Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle\Big\langle \{\sigma\}\{\nu_h\}[[\boldsymbol{u}\boldsymbol{n}_h]] : [[\boldsymbol{\psi}_{\boldsymbol{u}} \boldsymbol{n}_h]] \Big\rangle\Big\rangle_{\mathcal{I}_\Omega},
\end{aligned}
\tag{61}
$$

$$
\begin{aligned}
\mathrm{Vis}_v^{\boldsymbol{u}} = &-\Big\langle \nu_v \frac{\partial \boldsymbol{u}}{\partial z} \cdot \frac{\partial \boldsymbol{\psi}_{\boldsymbol{u}}}{\partial z} \Big\rangle_{\Omega} + \Big\langle\Big\langle \{\nu_v \frac{\partial \boldsymbol{u}}{\partial z}\} \cdot [[\boldsymbol{\psi}_{\boldsymbol{u}} n_z]] \Big\rangle\Big\rangle_{\mathcal{I}_h} \\
&+ \Big\langle\Big\langle [[\boldsymbol{u} n_z]] \cdot \{\nu_v \frac{\partial \boldsymbol{\psi}_{\boldsymbol{u}}}{\partial z}\} \Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle\Big\langle \{\sigma\}\{\nu_v\}[[\boldsymbol{u} n_z]] \cdot [[\boldsymbol{\psi}_{\boldsymbol{u}} n_z]] \Big\rangle\Big\rangle_{\mathcal{I}_h},
\end{aligned}
\tag{62}
$$

where $\mathcal{I}_\Omega = \mathcal{I}_h \cup \mathcal{I}_v$, and $\mathcal{I}_h$ and $\mathcal{I}_v$ are the set of horizontal and vertical interfaces, respectively; the colon refers to the Frobenius inner product: $\boldsymbol{A} : \boldsymbol{B} = \sum_{i,j} A_{ij} B_{ij}$, and $\boldsymbol{u}^{\mathrm{up}}$ denotes the conventional upwind value at the interfaces. The extra Lax–Friedrichs flux is used to stabilise the baroclinic pressure gradient, with the parameter γ representing an upper bound for the fastest phase speed of internal waves (Blaise et al., 2010). $\mathrm{Vis}_h^{\boldsymbol{u}}$ and $\mathrm{Vis}_v^{\boldsymbol{u}}$ denote the weak forms of horizontal and vertical viscosity, respectively, which are based on the Symmetric Interior Penalty Galerkin (SIPG) formulation (Epshteyn and Rivière, 2007) with a new penalty factor σ introduced. More information on parameter choices may be found in (Kärnä et al., 2018).

For the sake of simplicity, terms associated with boundary conditions have been omitted in the presentation of the weak forms here.

4.2.2. Vertical momentum equation

Similarly, the vertical velocity is first computed from the vertical momentum equation (16) without the non-hydrostatic term. The following weak form is derived as above with the test function $\psi_w \in \mathbb{V}$:

$$\Big\langle \frac{\partial w}{\partial t}\psi_w \Big\rangle_\Omega + \underbrace{\Big\langle\Big\langle w^{\text{up}}\{\boldsymbol{u}\}\cdot[[\psi_w \boldsymbol{n}_h]]\Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle w\boldsymbol{u}\cdot\nabla_h\psi_w\Big\rangle_\Omega}_{\text{horizontal advection } \nabla_h\cdot(\boldsymbol{u}w)}$$
$$+ \underbrace{\Big\langle\Big\langle w^{\text{up}}\{w - w_m\}[[\psi_w n_z]]\Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle w\,(w - w_m)\,\frac{\partial \psi_w}{\partial z}\Big\rangle_\Omega}_{\text{vertical advection } \frac{\partial (w-w_m)w}{\partial z}} = \text{Vis}_h^w + \text{Vis}_v^w, \tag{63}$$

with

$$\begin{aligned}\text{Vis}_h^w = &-\Big\langle \nu_h(\nabla_h w)\cdot(\nabla_h\psi_w)\Big\rangle_\Omega + \Big\langle\Big\langle\{\nu_h\nabla_h w\}\cdot[[\psi_w\boldsymbol{n}_h]]\Big\rangle\Big\rangle_{\mathcal{I}_\Omega}\\ &+\Big\langle\Big\langle[[w\boldsymbol{n}_h]]\cdot\{\nu_h\nabla_h\psi_w\}\Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle\Big\langle\{\sigma\}\{\nu_h\}[[w\boldsymbol{n}_h]]\cdot[[\psi_w\boldsymbol{n}_h]]\Big\rangle\Big\rangle_{\mathcal{I}_\Omega},\end{aligned} \tag{64}$$

$$\begin{aligned}\text{Vis}_v^w = &-\Big\langle \nu_v\frac{\partial w}{\partial z}\frac{\partial \psi_w}{\partial z}\Big\rangle_\Omega + \Big\langle\Big\langle\{\nu_v\frac{\partial w}{\partial z}\}[[\psi_w n_z]]\Big\rangle\Big\rangle_{\mathcal{I}_h}\\ &+\Big\langle\Big\langle[[w n_z]]\{\nu_v\frac{\partial \psi_w}{\partial z}\}\Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle\Big\langle\{\sigma\}\{\nu_v\}[[w n_z]][[\psi_w n_z]]\Big\rangle\Big\rangle_{\mathcal{I}_h},\end{aligned} \tag{65}$$

4.2.3. Tracer equation

Analogously, the weak formulation of the tracer equation reads:

$$\Big\langle \frac{\partial c}{\partial t}\psi_c \Big\rangle_\Omega + \underbrace{\Big\langle\Big\langle c^{\text{up}}\{\boldsymbol{u}\}\cdot[[\psi_c \boldsymbol{n}_h]]\Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle c\boldsymbol{u}\cdot\nabla_h\psi_c\Big\rangle_\Omega}_{\text{horizontal advection } \nabla_h\cdot(\boldsymbol{u}c)}$$
$$+ \underbrace{\Big\langle\Big\langle c^{\text{up}}\{w - w_m\}[[\psi_c n_z]]\Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle c\,(w - w_m)\,\frac{\partial \psi_c}{\partial z}\Big\rangle_\Omega}_{\text{vertical advection } \frac{\partial (w-w_m)c}{\partial z}} = \text{Diff}_h + \text{Diff}_v, \tag{66}$$

where Diff_h and Diff_v also adopt the SIPG formulation to represent the horizontal and vertical diffusion. For brevity, the specific terms are not given since they are identical to (64)–(65) except with the use of a different test function $\psi_c \in \mathbb{V}$, as well as the horizontal and vertical diffusivities κ_h, κ_v replacing the viscosities ν_h and ν_v.

In this study, a linear equation of state (18) is assumed and used to compute density as a function of the tracer fields. The baroclinic head (12) is solved for using the following weak form:

$$\underbrace{\Big\langle\Big\langle r n_z\psi_r\Big\rangle\Big\rangle_{\partial\Omega_b} + \Big\langle\Big\langle r^{\text{up}}[[\psi_r n_z]]\Big\rangle\Big\rangle_{\mathcal{I}_h} - \Big\langle r\frac{\partial \psi_r}{\partial z}\Big\rangle_\Omega}_{\text{partial derivative } \partial r/\partial z} = \underbrace{\Big\langle -\frac{1}{\rho_0}\rho'\psi_r\Big\rangle_\Omega}_{\text{density } -\rho'/\rho_0}, \tag{67}$$

where $\partial\Omega_b$ denotes the bottom boundary of the domain Ω, and the term for the free surface boundary $\partial\Omega_s$ has been omitted since $r|_{z=\eta} = 0$.

When separately computing the baroclinic pressure gradient $\boldsymbol{p}$, we use the following weak form with the test function $\boldsymbol{\psi_p} \in \mathbb{V}\times\mathbb{V}$:

$$\Big\langle \boldsymbol{p}\cdot\boldsymbol{\psi_p}\Big\rangle_\Omega = \Big\langle\Big\langle gr\boldsymbol{\psi_p}\cdot\boldsymbol{n}_h\Big\rangle\Big\rangle_{\partial\Omega_b\cup\partial\Omega_s} + \Big\langle\Big\langle g\{r\}[[\boldsymbol{\psi_p}\cdot\boldsymbol{n}_h]]\Big\rangle\Big\rangle_{\mathcal{I}_\Omega} - \Big\langle gr\nabla_h\cdot\boldsymbol{\psi_p}\Big\rangle_\Omega. \tag{68}$$

4.2.4. 3D Poisson equation

The intermediate velocities $\hat{\boldsymbol{u}}^* = (\boldsymbol{u}^*, w^*)^T$ are solved for based on the derived weak forms (60) and (63), which are then substituted into the continuity equation (3), leading to the 3D Poisson equation (50) for the non-hydrostatic pressure q. The weak formulation of (50) is

$$\underbrace{\left\langle\!\left\langle \psi_q \nabla q \cdot \boldsymbol{n} \right\rangle\!\right\rangle_{\partial\Omega} - \left\langle \nabla q \cdot \nabla \psi_q \right\rangle_{\Omega}}_{\text{Laplace operator } \nabla^2 q} = \underbrace{\left\langle\!\left\langle \frac{\rho_0}{\Delta t} \psi_q \hat{\boldsymbol{u}}^* \cdot \boldsymbol{n} \right\rangle\!\right\rangle_{\partial\Omega} - \left\langle \frac{\rho_0}{\Delta t} \hat{\boldsymbol{u}}^* \cdot \nabla \psi_q \right\rangle_{\Omega}}_{\text{velocity source } \frac{\rho_0}{\Delta t}\nabla\cdot\hat{\boldsymbol{u}}^*}, \tag{69}$$

where $\partial\Omega$ denotes boundaries of the domain Ω, which includes the bottom $\partial\Omega_b$, free surface $\partial\Omega_s$ and the lateral boundaries $\partial\Omega_l$, i.e. $\partial\Omega = \partial\Omega_b \cup \partial\Omega_s \cup \partial\Omega_l$. No interface terms appear in the weak form since a continuous Galerkin discretisation is adopted for the Poisson equation.

It should also be noted that boundary terms $\langle\langle \bullet \rangle\rangle_{\partial\Omega}$ in (69) vanish for closed boundaries due to the zero Neumann and impermeability conditions, while in terms of open boundaries, e.g. at inflow boundaries and the free surface, $q = 0$ can be specified using the Dirichlet condition.

4.3. Weak formulation of the 2D algorithm

4.3.1. Hydrostatic equations

In the hydrostatic step, the momentum equations are solved without the non-hydrostatic terms. The depth-integrated horizontal momentum equation (32) is solved first in non-conservative form, coupled with the free surface equation (9). The corresponding weak formulation is obtained by multiplying each equation by an arbitrary test functions $\phi_\eta \in \mathbb{W}$ and $\boldsymbol{\phi_u} \in \mathbb{W}\times\mathbb{W}$ and performing integration by parts over each element in the 2D domain Γ and summing, to yield:

$$\left\langle \frac{\partial \eta}{\partial t} \phi_\eta \right\rangle_\Gamma + \underbrace{\left\langle\!\left\langle \{H\} \bar{\boldsymbol{u}}^{\mathrm{r}} \cdot [[\phi_\eta \boldsymbol{n}_h]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} - \left\langle H \bar{\boldsymbol{u}} \cdot \nabla_h \phi_\eta \right\rangle_\Gamma}_{\text{divergence } \nabla_H\cdot(H\bar{\boldsymbol{u}})} = 0, \tag{70}$$

$$\begin{aligned}\left\langle \frac{\partial \bar{\boldsymbol{u}}}{\partial t} \cdot \boldsymbol{\phi_u} \right\rangle_\Gamma &+ \underbrace{\left\langle\!\left\langle \{\bar{\boldsymbol{u}}\} \cdot [[\boldsymbol{\phi_u}(\bar{\boldsymbol{u}} \cdot \boldsymbol{n}_h)]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} - \left\langle \bar{\boldsymbol{u}} \cdot \nabla_h \cdot (\boldsymbol{\phi_u} \bar{\boldsymbol{u}}) \right\rangle_\Gamma}_{\text{2D advection } \bar{\boldsymbol{u}}\cdot\nabla_h\bar{\boldsymbol{u}}} + \underbrace{\left\langle f \boldsymbol{e}_z \times \bar{\boldsymbol{u}} \cdot \boldsymbol{\phi_u} \right\rangle_\Gamma}_{\text{2D Coriolis } f\boldsymbol{e}_z\times\bar{\boldsymbol{u}}} \\ &+ \underbrace{\left\langle\!\left\langle g \eta^{\mathrm{r}} [[\boldsymbol{\phi_u} \cdot \boldsymbol{n}_h]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} - \left\langle g \eta \nabla_h \cdot \boldsymbol{\phi_u} \right\rangle_\Gamma}_{\text{2D elevation gradient } g\nabla_h\eta} + \underbrace{\left\langle g \nabla_h \bar{r} \cdot \boldsymbol{\phi_u} \right\rangle_\Gamma}_{\text{2D baroclinity } g\nabla_h\bar{r}} = \mathrm{Vis}_h^{\bar{\boldsymbol{u}}},\end{aligned} \tag{71}$$

with

$$\begin{aligned}\mathrm{Vis}_h^{\bar{\boldsymbol{u}}} = &-\left\langle \nu_h (\nabla_h \bar{\boldsymbol{u}}) : (\nabla_h \boldsymbol{\phi_u})^T \right\rangle_\Gamma + \left\langle \frac{\nu_h}{H} (\nabla_h H) \cdot \nabla_h \bar{\boldsymbol{u}} \cdot \boldsymbol{\phi_u} \right\rangle_\Gamma + \left\langle\!\left\langle \{\nu_h \nabla_h \bar{\boldsymbol{u}}\} : [[\boldsymbol{\phi_u} \boldsymbol{n}_h]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} \\ &+ \left\langle\!\left\langle [[\bar{\boldsymbol{u}} \boldsymbol{n}_h]] : \{\nu_h \nabla_h \boldsymbol{\phi_u}\} \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} + \left\langle\!\left\langle \{\sigma\}\{\nu_h\} [[\bar{\boldsymbol{u}} \boldsymbol{n}_h]] \cdot [[\boldsymbol{\phi_u} \boldsymbol{n}_h]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma},\end{aligned} \tag{72}$$

where the starred variables are based on an approximate Riemann solver; here the linear Roe solution (Comblen et al., 2010) is used:

$$\eta^{\mathrm{r}} = \{\eta\} + \sqrt{\frac{\{H\}}{g}} [[\bar{\boldsymbol{u}} \cdot \boldsymbol{n}_h]], \qquad \bar{\boldsymbol{u}}^{\mathrm{r}} = \{\bar{\boldsymbol{u}}\} + \sqrt{\frac{g}{\{H\}}} [[\eta \boldsymbol{n}_h]]. \tag{73}$$

The layer-averaged velocities $\bar{\boldsymbol{u}}_k$ are then solved for using the known value of η^*, based on the weak form:

$$\left\langle \frac{\partial \bar{\boldsymbol{u}}_k}{\partial t} \cdot \boldsymbol{\phi}_{\boldsymbol{u}_k} \right\rangle_\Gamma + \underbrace{\left\langle\!\left\langle \{\bar{\boldsymbol{u}}_k\} \cdot [[\boldsymbol{\phi}_{\boldsymbol{u}_k} (\bar{\boldsymbol{u}}_k \cdot \boldsymbol{n}_h)]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma} - \left\langle \bar{\boldsymbol{u}}_k \cdot \nabla_h \cdot (\boldsymbol{\phi}_{\boldsymbol{u}_k} \bar{\boldsymbol{u}}_k) \right\rangle_\Gamma}_{\text{layered advection } \bar{\boldsymbol{u}}_k \cdot \nabla_h \bar{\boldsymbol{u}}_k}$$

$$= \underbrace{\left\langle -g\nabla_h \eta^* \cdot \boldsymbol{\phi}_{\boldsymbol{u}_k} + F^n_{\bar{\boldsymbol{u}}_k} \cdot \boldsymbol{\phi}_{\boldsymbol{u}_k} \right\rangle_\Gamma}_{\text{explicit layered terms } -g\nabla_h\eta^* + F^n_{\bar{\boldsymbol{u}}_k}}, \qquad \forall k \in [1, N-1]. \tag{74}$$

The vertical velocities $\bar{w}_k$ and tracers $\bar{c}_k$ are obtained analogously by solving (24) and (25), respectively.

4.3.2. 2D Poisson equation

In the non-hydrostatic step, substitution of the corrected velocity fields (56)–(57) into the layered continuity equations (30) gives the layer-based Poisson equation for the non-hydrostatic pressure $\boldsymbol{q} = [q_{z_0}, q_{z_1}, \ldots, q_{z_{N-1}}]^T$. With the test function $\boldsymbol{\phi_q} \in \mathbb{W}^N$ the weak formulation then reads

$$\left\langle \boldsymbol{A}\nabla_h^2 \boldsymbol{q} \cdot \boldsymbol{\phi_q} \right\rangle_\Gamma + \left\langle \boldsymbol{B}\nabla_h \boldsymbol{q} \cdot \boldsymbol{\phi_q} \right\rangle_\Gamma + \left\langle (\boldsymbol{C}\boldsymbol{q} + \boldsymbol{D}) \cdot \boldsymbol{\phi_q} \right\rangle_\Gamma = 0,$$

$$\Downarrow$$

$$\left\langle\!\left\langle (\boldsymbol{A}\nabla_h \boldsymbol{q}) : \left(\boldsymbol{\phi_q} \boldsymbol{n}_h\right) \right\rangle\!\right\rangle_{\partial\Gamma} - \left\langle (\nabla_h \boldsymbol{A})\nabla_h \boldsymbol{q} \cdot \boldsymbol{\phi_q} + (\boldsymbol{A}\nabla_h \boldsymbol{q}) : \nabla_h \boldsymbol{\phi_q} \right\rangle_\Gamma$$

$$+ \left\langle\!\left\langle (\boldsymbol{B}\boldsymbol{q}) : \left(\boldsymbol{\phi_q} \boldsymbol{n}_h\right) \right\rangle\!\right\rangle_{\partial\Gamma} - \left\langle (\nabla_h \cdot \boldsymbol{B})\boldsymbol{q} \cdot \boldsymbol{\phi_q} + (\boldsymbol{B}\boldsymbol{q}) : \nabla_h \boldsymbol{\phi_q} \right\rangle_\Gamma + \left\langle (\boldsymbol{C}\boldsymbol{q} + \boldsymbol{D}) \cdot \boldsymbol{\phi_q} \right\rangle_\Gamma = 0, \tag{75}$$

where boundary terms $\langle\langle \bullet \rangle\rangle_{\partial\Gamma}$ are treated in a similar manner to (69).

4.4. Implementation in Firedrake

In this the weak forms given in the previous sections are specified using the Unified Form Language (Alnæs et al., 2014), in which the numerical model can be written in a concise form very close to the underlying mathematical presentation above. The discrete equations based upon the weak formulations are then assembled using low level C code which is generated at run time using the Firedrake finite element framework (Rathgeber et al., 2016); in turn the corresponding linear/nonlinear systems are solved using the PETSc library (Balay et al., 2019).

For more details about the approach taken interested readers can refer to the previous multi-layer work of Pan et al. (2019) and an application of Firerake to fluid dynamics modelling (Jacobs and Piggott, 2015). The solution method in Firedrake is also briefly illustrated in AppendixA.

In *Thetis* the 3D non-hydrostatic algorithm adopts the linear discontinuous Galerkin discretisation, $\mathrm{P}_1^{\mathrm{DG}}$, with vertex-based slope limiting (Kuzmin, 2010) utilised for 3D fields to help avoid any under/over-shoots. In terms of the discretisation under the 2D non-hydrostatic algorithm, three choices of basis functions are supported for the element pair representing $\bar{\boldsymbol{u}}$ and η, including equal order DG (DG–DG), Raviart-Thomas DG (RT–DG) and $\mathrm{P}_1^{\mathrm{DG}}$–$\mathrm{P}_2$. Except $\mathrm{P}_1^{\mathrm{DG}}$–$\mathrm{P}_2$, the other two choices are currently applicable to degrees $p \leq 2$. For the test cases to follow, the $\mathrm{P}_1^{\mathrm{DG}}$discretisation is utilised in order to most fairly compare the 3D and 2D algorithms.

5. Test cases

In this section the non-hydrostatic algorithms introduced above are validated and compared using a suite of test cases. These involve a series of different wave processes, including dispersive surface waves and nonlinear internal waves.

5.1. *Surface standing wave in a deep basin*

In this barotropic standing wave test case highly dispersive waves are simulated to test the model's accuracy in computing wave celerity and frequency dispersion. The domain is a vertical slice (i.e. resolved in x-z space while only being a single element in the y direction), closed at either end with a length of $L = 20$ m and deep water depth of $d = 80$ m (so that $kd = 8\pi$ where k denotes the wave number). Initially, the velocities are set to zero and the surface elevation is specified by $\eta_0 = a\cos(2\pi x/L)$. Here a small wave amplitude $a = 0.1$ m is used, and hence the standing wave dynamics can be considered linear with the expected wave celerity of $c = 5.59$ m/s based on the linear dispersion relation.

To compare the proposed 3D and 2D based non-hydrostatic algorithms, identical horizontal mesh resolution and time step are adopted. In this case, the computational domain is discretised with a split-quad mesh using 40 triangles with an element edge length of 0.5 m in the long x direction and a single element in y. The total simulation time is 20 s with the time step taken as 0.01 s. All boundaries are impermeable, and no explicit viscosity or bottom friction are included.

Fig. 2 shows comparisons of the surface elevation evolution in time at location $x = 17.5$ m between the analytical solution and numerical results from the both 3D and 2D based non-hydrostatic algorithms with three vertical layers. Since the 2D algorithm can be considered similar to the use of a piecewise-constant (i.e. $\mathrm{P}_0^{\mathrm{DG}}$) discretisation of the layer-averaged variables in the vertical direction, both $\mathrm{P}_0^{\mathrm{DG}}$ and $\mathrm{P}_1^{\mathrm{DG}}$are applied to the vertical discretisation of the 3D algorithm in this case (with the same horizontal discretisation $\mathrm{P}_1^{\mathrm{DG}}$) in order to provide a fair comparison against the 2D algorithm. The time series of elevation computed using the non-linear shallow water equations is also displayed to indicate that the non-hydrostatic extensions improve results significantly since they admit frequency dispersion. For such deep water scenarios ($kd = 8\pi$), the 2D algorithm is able to represent the frequency dispersion correctly using only three layers, while the results from the 3D algorithm (with $\mathrm{P}_0^{\mathrm{DG}}$ in the vertical) using the same number of layers display clear phase deviation compared with the analytical solution, and the results from the 3D algorithm (even with higher-order $\mathrm{P}_1^{\mathrm{DG}}$discretisation in the vertical) show clear improvement however with discrepancies due to wave decay. With the vertical $\mathrm{P}_0^{\mathrm{DG}}$ discretisation, the 3D algorithm requires more vertical layers (six layers in this case) in order to satisfactorily represent dispersion.

For various numbers of vertical layers ($1-10$), the L_2 error of the surface elevation is computed at the end of the simulation based upon comparing the numerical results with the exact solution from linear wave theory. Fig. 3a indicates that approximately second-order convergence (slope: -2.16) in the elevation field is achieved with the 2D algorithm for smaller numbers of vertical layers, while the 3D algorithm even with use of a $\mathrm{P}_1^{\mathrm{DG}}$discretisation in the vertical displays convergence below second-order with a slope of around -1.53, thus requiring far more layers to reach the same level of accuracy. It should also be noted that the convergence results discussed here are based upon use of the same horizontal resolution and time step, with only the number of vertical layers varying. Convergence studies associated with varying horizontal mesh resolutions have been performed previously for the hydrostatic form of *Thetis* (Kärnä et al., 2018) and the multi-layer non-hydrostatic form (Pan et al., 2019), indicating approximately second-order convergence for velocity and a convergence rate of around 3.2 for the elevation, respectively. The higher than expected convergence in the vertical for the 2D algorithm may originate from the underlying $\mathrm{P}_1^{\mathrm{DG}}$based shallow water solver, with multiple such solves utilised in the 2D algorithm. The saturation of the error for the 2D algorithm for higher numbers of layers can also be seen in Fig. 3a. This originates from the contribution to the overall error from the time step and the horizontal element size choices.

Fig. 3b demonstrates that for this test case the 2D algorithm has a lower CPU cost (based on a single Intel Xeon E5-1650 processing core) but at a more rapidly increasing rate compared to that of the 3D algorithm. Overall, this error vs cost information indicates that the 2D algorithm is likely preferable for surface wave simulations, where only a relatively small number of layers are generally required to represent dispersion accurately. In contrast, for the simulation of stratified flows and mass transport, the 3D algorithm is likely to be more advantageous since a relatively high vertical resolution is required to appropriately represent velocity and density fields. This will be shown further in later test cases 5.4 and 5.5.

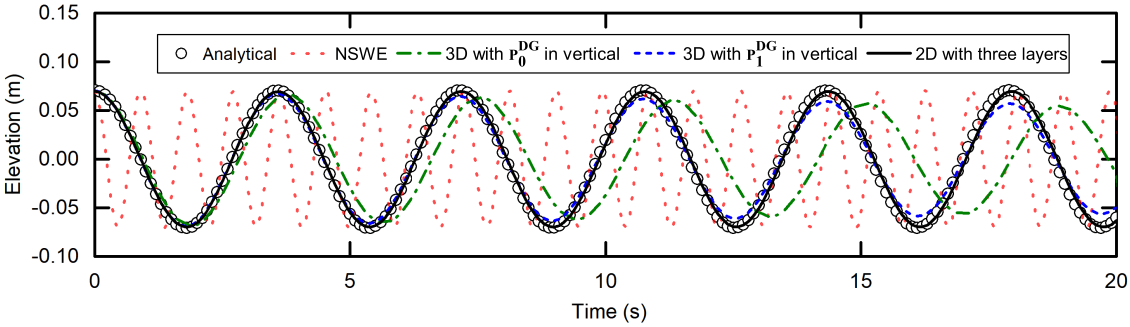

Figure 2: Comparisons of the wave elevation time-series at location $x = 17.5$ m for the surface standing wave in a deep closed basin. The analytical solution (circles) and numerical results from the hydrostatic shallow water equations (red dotted line), the non-hydrostatic 3D based algorithm with three layers adopting vertical P_0^{DG} (green dashed-dotted) and P_1^{DG}(blue dashed) discretisations, as well as the 2D based algorithm with three layers (black solid) are presented.

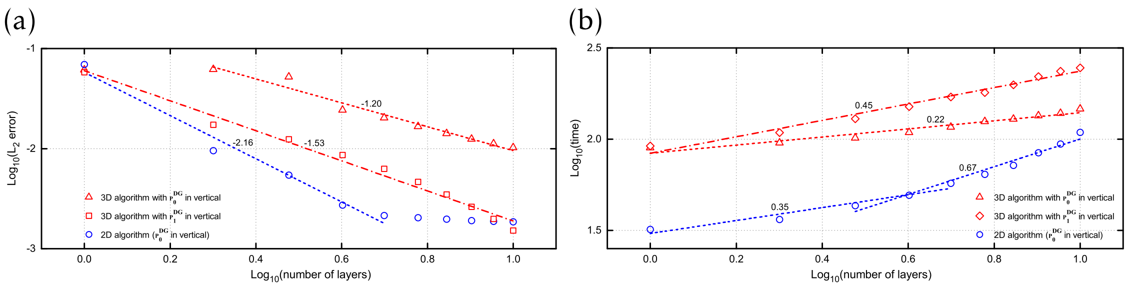

Figure 3: Convergence of the L_2 error in the free surface elevation at the end of the simulation (a), and the corresponding CPU times (b) with increasing number of vertical layers (1 – 10) in the standing wave case for both 3D (with P_0^{DG} and P_1^{DG}discretisation in the vertical) and 2D based non-hydrostatic algorithms. The slopes of the local least-squares fitted linear lines (red: 3D; blue: 2D) are also given in the plots.

5.2. Surface wave shoaling over a submerged bar

This test case is based upon the Beji and Battjes (1994) laboratory experiment in a wave flume. Regular waves propagate over a submerged bar and interact with the uneven bottom, leading to non-linear wave shoaling and significant wave transformation. In the process, higher harmonics appear on the upward slope of the bar, which past the bar are released as free waves with various speeds. In the numerical representation of this problem, good frequency dispersion accuracy is required.

A sketch of the bottom geometry and model setup are shown in Fig. 4. The flume is 30 m long and the still water depth is 0.4 m, which is reduced to a minimum of 0.1 m above the bar. The upward slope of the bar is 1:20 and the downward slope is 1:10. At the left-hand boundary, periodic regular waves with an amplitude of 1 cm and a period of 2.02 s are introduced. Here as addition 5 m sponge layer is placed at the right-hand end of the domain to absorb reflected waves. The whole computational domain is thus 35 m and is discretised with a split-quad mesh using 700 triangles with an element edge length of 0.1 m in the long x direction and a single element in y. In the z direction, three vertical layers are used. The time step is chosen as 0.01 s, and the total simulation time is 40 s. Again, no explicit viscosity or bottom friction are included.

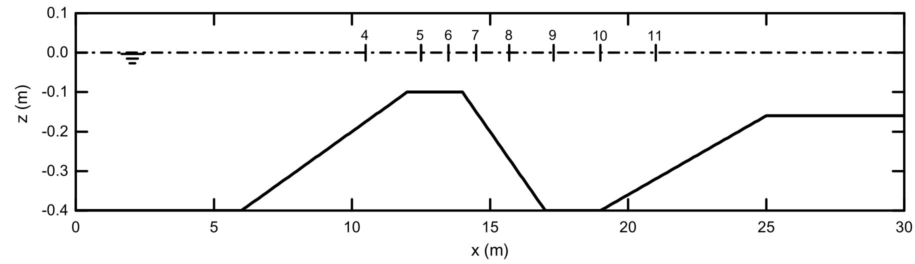

Figure 4: Bottom geometry and the locations of wave gauges 4 – 11, located at x = 10.5, 12.5, 13.5, 14.5, 15.7, 17.3, 19 and 21 m respectively, for the wave shoaling over a submerged bar test case. Based upon Beji and Battjes (1994).

Fig. 5 compares the simulated free surface elevation at eight gauges where measured data from the laboratory experiment is available. The time-series displayed indicate that the wave dynamics at all gauges are well represented by the both 3D and the 2D non-hydrostatic algorithms using three vertical layers, although the peak elevation at gauge 5 – 6 is slightly over-predicted by the 3D algorithm. The proposed algorithms correctly describe the shoaling phenomenon on the upward slope of the bar (gauge 4), and resolve the high-frequency dispersion at gauge 5 – 8 properly. This indicates that for such a surface wave case with respect to strong non-linear shoaling, relatively few vertical layers (e.g. three) is sufficient to give much improved dispersion capability compared to the hydrostatic model which can be seen to generate completely unrealistic waveforms in Fig. 5. It should be noted that the total CPU time for the 2D algorithm for this case is 11 min 1.32 s, which is significantly lower than the 23 min 37.93 s required for the 3D algorithm and is even still less than that in the 3D hydrostatic simulation (17 min 51.31 s), where they all use the same time step size of 0.01 s. Note further that the 2D algorithm can run stably for this problem with a time step 0.05 s and achieve nearly identical results (Pan et al., 2019) whereas the 3D algorithm fails due to numerical instability.

In addition, to demonstrate the model's capability to utilise multi-resolution meshes, Fig. 6 compares the surface elevation time-series at eight gauges between experimental data and numerical results from the both 3D and the 2D non-hydrostatic algorithms utilising multi-resolution meshes. A higher mesh resolution with an element edge length of 0.1 m is used around the submerged bar region ($x \in [10, 22]$ m), while a lower resolution with a larger element edge length (0.5 m) is adopted in the rest of the domain. As the time-series in Fig. 6 indicate both algorithms using this multi-resolution mesh represent the wave dynamics fairly well. A computational cost reduction of 24.8% and 33.2% is achieved for the 3D and 2D algorithm, respectively, compared to that of the simulations using uniform meshes. It should also be noted that the 3D algorithm displays a certain amount of wave dissipation, which can be improved by decreasing the time step size, increasing the submerged bar region, or slightly increasing the resolution in the coarse region.

5.3. Surface solitary wave generation by a 3D landslide

Submarine landslides represent important mechanism for tsunami wave generation. The subsequent free surface waves that are generated are often dispersive (Hill et al., 2014) and thus difficult to model accurately using hydrostatic models. In this test case,

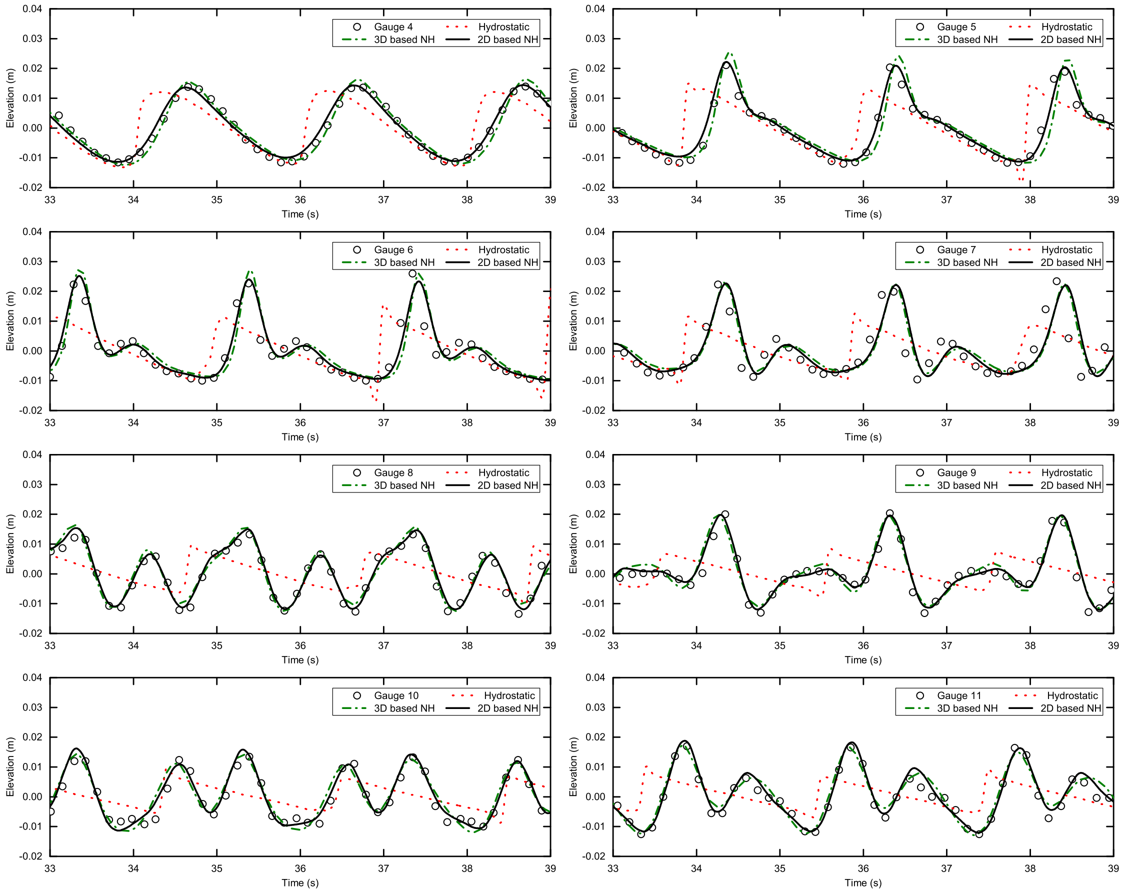

Figure 5: Comparisons between numerical and experimental surface elevation time series at eight wave gauges for the wave shoaling over a submerged bar test case. Laboratory data (circles) are compared against the hydrostatic numerical results (red dotted line) and non-hydrostatic numerical results from the 3D (green dash-dot) and the 2D (black solid) based algorithms, all using three vertical layers.

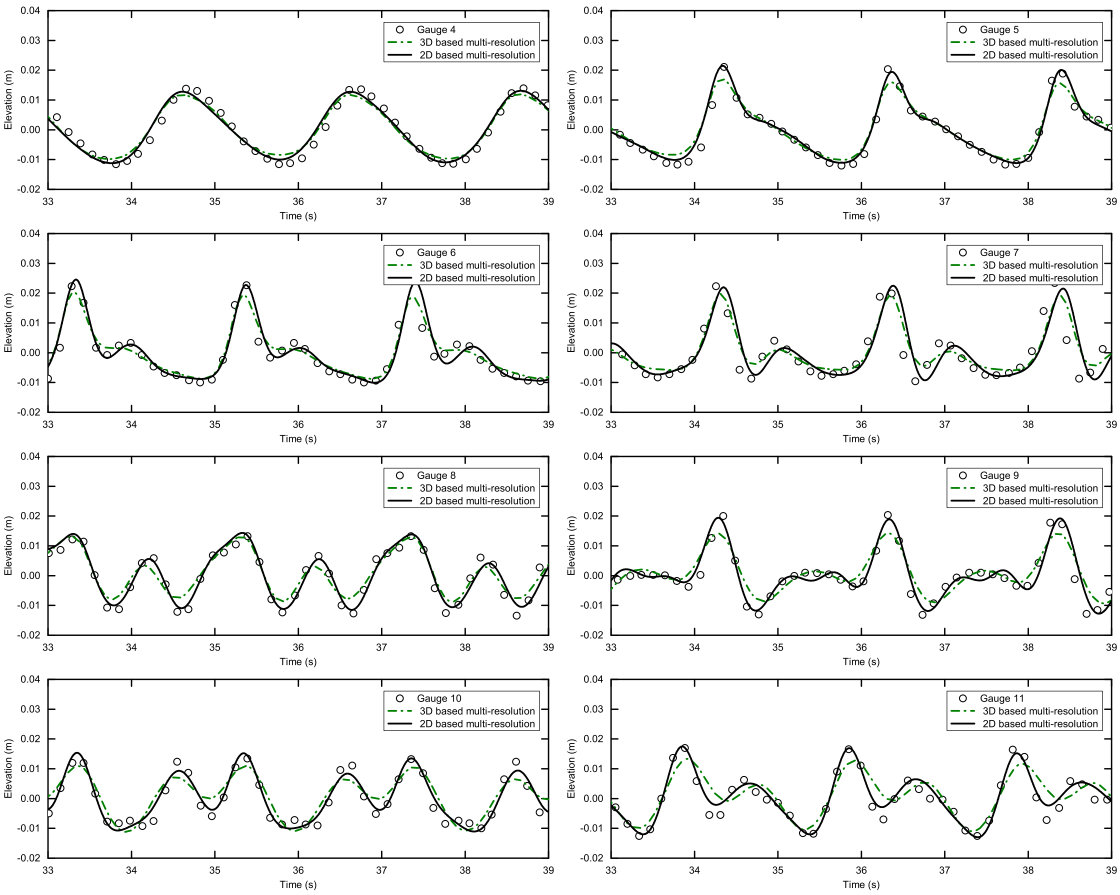

Figure 6: Comparisons of surface elevation time-series at eight gauges between experimental and numerical results using multi-resolution meshes for the submerged bar test case. Laboratory data (circles) are compared against the non-hydrostatic numerical results from the 3D (green dash-dot) and the 2D (black solid) based algorithms, all using three vertical layers. A higher mesh resolution with an element edge length of 0.1 m is used around the submerged bar region ($x \in [10, 22]$ m), while a lower resolution (0.5 m element edge length) is adopted in the rest of the domain.

the proposed non-hydrostatic algorithms are used to simulate surface solitary wave generation by an idealised 3D submarine landslide, based on a wave tank experiments performed by Enet and Grilli (2007).

The geometry of the tank is 30 m long, 3.7 m wide and 1.8 m deep with a plane slope of $\theta = 15°$. The vertical cross section of the landslide is sketched in Fig. 7, and the plan view of the domain and the landslide is indicated by the frames in Fig. 9. The slide shape is defined by the expression

$$\zeta = \frac{T}{1-\epsilon}\left[\mathrm{sech}(k_b x)\mathrm{sech}(k_w y) - \epsilon\right], \tag{76}$$

where $k_b = 2C/b$, $k_w = 2C/w$, $C = \mathrm{arccosh}(1/\epsilon)$, and truncation parameter $\epsilon = 0.717$. The centre of the slide is initially located at a submergence depth of $d = 61$ mm, and has a length of $b = 0.395$ m, a width of $w = 0.680$ m, and a thickness of $T = 0.082$ m. Its displacement down the slope is prescribed as

$$s(t) = s_0 \ln\left(\cosh\frac{t}{t_0}\right), \tag{77}$$

where $s_0 = u_t^2/a_0$, $t_0 = u_t/a_0$, with u_t and a_0 denoting the slide terminal velocity and initial acceleration, respectively. Here $u_t = 1.70$ m/s and a_0 =1.12 m/s^2.

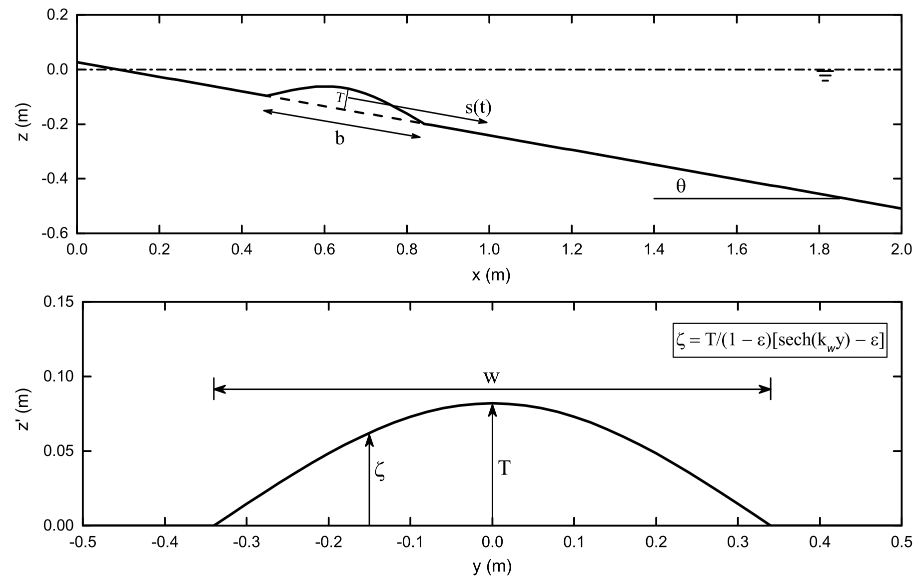

Figure 7: Vertical cross section of the landslide with length $b = 0.395$ m, width $w = 0.680$ m and thickness $T = 0.082$ m. The plane slope has an incline of $\theta = 15°$. Note that the second frame uses a local coordinate system with the z' oriented upward and perpendicular to the slope.

Due to the symmetry in this problem in the y direction, only half of the domain is included in the computational domain, and due to the limited length of simulation time only the leading portion in the x direction is considered. The computational domain in the horizontal is thus set as 6 m long and 1.8 m wide; it is discretised using 150 elements in the x direction and 36 elements in the y direction, with $\Delta x = 0.04$ m and $\Delta y = 0.05$ m. Three vertical layers are employed to capture dispersion of the generated short waves. The simulation is run for 4 s with a time step size of 0.005 s. All boundaries are impermeable, and at the left end wetting and drying is expected to occur. Again, no explicit bottom friction or viscosity is considered.

In Fig. 8, the numerical results for the generated wave elevations are compared with the measured data at three gauges whose locations are given in the caption. Good agreement is observed for both the 3D and 2D non-hydrostatic algorithms indicating that the wave generation mechanism has been implemented correctly, and the subsequent dispersive wave propagation is represented accurately. Although the wave train from the 3D algorithm seems less dissipative in the latter propagation, the main peaks and overall phase structure agree well with that obtained using the 2D non-hydrostatic algorithm. Additionally, the results from the 3D hydrostatic simulation are also displayed in Fig. 8. In contrast to the non-hydrostatic results which consisted of a series of dispersive waves lagging behind the relatively faster moving sliding mass, the hydrostatic result essentially show a strong immediate drawdown of the water column above the moving slide, and as would be expected, the drawdown diminishes as the submergence depth increases. These results are further illustrated with the snapshots of the wave elevations shown in Fig. 9. The similarity between panels (a) and (b) show that there is good agreement between the 3D and 2D non-hydrostatic algorithms, while panel (c) demonstrated the lack of a generated wave train with the hydrostatic algorithm. These results demonstrate the importance of representing frequency dispersion for accurate assessment of landslide-tsunami in certain parameter regimes.

Using four processing cores, the wall clock time for the 2D non-hydrostatic algorithm is approximately 76 min, which is approximately 64% of the time required for the 3D algorithm. Compared to running in serial mode, the parallel run of the 3D and 2D algorithms were roughly 3.17 and 2.95 times faster, respectively, showing that the scaling efficiency does not drop significantly despite the relatively small degree of freedom counts considered in these problems. For a parallel scaling analysis of larger test cases, in the case of the hydrostatic version of the model, see (Kärnä et al., 2018).

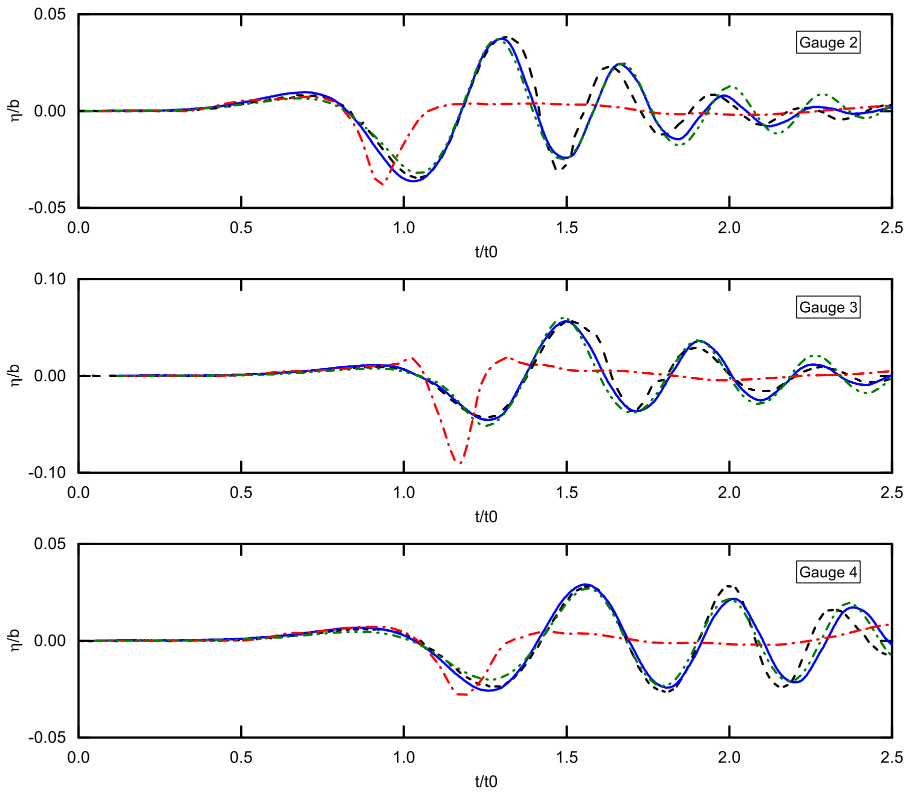

Figure 8: Comparisons of surface elevation time-series for landslide-generated waves at three gauges. Measured data (black dashed lines) is compared to the hydrostatic numerical results (red dash-dot) as well as non-hydrostatic numerical results (with the 3D algorithm: green dash-dot-dot; 2D algorithm: blue solid). The (x, y) locations of gauges 2 – 4 from the experiment are given by (1.469, 0.35) m, (1.929, 0) m, and (1.929, 0.5) m, respectively.

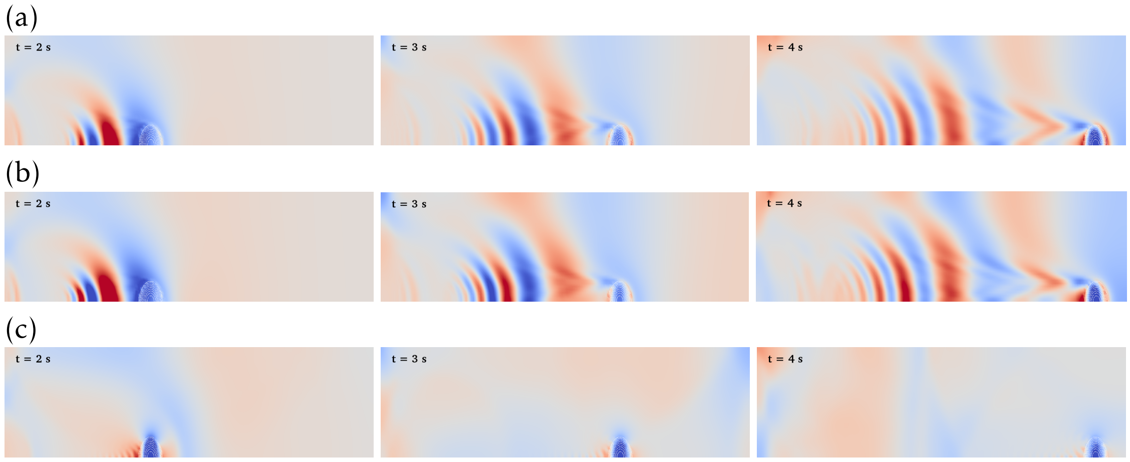

Figure 9: Two-dimensional plan view ((x, y)-coordinate) snapshots of landslide-generated wave patterns at $t = 2$, 3 and 4 s (plots from left to right). Top panel (a): non-hydrostatic with the 3D algorithm, middle (b): non-hydrostatic with the 2D algorithm, and bottom (c): hydrostatic simulation. The white lines are the contours of the moving slide. The colour scale for the free surface elevation range ranges between -0.012 $\sim$ 0.012 (blue to red) for all frames.

5.4. Internal solitary wave generation and propagation in exchange flow

The non-hydrostatic algorithms are tested against an exchange flow test case where an initial tilt of the interface between two water masses with different densities is allowed to evolve under the effects of gravity. The case is based upon the laboratory experiment performed by Horn et al. (2001) in a rectangular basin, where the density difference (20 kg/m^3) between the light and heavy water masses results in a buoyancy difference of $g' = g\Delta\rho/\rho_0 = 0.2$ m/s^2. With nonlinear steepening balanced by dispersive effects (Kanarska et al., 2007), internal solitary waves are generated and propagate along the interface. Note that a similar case where over-turning Kelvin-Helmholtz billows are generated is considered below.

The basin has a length of $L = 6$ m, a depth of $H = 0.29$ m, and a width of $W = 0.3$ m (Fig. 10). The amplitude of the initial tilt of the interface is η_0, and the depth of the lower layer is h_1. In the original experiment a series of seiche cases are considered with various dimensionless amplitude $\gamma = \eta_0/h_1$ and $\delta = h_1/H$ values. In this paper we only consider the case $\gamma = 0.3$ and $\delta = 0.9$ for brevity, which was also studied by Kanarska et al. (2007).

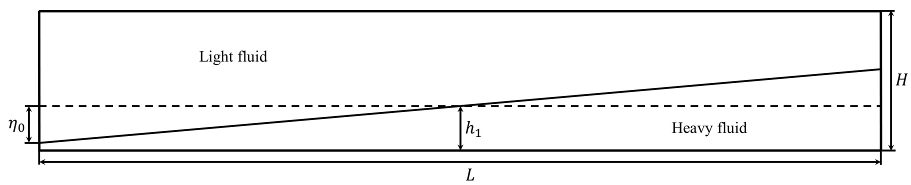

Figure 10: Numerical and experimental set-up for the case of internal solitary wave generation and propagation test case.

In the numerical set-up, to avoid an abrupt transition a pycnocline thickness of 0.015 m is assumed in order to smooth the interface. The computational domain is discretised using 400 elements with an edge length of 0.015 m in the x direction and only a single element in y. In terms of vertical discretisation, 80 equidistant σ layers are adopted. The total simulation time is 250 s with time step size of 0.01 s. All lateral boundaries are closed with no-normal flow conditions. Constant viscosity and diffusivity values are applied, using molecular values 10^{-6} m^2/s and 10^{-7} m^2/s, respectively. To improve numerical stability, vertex-based slope limiters (Kuzmin, 2010) are used for both velocity and tracer fields in this case.

Fig. 11 displays frames showing the degeneration of interfacial waves into a train of solitary waves in the numerical simulation. Qualitative agreement between the density evolution from the 3D based non-hydrostatic algorithm and the experiment (Horn et al., 2001) can also be observed. A time series of the interface displacement at the centre of the tank is shown in Fig. 12 where good agreement with laboratory data can be observed. This includes the correct propagation of the generated soliton train and the emergence of the seventh soliton after reflection. Since the 2D non-hydrostatic algorithm (layer-averaged) can be considered as a special form of a σ formulation with a low-order P_0^{DG} discretisation in the vertical direction, the leading soliton suffers from numerically diffusion at the same resolution (Fig. 13a). The 2D non-hydrostatic algorithm cannot describe the high-frequency solitary wave train generation process as well as the 3D non-hydrostatic algorithm, with higher resolution required to improve the result. In addition, as demonstrated by Vitousek and Fringer (2013), the implicitness parameter θ can affect the stability and consistency of non-hydrostatic models which utilise a first-order pressure projection method. In this case $\theta = 0.5$ is applied and the truncation errors of both discretisation algorithms are formally second-order, but it has to be acknowledged that the solution error might be different which could affect overall accuracy (Vitousek and Fringer, 2013). The different time integration techniques used for the 3D and 2D algorithms might also be the cause of the differences observed in terms of density distribution in baroclinic flows. Fig. 13b also shows that the process cannot be represented correctly in a hydrostatic simulation.

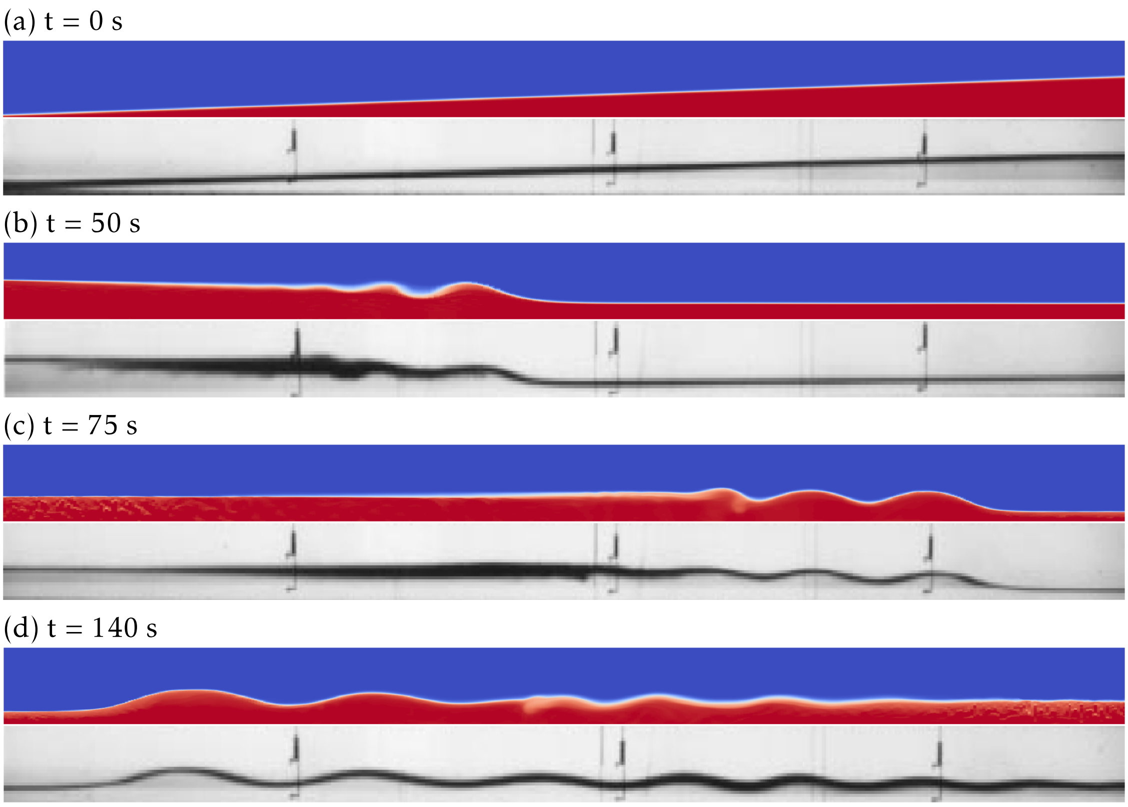

Figure 11: Comparisons between numerical (the 3D based non-hydrostatic algorithm) and experimental (Horn et al., 2001) density evolution in the internal soliton generation and propagation case at times $t = 0$, 50, 75 and 140 s. The scale of the density range is $0 \sim 20$ indicated by blue to red colours.

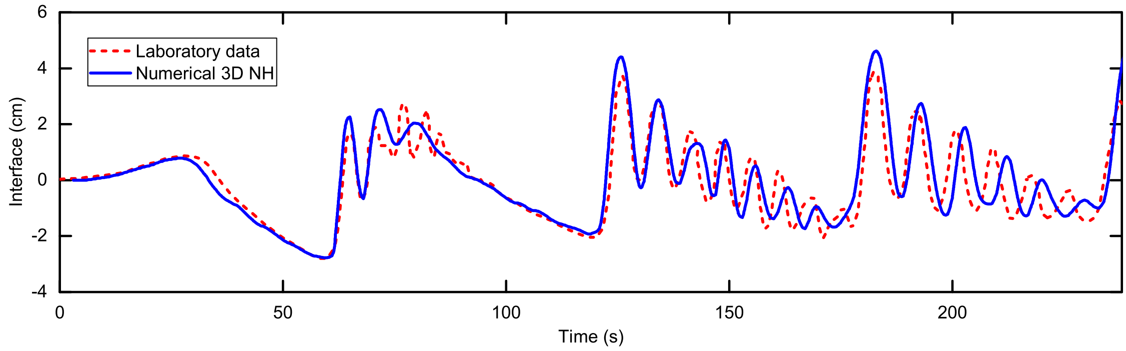

Figure 12: Time series of the interface displacement at the centre of the basin ($x = 0$) for the internal seiche case. Laboratory data (red dashed line) is compared to the non-hydrostatic numerical results with the 3D algorithm (blue solid).

(a)

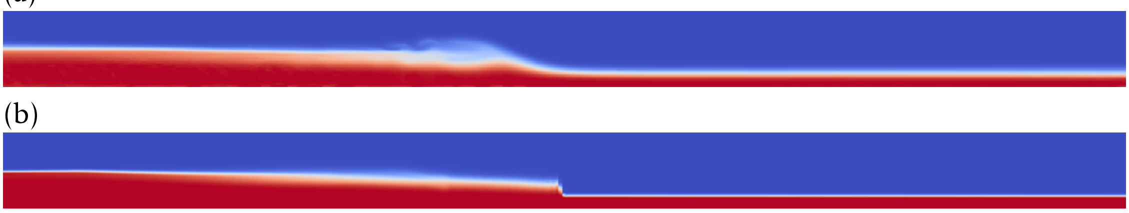

Figure 13: Density distribution at time $t = 50$ s in the internal seiche case for the 2D based non-hydrostatic algorithm (a), and a hydrostatic simulation (b). The scale of density range is $0 \sim 20$ shown by blue to red colours. Compare with the corresponding time level from Fig. 11.

5.5. *Non-hydrostatic lock exchange*

To further evaluate the validity of the baroclinic solver with the non-hydrostatic pressure correction, we adopt another exchange flow case that has been considerd in several other non-hydrostatic studies (Härtel et al., 2000; Fringer et al., 2006; Lai et al., 2010; Ma et al., 2013; Hiester et al., 2011). The model setup follows that considered in a direct numerical simulation (DNS) study (Härtel et al., 2000). Lighter fluid with zero salinity initially occupies the left hand side of the domain, while the right hand side is filled with heavier fluid with a salinity of $S = 1.3592$ psu. The two water masses are initially separated by a vertical barrier in the centre, which is a key difference from the small tilt considered in the previous test case. The constant mean density ρ_0 is 999.972 kg/m^3, and a linear equation of state (18) is used with parameters $\beta_S = 0.75 \times 10^{-3}\rho_0$ psu^{-1} and $\alpha_T = 0$ K^{-1}, giving a reduced gravity of $g' = g\Delta\rho/\rho_0 = 0.01$ m/s^2.

The domain is a vertical slice with a length of 0.8 m and a depth of 0.1 m. The thin horizontal domain is taken to be one element wide and discretised by a split-quad mesh of 800 triangular elements (0.002 m edge length) in the x direction with a single element in y. and 100 equidistant layers employed in the vertical. The total simulation runs for 30 s with a time step size of 0.01 s. All boundaries are closed with no-normal flow conditions. The viscosity and diffusivity are set to a constant value of 0.5×10^{-6} m^2/s in both horizontal and vertical directions. The vertex-based slope limiters are used in this case to reduce overshoots in tracer and velocity fields.

In addition to the hydrostatic algorithm, both the 3D and 2D non-hydrostatic algorithms are employed in this case with the same setup. Fig. 14 compares the resulting density fields at times $t = 10$ and 15 s. As can be observed and as expected, the generation of Kelvin-Helmholtz (KH) instabilities does not occur in the hydrostatic simulation while the KH billows appear in both non-hydrostatic results. Furthermore, the 3D non-hydrostatic algorithm (Fig. 14a) better captures the lock exchange flow and corresponding KH billows than the 2D one (Fig. 14b). Due in part to the higher numerical diffusion from the layer-averaged formulation with the vertical P_0^{DG} property, the 2D algorithm represents less billows. Higher vertical resolution (i.e. more vertical layers) can improve the density distribution and admit more billows, but as expected much higher computational cost is required. It should be noted that the results from the 3D non-hydrostatic algorithm do appear favourable when compared to other results in the literature, e.g. Ma et al. (2013) which used an HLL scheme present results similar to those obtained with our 2D algorithm (see their Fig. 2). The time series of surface and bottom horizontal velocities are presented in Fig. 15. With the 3D non-hydrostatic algorithm (Fig. 15a), the maximum flow velocities before hitting the end walls are 1.62 cm/s and 1.67 cm/s at the surface and bottom, respectively. These values are close to the theoretical value (1.59 cm/s) derived from energy balance theory (Turner, 1979). Even with excessive diffusion, the 2D algorithm can also be seen to be able to reproduce the flow velocities with reasonable accuracy (Fig. 15b).

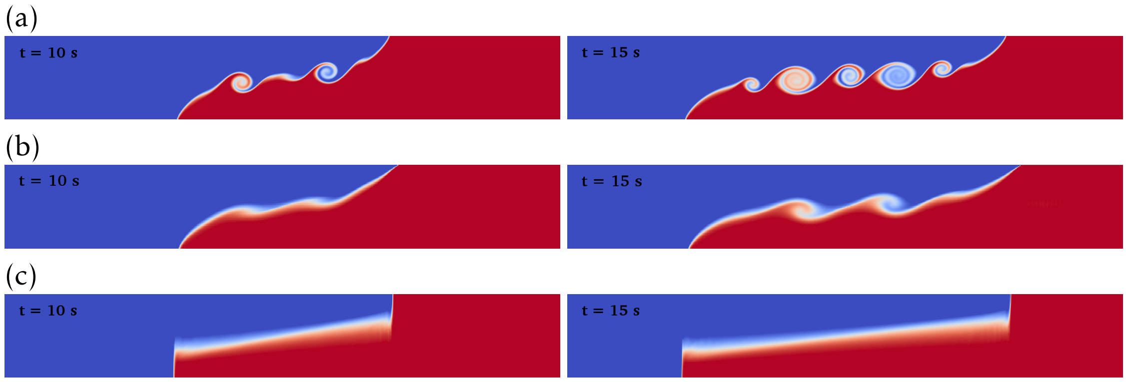

Figure 14: Snapshots of density in the lock exchange case at t = 10 and 15 s (left to right). Top panel (a): non-hydrostatic with the 3D algorithm, middle (b): non-hydrostatic with the 2D algorithm, and bottom (c): hydrostatic simulation. The scale of the density range is 0.1 ∼ 1.0 shown by blue to red colours.

6. Conclusions

This paper presents extensions to the hydrostatic DG-FE based coastal ocean model, *Thetis*, in which two alternative algorithms were used to incorporate non-hydrostatic effects. The 3D algorithm uses a two-stage ALE time integration scheme to advance the whole solution explicitly in time with SSPRK(2,2), which differs from the conventional mode-splitting method adopted in the hydrostatic *Thetis*. The multi-layer 2D algorithm is based on the existing depth-integrated 2D solver framework within Thetis, allowing a full 3D problem to be solved through a combination of 2D computations.

Both algorithms have been implemented and validated using a suite of test cases involving dispersive surface gravity waves and strongly nonlinear internal processes: standing wave oscillation, surface wave shoaling, surface solitary wave generation by a landslide, as well as internal solitary wave evolution and Kelvin-Helmholtz instabilities. Based on identical problem setups, the two algorithms have been compared, allowing an analysis of their respective applicability and limitations. Our findings indicate that the 2D-based algorithm is favourable for the simulation of barotropic surface waves due to its better computational efficiency and accuracy for cases where fewer vertical layers are required. The 3D-based algorithm is found to be better suited for fully baroclinic flows where higher vertical resolution is generally required in order to resolve density fields; in these cases the 2D-based algorithm is found to result in higher levels of diffusion. The

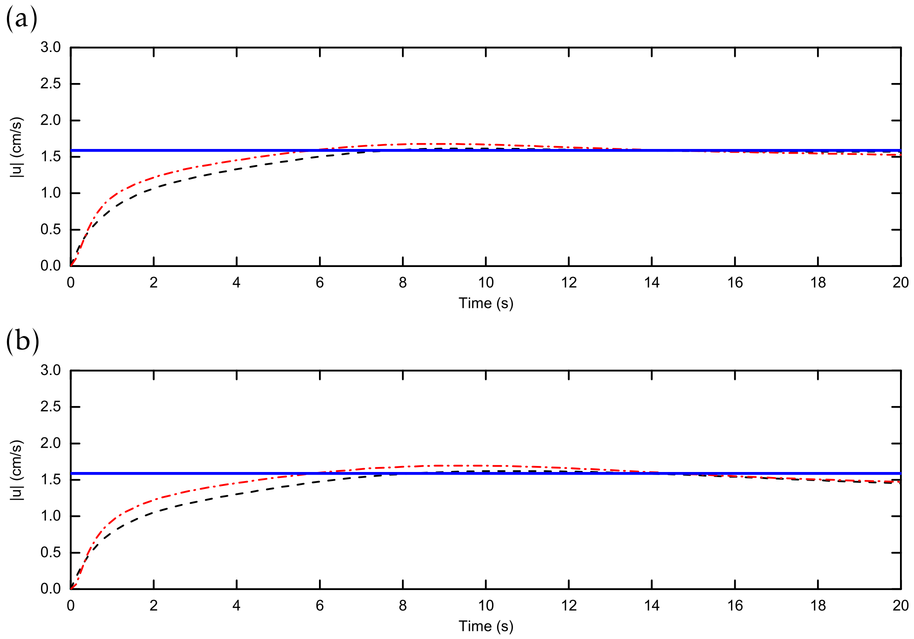

Figure 15: Time series of the horizontal velocities at the midpoint of the upper surface (black dashed line) and bottom of the domain (red dash-dot line) with the 3D NH algorithm (a) and the 2D NH algorithm (b). The blue heavy line indicates the theoretical value (1.59 cm/s) of the velocity (Turner, 1979).

resulting non-hydrostatic model, *Thetis-NH*, thus has good capability for accurately resolving frequency dispersion in an efficient manner, and depending on whether the flow is dominated by baroclinic or barotropic dynamics, two alternative algorithms (i.e. full 3D or multi-layer 2D) can be utilised.

The work presented here has demonstrated non-hydrostatic extensions of both hydrostatic (3D and 2D) solvers present in *Thetis*. The applicability of the *Thetis* model has thus been significantly enhanced in order to simulate a wider variety of geophysical flows. In future work the model will be applied to landslide-tsunami where the generation phase requires the simulation of non-hydrostatic baroclinic flows, and the propagation and inundation phases generally require the consideration of dispersive wave dynamics. Other ongoing work includes the incorporation of goal-based mesh adaptivity and wetting and drying schemes within non-hydrostatic algorithms.

Acknowledgements

WP would like to acknowledge financial support under a China Scholarship Council (CSC) – Imperial College London PhD scholarship. SK and MP would like to acknowledge the support of the UK's Engineering and Physical Sciences Research Council (EPSRC) under grants EP/M011054/1, EP/L000407/1 and EP/R029423/1. All authors would like to thank the Imperial College London Research Computing Service for their support.

AppendixA. Solution method in Firedrake

Firedrake seeks the solutions from a discrete system based on appropriate weak forms. Here take the depth-integrated equations (70) and (71) as an example. By the DG-FE approximations η, d and $\bar{\boldsymbol{u}}$ are treated as $\eta = \sum_i \phi_i \eta_i$, $d = \sum_i \phi_i d_i$, $\bar{\boldsymbol{u}} = \sum_k \phi_k \bar{\boldsymbol{u}}_k$ respectively, and (70) can then be expressed as

$$\left\langle \frac{\partial \sum_i \phi_i \eta_i}{\partial t} \phi_j \right\rangle_\Gamma = \left\langle \left(\sum_i \phi_i (\eta_i + d_i) \right) \left(\sum_k \phi_k \bar{\boldsymbol{u}}_k \right) \cdot \nabla_h \phi_j \right\rangle_\Gamma$$
$$- \left\langle\!\left\langle \left(\sum_i \{\phi_i (\eta_i + d_i)\} \right) \bar{\boldsymbol{u}}^* \cdot [[\phi_j \boldsymbol{n}]] \right\rangle\!\right\rangle_{\mathcal{I}_\Gamma}, \tag{A.1}$$

where ϕ_i and ϕ_k are basis functions of η and $\bar{\boldsymbol{u}}$, ϕ_j denotes the test function, and $\bar{\boldsymbol{u}}^*$ is given by

$$\bar{\boldsymbol{u}}^* = \sum_k \{\phi_k \bar{\boldsymbol{u}}_k\} + \sqrt{\frac{g}{\sum_i \{\phi_i (\eta_i + d_i)\}}} \left[\left[\sum_i \phi_i \eta_i \boldsymbol{n}_h \right]\right].$$

The matrix form of (A.1) can be written as

$$\boldsymbol{M} \frac{d\boldsymbol{E}}{dt} = \boldsymbol{B}_\eta \iff \frac{d\boldsymbol{E}}{dt} = (\boldsymbol{M})^{-1} \boldsymbol{B}_\eta, \tag{A.2}$$

in which $[\boldsymbol{M}]_{ij} = \left\langle \phi_i \phi_j \right\rangle_\Gamma$ is the mass matrix, $[\boldsymbol{E}]_i = \eta_i$ and $\boldsymbol{B}_\eta$ denotes the right hand side matrix of (A.1). Analogously, the matrix form of (71) can be obtained, which is coupled with (A.2) giving a full block-coupled system.

In term of the time integration, (A.2) becomes

$$\frac{\boldsymbol{E}^{n+1} - \boldsymbol{E}^n}{\Delta t} = (\boldsymbol{M})^{-1} \left[\theta \boldsymbol{B}_\eta^{n+1} + (1-\theta) \boldsymbol{B}_\eta^n \right], \tag{A.3}$$

where the θ-scheme is adopted here, and if $\theta = 0.5$, it is equivalent to Crank-Nicolson.

The PETSc library is used to solve the full block-coupled system, via which various choices of solvers and preconditioners are available. Consider the general form of the system:

$$\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_2 \\ \boldsymbol{M}_3 & \boldsymbol{M}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{E} \\ \bar{\boldsymbol{U}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_\eta \\ \boldsymbol{B}_{\boldsymbol{u}} \end{bmatrix}, \tag{A.4}$$

where $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_4$ are matrix blocks and $\boldsymbol{B}_{\boldsymbol{u}}$ is the right hand side of the momentum equation (71).

Factorising the left hand side of (A.4) by LDU (lower-diagonal-upper) block factorisation (Elman et al., 2014), we have

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{M}_3 \boldsymbol{M}_1^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_1^{-1} \boldsymbol{M}_2 \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{A.5}$$

where $\boldsymbol{S} = \boldsymbol{M}_4 - \boldsymbol{M}_3 \boldsymbol{M}_1^{-1} \boldsymbol{M}_2$ is the Schur complement. The inverse of the factorised system is

$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{M}_1^{-1} \boldsymbol{M}_2 \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_1^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{S}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\boldsymbol{M}_3 \boldsymbol{M}_1^{-1} & \boldsymbol{I} \end{bmatrix}. \tag{A.6}$$

The fieldsplit preconditioner is employed to find approximations to the actions of $\boldsymbol{S}^{-1}$ and $\boldsymbol{M}_1^{-1}$, for which a broad configuration space is available in PETSc, allowing various iterative solvers and preconditioners for $\boldsymbol{S}^{-1}$ and $\boldsymbol{M}_1^{-1}$. In this study, incomplete LU (ILU) factorisation is used, and the convergence criterion adopts a relative error of 10^{-7} for the iterative solvers.

References

Adcroft, A., Campin, J., Dutkiewicz, S., Evangelinos, C., Ferreira, D., Forget, G., Fox-Kemper, B., Heimbach, P., Hill, C., Hill, E., et al., 2008. Mitgcm user manual.

Alnæs, M. S., Logg, A., Ølgaard, K. B., Rognes, M. E., Wells, G. N., 2014. Unified form language: A domain-specific language for weak formulations of partial differential equations. ACM Transactions on Mathematical Software (TOMS) 40 (2), 9.

Angeloudis, A., Kramer, S. C., Avdis, A., Piggott, M. D., 2018. Optimising tidal range power plant operation. Applied Energy 212, 680 – 690.

Bai, Y., Cheung, K. F., 2016. Linear and nonlinear properties of reduced two-layer models for non-hydrostatic free-surface flow. Ocean Modelling 107, 64–81.

Balay, S., Abhyankar, S., Adams, M., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W., et al., 2019. Petsc users manual.

Beji, S., Battjes, J. A., 1994. Numerical simulation of nonlinear wave propagation over a bar. Coastal Engineering 23 (1), 1–16.

Blaise, S., Comblen, R., Legat, V., Remacle, J.-F., Deleersnijder, E., Lambrechts, J., 2010. A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Ocean Dynamics 60 (6), 1371–1393.

Bonev, B., Hesthaven, J. S., Giraldo, F. X., Kopera, M. A., 2018. Discontinuous Galerkin scheme for the spherical shallow water equations with applications to tsunami modeling and prediction. Journal of Computational Physics 362, 425–448.

Burchard, H., Bolding, K., 2002. Getm–a general estuarine transport model. Scientific documentation, no EUR 20253.

Casulli, V., Stelling, G. S., 1998. Numerical simulation of 3d quasi-hydrostatic, free-surface flows. Journal of Hydraulic Engineering 124 (7), 678–686.

Casulli, V., Zanolli, P., 2002. Semi-implicit numerical modeling of nonhydrostatic free-surface flows for environmental problems. Mathematical and computer modelling 36 (9-10), 1131–1149.

Chen, C., Beardsley, R., Cowles, G., 2006. An unstructured grid, finite-volume coastal ocean model. FVCOM User Manual,.

Chorin, A. J., 1968. Numerical solution of the Navier-Stokes equations. Mathematics of computation 22 (104), 745–762.

Clare, M. C. A., Percival, J. R., Angeloudis, A., Cotter, C. J., Piggott, M. D., 2019. Hydro-morphodynamics 2D modelling using a discontinuous galerkin discretisation. Computers & Geosciences (submitted).

Comblen, R., Lambrechts, J., Remacle, J.-F., Legat, V., 2010. Practical evaluation of five partly discontinuous finite element pairs for the non-conservative shallow water equations. International Journal for Numerical Methods in Fluids 63 (6), 701–724.

Donea, J., Huerta, A., Ponthot, J.-P., Rodríguez-Ferran, A., 2004. Arbitrary l agrangian–e ulerian methods. Encyclopedia of computational mechanics.

Elman, H. C., Silvester, D. J., Wathen, A. J., 2014. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics. OUP Oxford.

Enet, F., Grilli, S. T., 2007. Experimental study of tsunami generation by three-dimensional rigid underwater landslides. Journal of waterway, port, coastal, and ocean engineering 133 (6), 442–454.

Epshteyn, Y., Rivière, B., 2007. Estimation of penalty parameters for symmetric interior penalty galerkin methods. Journal of Computational and Applied Mathematics 206 (2), 843–872.

Fringer, O. B., Gerritsen, M., Street, R. L., 2006. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator. Ocean Modelling 14 (3-4), 139–173.

Gottlieb, S., Shu, C.-W., 1998. Total variation diminishing runge-kutta schemes. Mathematics of computation of the American Mathematical Society 67 (221), 73–85.

Härtel, C., Meiburg, E., Necker, F., 2000. Analysis and direct numerical simulation of the flow at a gravity-current head. part 1. flow topology and front speed for slip and no-slip boundaries. Journal of Fluid Mechanics 418, 189–212.

Hiester, H. R., Piggott, M. D., Allison, P. A., 2011. The impact of mesh adaptivity on the gravity current front speed in a two-dimensional lock-exchange. Ocean Modelling 38 (1-2), 1–21.

Higdon, R. L., de Szoeke, R. A., 1997. Barotropic-baroclinic time splitting for ocean circulation modeling. Journal of Computational Physics 135 (1), 30–53.

Hill, J., Collins, G. S., Avdis, A., Kramer, S. C., Piggott, M. D., 2014. How does multiscale modelling and inclusion of realistic palaeobathymetry affect numerical simulation of the storegga slide tsunami? Ocean Modelling 83, 11–25.

Horn, D., Imberger, J., Ivey, G., 2001. The degeneration of large-scale interfacial gravity waves in lakes. Journal of Fluid Mechanics 434, 181–207.

Jacobs, C. T., Piggott, M. D., 2015. Firedrake-fluids v0.1: numerical modelling of shallow water flows using an automated solution framework. Geoscientific Model Development 8 (3), 533–547.

Kanarska, Y., Shchepetkin, A., McWilliams, J., 2007. Algorithm for non-hydrostatic dynamics in the regional oceanic modeling system. Ocean Modelling 18 (3-4), 143–174.

Kärnä, T., 2019. Discontinuous galerkin discretization for two-equation turbulence closure model. arXiv preprint arXiv:1907.04624.

Kärnä, T., Kramer, S. C., Mitchell, L., Ham, D. A., Piggott, M. D., Baptista, A. M., 2018. Thetis coastal ocean model: discontinuous Galerkin discretization for the three-dimensional hydrostatic equations. Geoscientific Model Development 11, 4359–4382.

Kärnä, T., Legat, V., Deleersnijder, E., 2013. A baroclinic discontinuous galerkin finite element model for coastal flows. Ocean Modelling 61, 1–20.

Kuzmin, D., 2010. A vertex-based hierarchical slope limiter for p-adaptive discontinuous galerkin methods. Journal of computational and applied mathematics 233 (12), 3077–3085.

Lai, Z., Chen, C., Cowles, G. W., Beardsley, R. C., 2010. A nonhydrostatic version of FVCOM: 1. validation experiments. Journal of Geophysical Research: Oceans 115 (C11).

Logg, A., Mardal, K.-A., Wells, G., 2012. Automated solution of differential equations by the finite element method: The FEniCS book. Vol. 84. Springer Science & Business Media.

Ma, G., Kirby, J. T., Shi, F., 2013. Numerical simulation of tsunami waves generated by deformable submarine landslides. Ocean Modelling 69, 146–165.

Madec, G., et al., 2015. Nemo ocean engine.

Marshall, J., Adcroft, A., Hill, C., Perelman, L., Heisey, C., 1997. A finite-volume, incompressible navier stokes model for studies of the ocean on parallel computers. Journal of Geophysical Research: Oceans 102 (C3), 5753–5766.

Pan, W., Kramer, S. C., Piggott, M. D., 2019. Multi-layer non-hydrostatic free surface modelling using the discontinuous galerkin method. Ocean Modelling 134, 68–83.

Piggott, M. D., Gorman, G. J., Pain, C. C., Allison, P. A., Candy, A. S., Martin, B. T., Wells, M. R., 2008a. A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. International Journal for Numerical Methods in Fluids 56 (8), 1003–1015.

Piggott, M. D., Pain, C. C., Gorman, G. J., Marshall, D. P., Killworth, P. D., 2008b. Unstructured Adaptive Meshes for Ocean Modeling. In: Hecht, M. W., Hasumi, H. (Eds.), Ocean Modeling in an Eddying Regime. American Geophysical Union, pp. 383–408.

Rathgeber, F., Ham, D. A., Mitchell, L., Lange, M., Luporini, F., McRae, A. T., Bercea, G.-T., Markall, G. R., Kelly, P. H., 2016. Firedrake: automating the finite element method by composing abstractions. ACM Transactions on Mathematical Software (TOMS) 43 (3), 24.

Ringler, T., Petersen, M., Higdon, R. L., Jacobsen, D., Jones, P. W., Maltrud, M., 2013. A multi-resolution approach to global ocean modeling. Ocean Modelling 69, 211–232.

Shchepetkin, A. F., McWilliams, J. C., 2005. The regional oceanic modeling system (roms): a split-explicit, free-surface, topography-following-coordinate oceanic model. Ocean modelling 9 (4), 347–404.

Smith, R. C., Hill, J., Collins, G. S., Piggott, M. D., Kramer, S. C., Parkinson, S. D., Wilson, C., 2016. Comparing approaches for numerical modelling of tsunami generation by deformable submarine slides. Ocean Modelling 100, 125 – 140.

Stelling, G., Zijlema, M., 2003. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. International Journal for Numerical Methods in Fluids 43 (1), 1–23.

Turner, J. S., 1979. Buoyancy effects in fluids. Cambridge university press.

Vallaeys, V., Kärnä, T., Delandmeter, P., Lambrechts, J., Baptista, A. M., Deleersnijder, E., Hanert, E., 2018. Discontinuous galerkin modeling of the columbia river's coupled estuary-plume dynamics. Ocean Modelling 124, 111–124.

Vitousek, S., Fringer, O. B., 2013. Stability and consistency of nonhydrostatic free-surface models using the semi-implicit θ-method. International Journal for Numerical Methods in Fluids 72 (5), 550–582.

Vouriot, C. V. M., Angeloudis, A., Kramer, S. C., Piggott, M. D., 2018. Fate of large-scale vortices in idealized tidal lagoons. Environmental Fluid Mechanics.
URL https://doi.org/10.1007/s10652-018-9626-4

Zhang, Y., Baptista, A. M., 2008. Selfe: a semi-implicit eulerian–lagrangian finite-element model for cross-scale ocean circulation. Ocean modelling 21 (3-4), 71–96.